



# AI-Guided Rocket Landing: Navigating Precision Descent Strategies

Hicham Bouchana<sup>1</sup>, Meftah Zouai<sup>1,2</sup> Ahmed Aloui<sup>1</sup>, Guadalupe Ortiz<sup>2</sup>, and Dounya Kassimi<sup>3,\*</sup>

<sup>1</sup> Mohamed Khider University, Biskra, Algeria,

<sup>2</sup> University of Cadiz, Cadiz, Spain

<sup>3</sup> University Center Aflou, Aflou, Algeria

\* Corresponding author: Dounya Kassimi [d.kassimi@cu-aflou.edu.dz](mailto:d.kassimi@cu-aflou.edu.dz)

**Abstract.** Autonomous rocket landing stands as a crucial milestone in aerospace engineering, pivotal for the realization of safe and cost-effective space missions. This paper introduces a pioneering approach that harnesses reinforcement learning methodologies to enhance the precision and efficiency of rocket landing procedures. Grounded on a realistic Falcon 9 model, the study integrates sophisticated control mechanisms including Thrust Vector Control (TVC) and Cold Gas Thrusters (CGT), ensuring agile propulsion and balance adjustments. Observational data, encompassing critical parameters like rocket position, orientation, and velocity, guide the reinforcement learning algorithm in making real-time decisions to optimize landing trajectories. Through the strategic implementation of curriculum learning strategies and the Proximal Policy Optimization (PPO) algorithm, the rocket agent undergoes iterative training, steadily improving its capabilities to execute soft landings on designated pads. Experimental results underscore the efficacy of the proposed methodology, exhibiting remarkable proficiency in achieving precise and controlled descents. This research represents a significant stride in the advancement of autonomous landing systems, poised to revolutionize space exploration missions and unlock new frontiers in commercial rocketry endeavors.

**Keywords:** Autonomous rocket landing; Reinforcement learning; Precision landing; Real-time decision-making; Proximal Policy Optimization (PPO) algorithm; Unity ML-Agents

## 1 Introduction

Autonomous rocket landing represents a critical aspect of modern aerospace engineering, with profound implications for space exploration and commercial rocketry. The ability to safely and precisely land rockets not only facilitates the recovery and reuse of expensive space hardware but also enables missions to remote or challenging terrains on celestial bodies.

Traditional approaches to rocket landing have often relied on pre-programmed trajectories and manual intervention, limiting their adaptability to dynamic environmental conditions and increasing the risk of mission failure. In recent years,

© The Author(s) 2024

C. A. Kerrache et al. (eds.), *Proceedings of the International Conference on Emerging Intelligent Systems for Sustainable Development (ICEIS 2024)*, Advances in Intelligent Systems Research 184,

[https://doi.org/10.2991/978-94-6463-496-9\\_28](https://doi.org/10.2991/978-94-6463-496-9_28)

however, there has been a growing interest in leveraging advanced artificial intelligence techniques, particularly reinforcement learning, to enhance the autonomy and efficiency of rocket landing procedures.

This paper presents a novel approach to autonomous rocket landing that harnesses the power of reinforcement learning to enable real-time decision-making and trajectory optimization. Building upon a realistic Falcon 9 model [3], our research integrates sophisticated control mechanisms, including Thrust Vector Control (TVC) [11] and Cold Gas Thrusters (CGT) [1], to achieve precise propulsion and balance adjustments during descent.

By capturing key observational data such as rocket position, orientation, and velocity, our reinforcement learning algorithm learns to make informed decisions to optimize landing trajectories, adapt to changing environmental conditions, and ensure safe and controlled descents onto designated landing pads. Through the strategic implementation of curriculum learning strategies and the Proximal Policy Optimization (PPO) algorithm [10], our rocket agent undergoes iterative training to continually refine its landing capabilities and improve overall performance.

The outcomes of this research have significant implications for the future of space exploration and commercial rocketry, offering a path towards more cost-effective and reliable missions, broader access to space, and the realization of ambitious scientific objectives.

## 2 Related Work

In [13], the authors introduce an intelligent control algorithm for rocket landings, utilizing deep reinforcement learning and Long Short Term Memory (LSTM) neural networks. By combining imitation and reinforcement learning, the method accelerates training and achieves real-time control on an embedded platform. The algorithm demonstrates superior landing accuracy, rapid convergence, and enhanced adaptability, showcasing its potential for autonomous rocket landings without heavy reliance on precise control models.

The project [9] aims to use Unity ML-Agents and deep reinforcement learning for autonomous orbital rocket landings, focusing on reusable first-stage rockets. This introduces innovative autonomous control, aligning with SpaceX's progress in rapid reusability of launch vehicles.

This project [4] develops a simulation for evaluating classical control and machine learning algorithms in vertical rocket landings. It implements Proportional Integral Derivative (PID) controller, Model Predictive Control (MPC), and RL algorithms like Deep Deterministic Policy Gradients (DDPG), showcasing potential for stable and consistent landings. Aligned with SpaceX and Blue Origin's successes, it contributes to the broader goal of reusable space systems, addressing the challenge of autonomous rocket landings.

It's important to acknowledge the advancements our simulation offers compared to previous research by Ferrante et al [4]. Our work leverages a fully immersive 3D environment, allowing for a more comprehensive evaluation of the

AI control system's capabilities. In contrast, Ferrante et al.'s research employed a 2D simulation setting, potentially limiting the complexity of the landing scenarios explored.

Furthermore, our simulation expands the control options available to the AI compared to the two degrees of control (X-axis for lateral movements) and two Cold Gas Thrusters (CGTs) on the left and right featured in his work. We introduce a more comprehensive control scheme by incorporating four Cold Gas Thrusters (left, right, front, and back) and a thrust vector direction that can be adjusted along both the X and Z axes. This enhanced controllability allows for a more realistic simulation of rocket dynamics and a more rigorous assessment of the AI's ability to perform complex maneuvers during landing.

## 3 Methodology

### 3.1 Rocket Representation

In the Unity simulation environment [8], we meticulously model the rocket and its landing pad, mirroring the design of the Falcon 9. The rocket, governed by real-world physics, possesses essential attributes such as mass, a rigid body structure, and gravitational forces. Two crucial components, the Thrust Vector Control (TVC) and Cold Gas Thrusters (CGT), are integrated for precise rocket maneuvering in space.

**Falcon 9 Model** The Falcon 9 model serves as the foundation, adhering to the laws of physics with realistic attributes like mass, a rigid body structure, and gravitational interactions [3].

**TVC and CGT Integration** The TVC and CGT components are implemented, introducing realistic control mechanisms for propulsion and balance.

### 3.2 Observations for Reinforcement Learning

The reinforcement learning agent relies on a comprehensive set of observations (22 in total) to guide its decision-making throughout the landing process. These observations encompass critical aspects of the rocket's state and the landing environment:

**Rocket State (9 observations):** Position (X, Y, Z), Rotation (Pitch, Yaw, Roll via Euler angles), Velocity (X, Y, Z)

**Landing Environment (3 observations):** Landing pad position (X, Y, Z)

**Distance (2 observations):** The agent receives the horizontal distance to the landing pad, aiding in centering the landing

**Landing Gear (4 observations):** The agent senses if each leg (1-4) is touching the ground, aiding in understanding landing stability.

**Thrusters (4 observations):** The agent tracks whether each Cold Gas Thruster (right, left, front, back) is firing. This information helps the agent understand the fine-tuning of its balance during descent by correlating Cold Gas Thruster activation with the overall TVC (Thrust Vector Control) propulsion. This allows for precise adjustments beyond the control offered by the TVC system alone. This rich observational data stream enables the agent to perceive its position, orientation, movement, and its relative location to the landing pad. This information is crucial for the agent to learn effective control strategies that lead to precise and safe landings.

### 3.3 TVC Actions and Realism Enhancement

The TVC, a key element in rocket control, initially employs six discrete actions, allowing precise force application in different directions (Up, Left, Right, Front, Back, Off). Future iterations aim to enhance realism by transitioning to continuous actions, enabling a broader range of movement directions within realistic constraints.

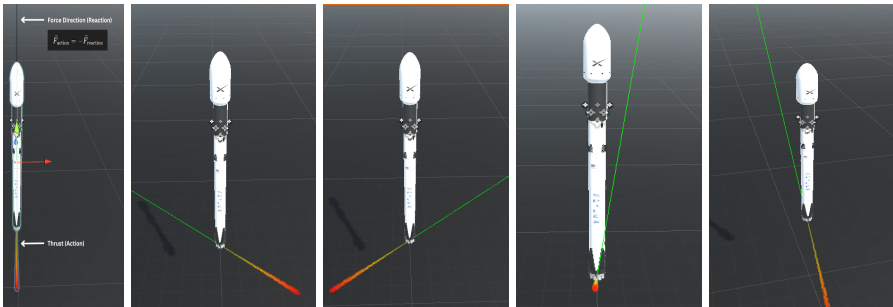
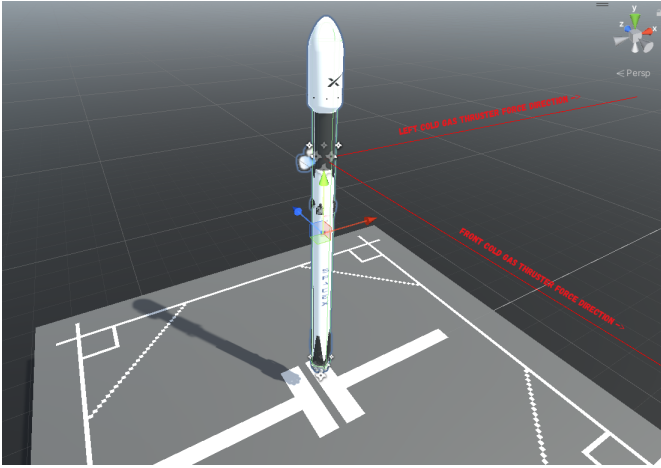


Fig. 1. Rocket with TVC Activation (UP,Right,Left,Front,Back).

### 3.4 Cold Gas Thrusters (CGT)

Four Cold Gas Thrusters, strategically positioned atop the first stage, augment the TVC by providing corrective forces for balance. These thrusters contribute to the overall stability of the rocket, serving as a contingency mechanism if the TVC encounters challenges. This comprehensive representation of the rocket environment, incorporating realistic physics, precise control mechanisms, and dynamic observations, forms the foundation for our reinforcement learning system. The integration of TVC, CGT, and accurate observations positions our project at the forefront of simulating and optimizing rocket landing strategies.



**Fig. 2.** Rocket with CGT Activation (Left and Front).

## 4 Rocket Agent Actions

The rocket agent controls two primary systems: the Thrust Vector Control (TVC) and the Cold Gas Thrusters (CGT).

1. Thrust Vector Control (TVC):
  - The TVC can be activated in five directions: Down, Right, Left, Front, Back, or Off.
  - Only one direction can be active at a time.
  - This action directly influences the rocket’s main engine thrust vector, providing directional control.
2. Cold Gas Thrusters (CGT):
  - Each thruster (Left, Right, Front, Back) can be independently switched on or off.
  - The agent can activate any combination of thrusters simultaneously (including none or all four).
  - These thrusters provide fine-grained control over the rocket’s rotational movement.

The rocket agent controls two primary systems, offering a total of 14 possible actions through their combined functionality.

**Table 1.** Rocket Agent Control Options

System	Action	Force Vector (X,Y,Z)	Description
TVC	Down	(0, 1, 0)	Main engine thrust directed downwards.
	Right	(1, 1, 0)	Main engine thrust directed rightwards.
	Left	(-1, 1, 0)	Main engine thrust directed leftwards.
	Front	(0, 1, 1)	Main engine thrust directed forwards.
	Back	(0, 1, -1)	Main engine thrust directed backwards.
	Off	(0, 0, 0)	No main engine thrust.
CGT	Left (ON)	(1, 0, 0)	Left thruster firing, providing a positive force to the right.
	Left (OFF)	(0, 0, 0)	Left thruster not firing.
	Right (ON)	(-1, 0, 0)	Right thruster firing, providing a positive force to the left.
	Right (OFF)	(0, 0, 0)	Right thruster not firing.
	Front (ON)	(0, 0, -1)	Front thruster firing, providing a positive force backward.
	Front (OFF)	(0, 0, 0)	Front thruster not firing.
	Back (ON)	(0, 0, 1)	Back thruster firing, providing a positive force forward.
	Back (OFF)	(0, 0, 0)	Back thruster not firing.

By combining TVC maneuvers with strategic CGT activation based on observed rotation, the agent learns to control the rocket’s orientation and achieve a successful landing.

## 5 Training The Rocket Agent

Reinforcement Learning (RL) is a machine learning [7] paradigm where an agent learns to make decisions by interacting with an environment. Through a series of actions, the agent receives feedback in the form of rewards or punishments, shaping its behavior over time. RL is characterized by the agent’s exploration of actions to discover optimal strategies and its exploitation of learned knowledge to maximize cumulative rewards.

Our project utilizes a discrete deep reinforcement learning approach to train a rocket agent to perform a soft landing on a designated landing pad within a simulated environment. The agent receives observations about its state (position, rotation, velocity) and the landing pad’s position, and then takes discrete actions to control the thrust direction and force of the rocket’s engines (TVC) and Cold Gas Thrusters (CGT).

The agent continuously interacts with the environment, and a reward function is used to evaluate the success of these actions. This feedback signal allows the agent to learn through trial and error, progressively improving its control policy to achieve the desired outcome (soft landing).

### 5.1 Reward System

The reward function is a pivotal component in shaping the agent’s behavior, assigning a numerical reward signal to the agent after each action, indicating its

performance relative to the goal. Our reward system takes into account several factors contributing to a successful landing:

- **Distance to Landing Pad:** A positive reward is granted as the agent approaches the landing pad, encouraging movement towards the target location [12].
- **Rocket Rotation:** The agent receives positive rewards for maintaining a relatively upright position during descent. Penalties are applied for exceeding a predefined rotation threshold on specific axes (pitch and roll) to discourage tilting or flipping [5].
- **Altitude:** Positive rewards are given as the rocket descends towards the landing pad. However, reaching zero altitude is not the sole objective, promoting a controlled descent rather than a crash landing [2].

The final reward at each step is calculated by combining these individual rewards using weighted averages. The weights are tuned to prioritize different aspects based on the desired landing criteria (e.g., assigning a higher weight to distance for prioritizing landing on the pad over achieving a perfectly upright posture).

## 5.2 Curriculum Learning

To enhance the adaptability and learning capabilities of the rocket agent, we implement curriculum learning strategies. This involves systematically adjusting the complexity of the learning environment to facilitate a more gradual learning process. In our case, curriculum learning is applied through the following mechanisms:

- **Randomized Initial Conditions:** The rocket’s initial conditions, including yaw, roll, and pitch angles, are randomized within a predefined range, such as from -10 to 10 degrees. This variation encourages the agent to learn robust control policies that can handle a diverse set of starting orientations.
- **Extended Range of Angles:** The initial angle ranges are progressively expanded during training. By gradually increasing the range of yaw, roll, and pitch values, the agent is exposed to a wider spectrum of scenarios. This approach encourages the development of more versatile control strategies capable of handling a broader array of starting conditions.
- **Varied Starting Altitudes:** In addition to angle variations, the rocket is dropped from different starting altitudes. This introduces variability in the descent phase, promoting the acquisition of adaptive control strategies under diverse altitude conditions.

These curriculum learning techniques aim to provide a well-structured training environment that challenges the rocket agent progressively, facilitating the acquisition of robust and generalized control policies.

## 6 Final Touches

In this final phase of our rocket landing project, we delve into the implementation details and fine-tuning aspects that contribute to the success of our reinforcement learning system. The Proximal Policy Optimization (PPO) algorithm is employed as the chosen technique for training our rocket agent. Furthermore, the environment setup for the learning process involves the use of multiple copies of the training area to expedite the training process.

### 6.1 Technique: Proximal Policy Optimization (PPO)

The Proximal Policy Optimization (PPO) algorithm serves as the cornerstone of our reinforcement learning strategy. PPO is a robust and widely utilized approach in training agents for complex tasks. Its ability to strike a balance between stability and sample efficiency makes it particularly suitable for our rocket landing scenario [6].

### 6.2 Environment Setup for Accelerated Learning

To accelerate the learning process and enhance the training efficiency, we employ multiple copies of the training area as seen in figure 3. This approach allows the rocket agent to gather diverse experiences concurrently, fostering a more rapid acquisition of optimal landing strategies. The parallelized setup contributes to the scalability and effectiveness of our reinforcement learning system.

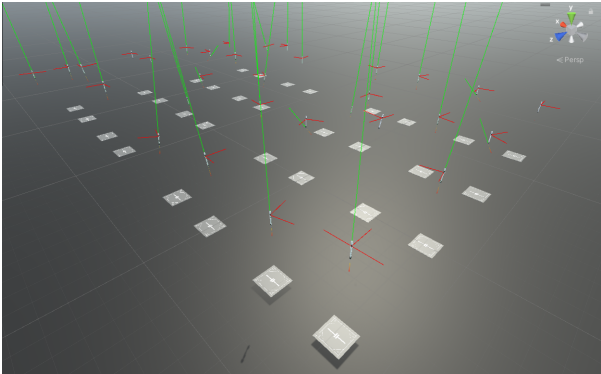


Fig. 3. Parallelized Training Environments for Rocket Agent.

### 6.3 Configuration Summary

The table 2 provides a concise summary of the key parameters and settings in the "RocketNNsettings.yaml" configuration file for the rocket agent's training:

**Table 2.** Rocket Agent Training Configuration Summary

Parameter	Value
Batch Size	64
Buffer Size	12000
Learning Rate	0.0003
Beta	0.001
Epsilon	0.2
Lambda	0.99
Num Epoch	3
Learning Rate Schedule	Linear
Normalize	True
Hidden Units	128
Num Layers	2
Vis Encode Type	Simple
Gamma (Extrinsic)	0.99
Strength (Extrinsic)	1.0
Keep Checkpoints	5
Max Steps	100,000,000
Time Horizon	1000
Summary Frequency	10,000

The corresponding value to each key parameter is used for training the rocket agent within the Unity ML-Agents framework. These hyperparameters were chosen through a combination of:

**Prior research and best practices:** We considered baseline values and common ranges used for PPO (Proximal Policy Optimization) reinforcement learning algorithms in similar continuous control tasks.

**Preliminary experimentation:** We conducted initial training runs with various parameter settings to assess their impact on the agent’s learning performance and convergence rate.

**Fine-tuning:** Based on the initial experiments, we further refined specific parameters to achieve optimal training efficiency and performance for our specific rocket landing task.

Here’s a breakdown of some key parameters and their justifications:

- **Batch Size (64):** The Batch Size parameter determines how many completed episodes are grouped together for updating the agent’s policy network (the “brain” that guides its decision-making). This is why 64 is a common choice for batch-based reinforcement learning algorithms, balancing computational efficiency with learning progress. Typical range: 32 - 512
- **Learning Rate (0.0003):** This value helps the agent learn at a steady pace without encountering instability or slow convergence. Typical range:  $1e^{-5} - 1e^{-3}$
- **Beta (0.001):** Beta’s Role: Beta directly influences the clipping range used in PPO. A higher Beta value leads to a stricter clipping range, meaning the updated policy is closer to the previous one, promoting exploitation.

Conversely, a lower Beta value allows for a looser clipping range, giving the agent more freedom to explore new actions. Typical range:  $1e^{-4} - 1e^{-2}$

- **Epsilon (0.2)**: This exploration parameter allows the agent to balance exploitation (choosing known good actions) with exploration (trying new actions) during training. Typical range: 0.1 - 0.3
- **Hidden Units (128) and Num Layers (2)**: This configuration provides a sufficient neural network capacity for the agent to learn complex control policies for the rocket landing task.

The remaining parameters in Table 2 control various aspects of the training process, such as buffer size, learning rate schedule, and checkpoint frequency. These were chosen to ensure efficient training and facilitate the evaluation of the trained agent’s performance.

## 7 Results

In this section, we delve into the outcomes and insights derived from the experimental phase of our AI-guided rocket landing project. The primary objectives of these experiments were to assess the performance and effectiveness of our reinforcement learning system, particularly in the context of autonomous rocket landings.

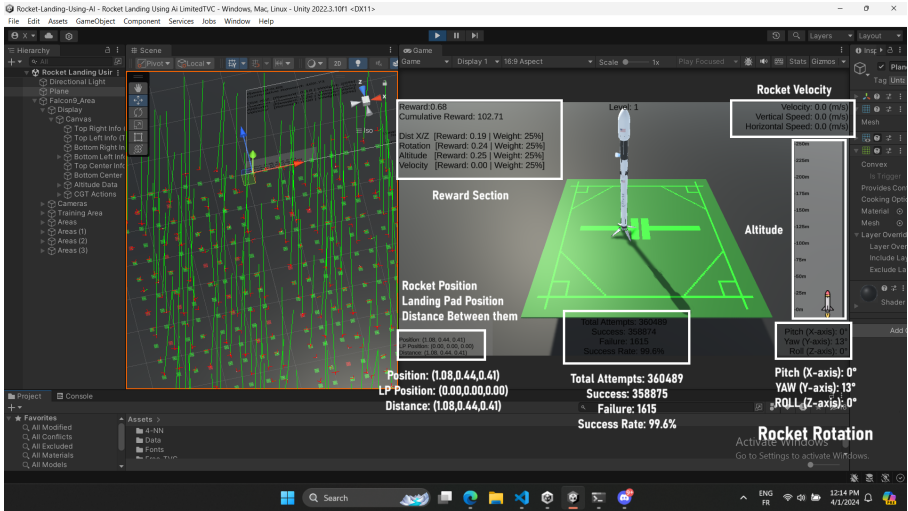


Fig. 4. Unity Simulation Scene + Dynamic Monitoring View

This section details the visualization components within the Unity engine used to monitor and evaluate the performance of the reinforcement learning agent during its training for autonomous rocket landings.

**Simulation Scene with Raycast:** The left window of Figure 4 showcases the simulation environment. This includes a visualization of the rocket and the landing pad. Here, a raycast is employed to depict the force direction applied by the agent’s control decision (action) at each timestep. This visual aid helps understand the relationship between the chosen action and the resulting force acting on the rocket.

**Camera View and Force Visualization:** The right window provides a camera view attached to the rocket, allowing observation of its orientation and balance throughout the descent. Additionally, a particle system dynamically visualizes the thrust forces (both Cold Gas Thrusters (CGT) and Thrust Vector Control (TVC)) acting on the rocket. The particles are emitted in the opposite direction of the applied force, offering a clear representation of the forces influencing the rocket’s motion.

**Reward Structure and Visualization:** The top right corner of the right window displays the current velocity of the rocket, including its vertical and horizontal components.

The top left corner displays the real-time reward information for various aspects of the rocket’s state:

- Distance (X/Z Axis) Reward (25%): This reward encourages the agent to land close to the center of the landing pad. As the distance from the center increases, the reward decreases.
- Rotation Reward (25%): This reward incentivizes the agent to maintain a minimal roll, pitch, and yaw throughout descent. A value of zero for all rotations yields the maximum reward.
- Altitude Reward (25%): This reward motivates the agent to descend towards the landing pad. The reward starts at zero and increases as the rocket gets closer to the landing pad’s height.
- Velocity Reward (25%): This reward encourages a controlled descent. The maximum reward is received when the rocket’s velocity reaches a target value (e.g., 10 m/s in this case). This target velocity can be adjusted to reflect real-world landing requirements.

These individual rewards are all added up for each step the agent takes during the landing process. Since each contributes 25% (0.25), the total reward can potentially reach a maximum of 1 each step if all aspects are performing perfectly throughout the entire landing attempt.

**Footnote:** Within the context of our reinforcement learning setup for autonomous rocket landing, a “step” signifies a fundamental unit of time advancement within the Unity simulation environment. During each step, the environment updates its state, the agent receives observations, and the agent makes a control decision based on those observations. The total reward an agent accumulates is calculated by summing up the individual rewards received at each step throughout the landing episode.

we establish a maximum number of steps allowed for each difficulty level (initial drop height) 3. This value reflects the increasing complexity associated

with higher initial starting heights (Levels 1-10). Episodes exceeding this limit are considered unsuccessful.

This approach provides a clearer understanding of the training process, acknowledging the dynamic nature of episode lengths while showcasing the maximum effort allocated to completing a run at each difficulty level.

**Table 3.** Maximum steps allowed in each run

<b>Initial drop height</b>	<b>(100m)</b>	<b>(100m, 200m)</b>	<b>(200m, 300m)</b>	<b>(300m, 400m)</b>	<b>(400m, 500m)</b>	<b>(500m, 700m)</b>	<b>(700m, 1000m)</b>	<b>(1000m, 1300m)</b>	<b>(1300m, 1700m)</b>	<b>(1700m, 2000m)</b>
<b>Maximum steps</b>	1500	2000	2500	3000	3500	5500	6000	7000	11500	18000

**Additional Information Display:**

**Bottom Right:** Pitch, Yaw, and Roll angles of the rocket in real-time.

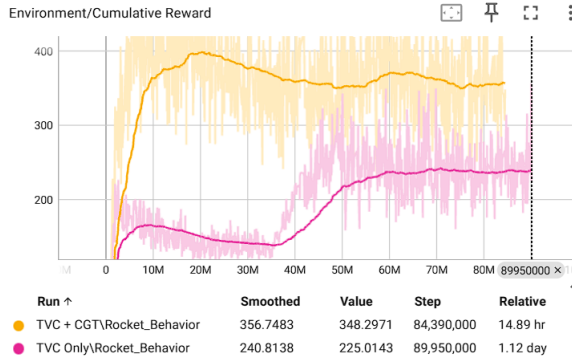
**Bottom Left:** The current position of the rocket and the landing pad, along with the distance between them.

**Bottom Center:** Total training attempts made, Successful landing attempts, Failed landing attempts, Success rate.

**Top Center:** Difficulty level of the current training scenario. This level determines the initial randomness applied to the rocket’s starting drop height and orientation.

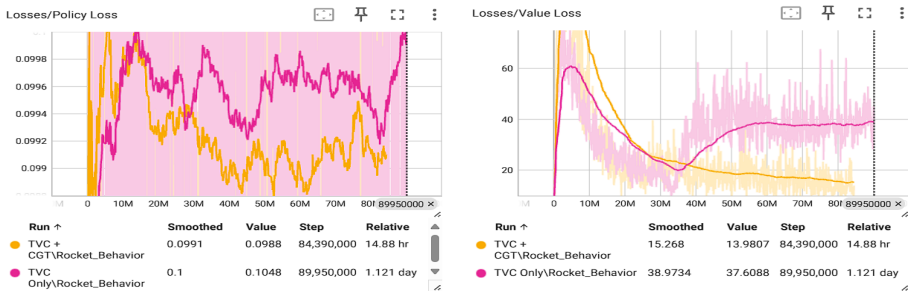
**Criteria for Successful Landing:** The landing pad visually communicates the outcome of each landing attempt. A successful landing is defined as the rocket remaining upright and stationary on the landing pad for a minimum duration of 5 seconds. If this success criterion is met, the landing pad turns green. Conversely, if the rocket tips over, crashes, or fails to remain stationary for the specified duration, the landing pad turns red, indicating an unsuccessful landing.

Figure 4 showcases an impressive achievement. After **360,489** training attempts, the agent achieved a remarkable success rate of **99.6%**, leaving only a negligible **0.4%** failure rate. This exceptional outcome demonstrates the effectiveness of the reinforcement learning approach in enabling the agent to acquire control strategies for precise and safe landings within the simulated environment.



**Fig. 5.** Cumulative Reward Comparison: TVC-only vs. TVC with CGT

Figure 5 presents the cumulative reward obtained by the reinforcement learning agent during training with two control configurations: TVC+CGT (Thrust Vector Control and Cold Gas Thrusters) and TVC Only. As evident in the graph, the agent utilizing both TVC and CGT consistently achieves a higher cumulative reward throughout the training compared to the agent with TVC alone. This initial observation implies that incorporating Cold Gas Thrusters alongside Thrust Vector Control offers a potential advantage in accumulating reward during descent. The higher reward might be attributed to the agent’s capability to perform finer adjustments and maintain a more balanced posture using both control mechanisms, leading to a more optimal trajectory.



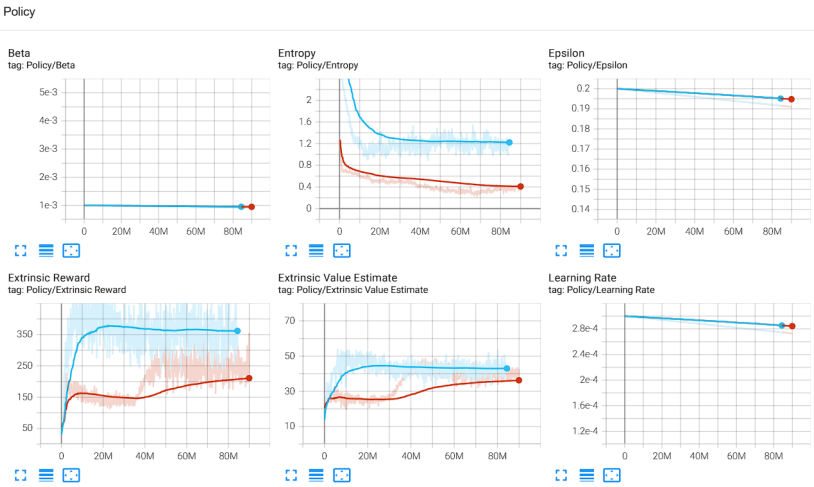
**Fig. 6.** Value Loss and Policy Loss During Training with TVC+CGT and TVC Only

Figure 6 illustrates the value loss and policy loss incurred during the training process with two control configurations: TVC+CGT (Thrust Vector Control and Cold Gas Thrusters) and TVC Only. The value loss signifies the discrepancy between the predicted and actual rewards received by the agent, while the policy

loss reflects how well the current control strategy performs in achieving the landing objective.

As observed in the graph, the TVC+CGT configuration consistently exhibits lower value loss compared to TVC Only throughout the training run. This pattern suggests that the value function for the TVC+CGT agent is more accurate in predicting the rewards associated with utilizing both control mechanisms for maintaining rocket stability during descent.

In contrast, the policy loss curves for both configurations seem to converge over time. This initial convergence might indicate that both control strategies eventually lead the agent to learn policies that achieve some level of control. However, the continued lower value loss for TVC+CGT suggests that the agent using both TVC and CGT might be acquiring a more efficient policy for accumulating reward, potentially leading to a more optimal landing trajectory in the long run.



**Fig. 7.** Policy Evaluation Metrics: Beta, Entropy, Epsilon, and More TVC with CGT (Blue color) TVC-only (Red color)

In our exploration of detailed policy dynamics, we delve into specific metrics that unravel the intricacies of the Proximal Policy Optimization (PPO) algorithm during the training of our rocket agent as shown in Figure 7. These metrics provide a nuanced understanding of the learning process, shedding light on stability, convergence, and performance enhancements.

## 8 Conclusion

In conclusion, our AI-guided rocket landing project represents a substantial step forward at the intersection of artificial intelligence and aerospace engineering. Leveraging the Proximal Policy Optimization (PPO) algorithm, we have made notable progress in training a rocket agent for autonomous navigation with the goal of precise landings on a designated pad.

The meticulous modeling of the rocket environment, incorporating real-world physics and control mechanisms, laid the foundation for a robust reinforcement learning system. The integration of Thrust Vector Control (TVC) and Cold Gas Thrusters (CGT), along with accurate observations, contributed to a comprehensive representation of the rocket's dynamics. The reward system, emphasizing distance, rotation, and altitude, has been designed to shape the agent's behavior, fostering controlled descents and improving landing precision.

Our approach to curriculum learning, incorporating randomized initial conditions and varied starting altitudes, showcased the adaptability of the trained agent to diverse scenarios. The use of multiple copies of the training area, combined with the expedited learning process using PPO, highlighted the scalability and efficiency of our methodology.

With one of our primary objectives being the improvement of TVC movement to be more free and realistic within its constraints, our intention is to facilitate the seamless transfer of this model to real-world hardware. This endeavor aligns with our broader vision of contributing to the development of autonomous rocketry that can be effectively applied in practical aerospace scenarios.

As we continue our pursuit of autonomous rocket landings, addressing challenges and refining our approach will be crucial for future success. This ongoing research sets a solid foundation for the continuous evolution of AI-guided rocketry, marking a promising trajectory towards the future of space exploration.

## References

- [1] Alireza Naderi Akhormeh et al. "Conceptual Design and Simulation of Cold Gas Thrusters as Wearable Fall Arresting Devices". In: *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2023, pp. 208–214.
- [2] Fergus William Downey. "Progress Towards Controlled Re-entry and Recovery of CubeSats". PhD thesis. Curtin University, 2023.
- [3] Martin Elvis, Charles Lawrence, and Sara Seager. "Accelerating astrophysics with the SpaceX Starship". In: *Physics Today* 76.2 (2023), pp. 40–45.
- [4] Reuben Ferrante. "A robust control approach for rocket landing". In: *Master's thesis* (2017).
- [5] M Jiang et al. "Ballistic Simulation and Analysis of Turntable Rotation on the Flight Stability of Anti-aircraft Rocket". In: *Journal of Physics: Conference Series*. Vol. 2460. 1. IOP Publishing, 2023, p. 012037.

- [6] Shengbo Eben Li. “Deep reinforcement learning”. In: *Reinforcement Learning for Sequential Decision and Optimal Control*. Springer, 2023, pp. 365–402.
- [7] Thomas M Moerland et al. “Model-based reinforcement learning: A survey”. In: *Foundations and Trends® in Machine Learning* 16.1 (2023), pp. 1–118.
- [8] Octavio Piñal and Amadeo Arguelles. “Mixed reality and digital twins for astronaut training”. In: *Acta Astronautica* (2024).
- [9] Ashwinkumar Rathod et al. “Rocket Landing Using Reinforcement Learning in Simulation”. In: *International Journal of Creative Research Thoughts (IJCRT)* 11.5 (2023), f682–f686. ISSN: 2320-2882.
- [10] John Schulman et al. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).
- [11] Blake Stuart and Jesse McEnulty. “Overview of the SLS Core Stage Thrust Vector Control System Design”. In: *45th Annual AAS Guidance, Navigation and Control (GN&C) Conference*. AAS 23-152. 2023.
- [12] Lizhen Wu et al. “Deep Reinforcement Learning with Corrective Feedback for Autonomous UAV Landing on a Mobile Platform”. In: *Drones* 6.9 (2022), p. 238.
- [13] Shuai Xue et al. “Research on Intelligent Control Method of Launch Vehicle Landing Based on Deep Reinforcement Learning”. In: *Mathematics* 11.20 (2023), p. 4276.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

