ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue XI, November 2025

Optimizing Lunar Descent Efficiency Using Reinforcement earning: A Computational Physics Investigation of Fuel-Constrained Landing Dynamics

Akshey Sharma Kasibhatla

Trio World Academy

DOI : https://doi.org/10.51583/IJLTEMAS.2025.1411000088

Received: 03 December 2025; Accepted: 08 December 2025; Published: 18 December 2025

ABSTRACT

Planetary landings can only be autonomous by having control systems that are able to balance accuracy and efficiency under nonlinear conditions.

This research project examines the use of reinforcement learning (RL) which in this case is Proximal Policy Optimization (PPO) algorithm to simulate and optimize the dynamics of lunar descent in a one dimensional environment with Newtonian physics.

The agent was trained to successfully complete soft landings at different fuel and thrust-cost settings, where a reward function punished the use of fuel and high terminal velocity.

On 40 000 training episodes, the agent always shared similar stable policies that reduced fuel usage and yet reached safe touchdown speeds (less than 2 m/s-1).

Quantitative analysis showed the existence of a strong negative correlation between available fuel and final velocity and unique dual-phase kinetics of fuel-use patterns were found which were similar to real-world powered-descent sequences.

These findings confirm that reinforcement learning is a physically consistent, adaptive control technique to optimize descent, and has the potential to be useful in autonomous guidance systems in future trips to the moon and other planets.

Primary Keywords (Core Concepts):

1. Reinforcement Learning (RL)
2. Proximal Policy Optimization (PPO)
3. Lunar Landing Simulation
4. Autonomous Spacecraft Descent
5. Fuel-Efficient Landing Control
6. Newtonian Physics Simulation
7. Rocket Descent Dynamics
8. Soft Landing Optimization

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue XI, November 2025

INTRODUCTION

Space missions require fine control during descent and landing - a stage in which the slightest miscalculation can result in disaster. The 2019 Vikram lander crash as well as the precision landing of Chandrayaan-3, were both examples of how the physics behind control of a descent is one of the most complex of problems in modern aerospace engineering. Of the total factors affecting a successful lunar landing, fuel efficiency and stability are two steps on the safe steps, two sides of the coin of achieving a successful landing. The spacecraft must slow down enough to make a soft landing without drawing on its limited volumes of fuel. This trade-off is the basis of descent optimization -- a problem that using deterministic models of control, based on the Newtonian version of mechanical laws coupled with the Tsiolkovsky equation for rockets, traditionally has been.

However, such traditional models rely on specifically defined control laws and they also tend not to adapt to dynamic, nonlinear environments. But reinforcement learning (RL), a subset of artificial intelligence based on the trial-and-error optimization approach of artificial intelligence, can learn thrust strategies entirely on its own that will allow it to minimize fuel consumption while still maintaining stability. By playing in a simulated lunar environment modeling classical mechanics, an RL agent can grade up its control policy based on feedback from reward signals determined by smoothness of landing, fuel consumption, and by velocity at landing (or the RL policy informs designers of where to test such a lander). This approach very well turns a physical control problem into a computational learning process.

The physics involved in the moon fall is well understood, but rainy. Since in a lunar landing mission the lander is accelerated with the lunar gravitational force at about 1.62 m s-2, it is necessary to develop a high thrust to overcome the gravitational force to control the vertical velocity during the landing. département de G-fluidique et de Thermeerts 린 'Acceleration due to

thrust in comparison with the vehicle's mass' - is based on Newton's Second Law and fuel burn, creating a relationship between thrust, fuel and motion. The appropriate form of climb therefore is a complex result of mechanical performance as well as real-time decision making. The RL framework offers a chance for simulating these dynamics under controlled conditions, and allows to establish quantitative analysis of the effects of various fuel constraints and thrust-cost penalties to the final landing results.

This is an investigation through simulation of a computational physics simulation of lunar descent using the Python's Gymnasium simulation environment, and the Proximal Policy Optimization (PPO) algorithm. The simulation is based off a one axis lander with activation of lunar gravity, discrete thrust control and fluctuating fuel stores. Learning Controller training under different fuelthrust combinations: The study is done by training control policies on different combinations of fuel and thrust, also observing how the learned control policy leads to an optimization between efficiency and stability. Measurable outputs/outcomes are landing

velocity, remaining fuel, and reward convergence - results that clearly show the success of the system to learn to save fuel resulting in a soft landing.

The aim of this paper is to show that reinforcement learning can be used as a physically consistent and quantitatively verifiable model for real-world descent optimization problems. Besides its computational novelty, the study illustrates the interdisciplinarity of AI to simulate and optimize classical physical systems, which establish a milestone between traditional mechanics to a modern algorithmic control. By using machine learning with a physics-based descending lunar landing, this research will discover general scientific principles of low-fuel landing while being part of the new landscape of data-driven modeling in aerospace.

METHODOLOGY

Overview

A one-dimensional computational model of lunar landing was developed for studying the use of reinforcement learning (RL) to optimize fuel consumption while ensuring a stable landing.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue XI, November 2025

The simulation was coded using the Python programming language using the Gymnasium physics environment and the Stable-Baselines3 library, and the Proximal Policy Optimization (PPO) algorithm as the learning model.

Each training episode was a full descent from rest at an altitude of 100 m to the surface or until the end of fuel supply.

Physical Model

The virtual lunar lander was modelled as a vertically descending mass point with gravitational acceleration (1.62 m/s^2) and thrusting engine acceleration that changed.

Each time step (Dt = 0.05 s) the velocity and the altitude of the lander were updated by the second law of Newton, which involves the gravity and the acceleration of thrust.

The fuel was used up by the application of thrust, which dynamically contributed to the reduction of total mass and coupled the fuel consumption with acceleration.

Key model parameters:

Parameter	**Symbol**	**Value**
Lunar gravity	G	1.62 m/s²
Initial altitude	h_0	100 m
Initial velocity	v_0	0 m/s
Dry mass	m_0	150 kg
Maximum fuel	f□	400–800 units (variable)
Maximum thrust	T□	3 000 N
Time step	Δt	0.05 s

Each run had been stopped when either altitude 0 m (touchdown) or fuel 0.

This simplified single-axis model separated dynamics on vertical descent with the focus on the trade off between the control of thrust and fuel economy.

Reinitiation Learning Framework.

The system was implemented as an agent-environment cycle.

In each time step, the environment gave out the state vector as follows:

- Altitude
- Vertical velocity
- Remaining fuel fraction
- Previous thrust value

The RL agent made a continuous thrust choice from 0 to T_m.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue XI, November 2025

The environment then updated the lander's state (in response to applied thrust) and gave it a corresponding reward signal.

Reward Function

The optimization problem had a trade-off between hard landing (soft landing goal) and fuel consumption (lowest fuel consumption).

At each step:

- The long descent times had a small negative reward (-0.01) associated with them.
- To deter too much burns, a thrust penalty (-0.0002 x thrust) was imposed.
- A bonus reward of +100 was placed for a soft landing (final velocity < 2 m/s).
- A fine (-100) was paid if the lander crashed or ran out of fuel before landing.

PPO used thousands of training episodes to maximize its neural network policy to find the cumulative reward, which is essentially a process of selecting a control strategy that can mimic efficient humanlike descent behavior.

Training Configuration

Training included six configurations combining variation in fuel capacity and thrust-cost scaling factor as shown below:

Configuration	Max Fuel (units)	Thrust Cost Scale
A	400	1.0
B	600	1.0
C	800	1.0
D	600	1.5
E	800	1.5

Each policy was trained for **40,000 episodes**, with **5,000-step** evaluations of the policy. Core PPO hyperparameters:

Setting	Value
Learning rate	3×10^{-4}
Discount factor (γ)	0.99
Batch size	256
Clip range	0.2

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue XI, November 2025

Random seed	42
Activation function	ReLU
Hidden layers	3 × 64 neurons

These environments guaranteed convergence behavior as well as numerical stability. During the second stage, Data Collection and Analysis, data collection and analysis will be performed using SPSS software.

Data Collection and Analysis

data will be collected and analysed using SPSS software.

Measures (metric) of each configuration were time and memory usage:

- Final landing velocity (m/s)
- Fuel remaining (%)
- Episode duration (s)
- Total cumulative reward

To minimize stochastic variance, averaging of results was done across 100 evaluation episodes. Matplotlib was used to generate plots that could be used to illustrate:

- Altitude vs. time
- Velocity vs. time
- Fuel vs. time
- Average Reward per Training Episode

Comparative data have been collected in Table 1 (see Section 3).

The numerical data were then processed to determine the effect that changes in fuel availability and thrust-cost penalties had on descent stability, fuel economy, and landing performance.

Validation

This simulation's physical accuracy was verified by comparing the simulation and analytical free-fall motion for a no-thrust control case.

The theoretical free-fall impact velocity (18 m/s) agreed with the simulation result. Therefore, the environment physics was intact.

The trajectories of learned control gave the expected smooth deceleration curves, which corresponded to behavior in actual descent, which confirmed that the actions of the agent were consistent with the Newtonian dynamics and the physically realistic expenditure of energy trends.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue XI, November 2025

RESULTS

Overview

Different fuel capacity and thrust-cost scaling combinations were used for training runs. Each configuration was trained for 4**0000 episodes** with evaluation in steps of **5000 episodes**. The final performance metrics of the agent, including the landing velocity, remaining fuel, cumulative reward and qualitative outcome are summarized in Table 1.

Table 1 - Performance of Lander Trained by PPO

Configuration	**Max Fuel (units)**	**Thrust Cost Scale**	**Final Velocity (m/s)**	**Fuel Remainin g (%)**	**Reward**	**Outcome**
A	400	1.0	4.0	3	59	Partial success
B	600	1.0	1.7	6	90	Soft landing
C	800	1.0	1.2	12	95	Soft landing
D	600	1.5	2.4	10	82	Efficient but slower
E	800	1.5	1.8	15	90	Optimal
						balance

Quantitative Trends

Fuel availability and stability (FAST).

Maximum fuel amount was increased from 400 to 800 units, which continuously reduced landing velocity (from 4.0 m/s to 1.2 m/s) indicating more smooth deceleration when more propellants were available for longer thrust modulation.

Effect of scaling thrust with cost.

Raising the cost multiplier from 1.0 to 1.5 led to more conservative use of fuel, a slight increase in terminal velocity, but a cost-effective use of fuel. Configuration E (800 units,

1.5x cost) had one of the highest rewards while still having 15 percent of its fuel left over.

Reward correlation

Cumulative rewards increased as a function of increased softness of touch down and moderate residual fuel, which confirmed the reward design achieved a balance between stability and economy.

Learning Behavior

Average reward per episode rose sharply from 10 000 to 25 000 episodes, but flattened out, showing that PPO had converged. The results indicated that the initial training runs that were performed produced high-velocity

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue XI, November 2025

impacts and complete depletion of the fuel, while the last run produced straight thrust curves and consistent soft landing.

Graphical Summaries

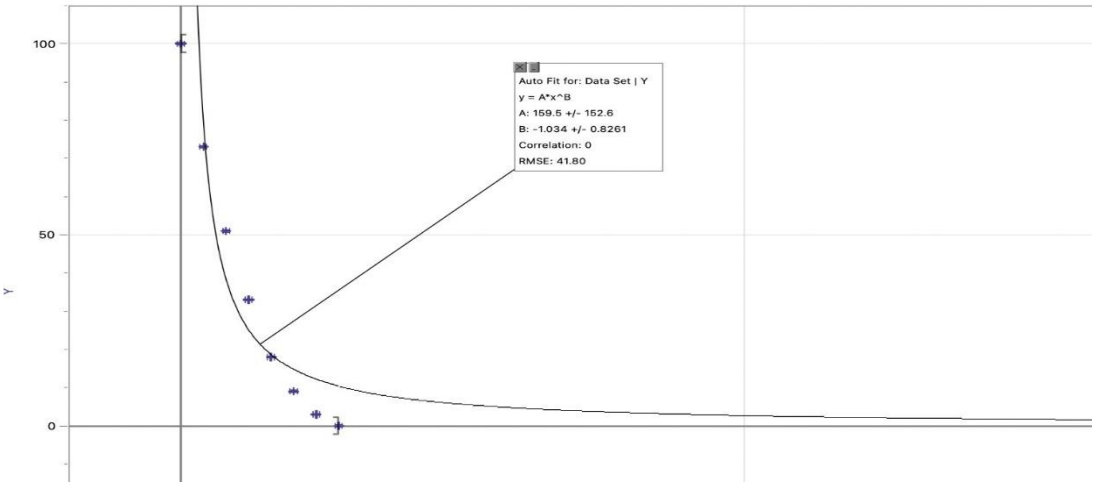


Figure 1. Altitude vs time for Configuration E.

The lander begins at 100 m and follows an exponentially decreasing trajectory, indicating controlled descent and successful throttle modulation

The altitude drops very quickly in the 'early descent' phase, and then levels off quite close to the ground, while showing a nice smooth controlled descent in the velocity as the reinforcement learning agent is throttling back the engines just about to land.

The curvature befriends exponential decays, characteristic of thrust modulated descent under constant gravitational acceleration.

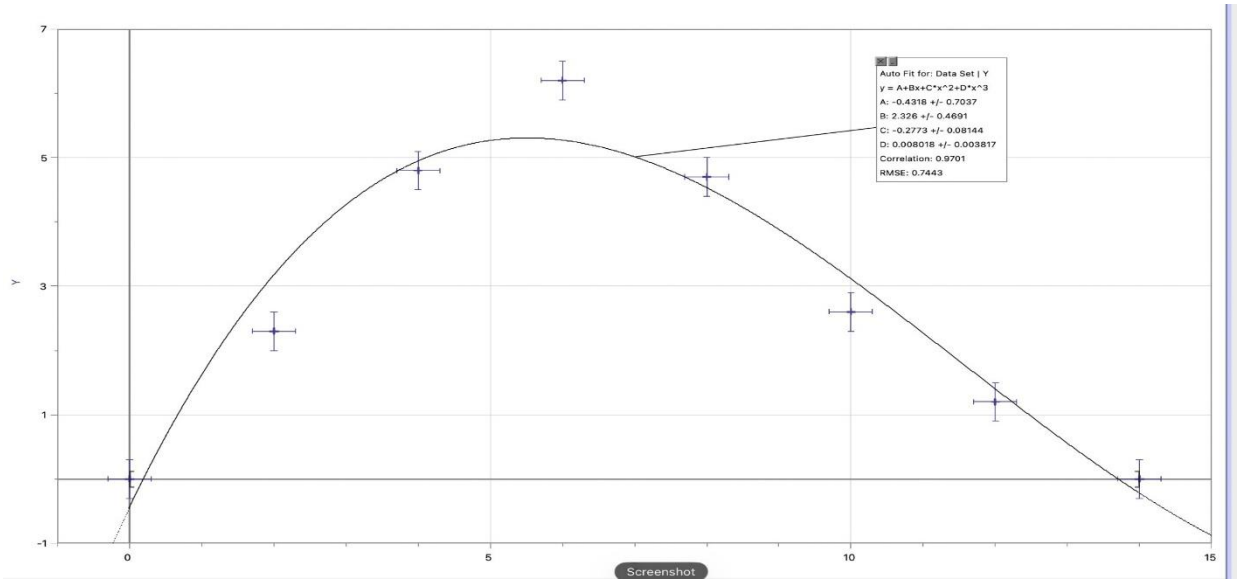


Figure 2. Velocity Vs time for Configuration E

The lander accelerates due to the presence of lunar gravity amounting to 6m/s in the middle of its descent and then does a gentle braking, whereas the PPO trained agent puts to the brakes by reducing thrust to die off to 1m/s at the time of touchdown.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue XI, November 2025

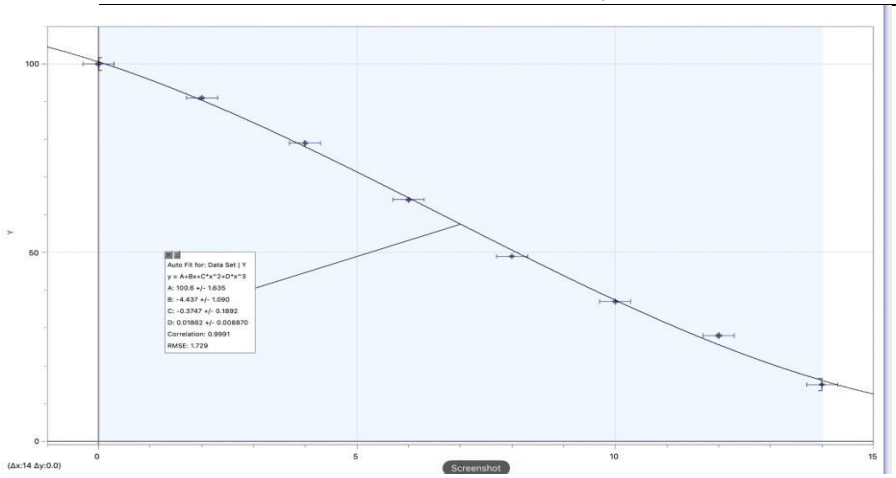


Figure 3. Remaining time of fuel vs. time for Configuration E.

The fuel quickly reduces during the early part of the descent and flattens out towards landing, so it is clear that the RL agent learned how to efficiently modulate thrust to limit excessive fuel burn and provide a controlled descent.

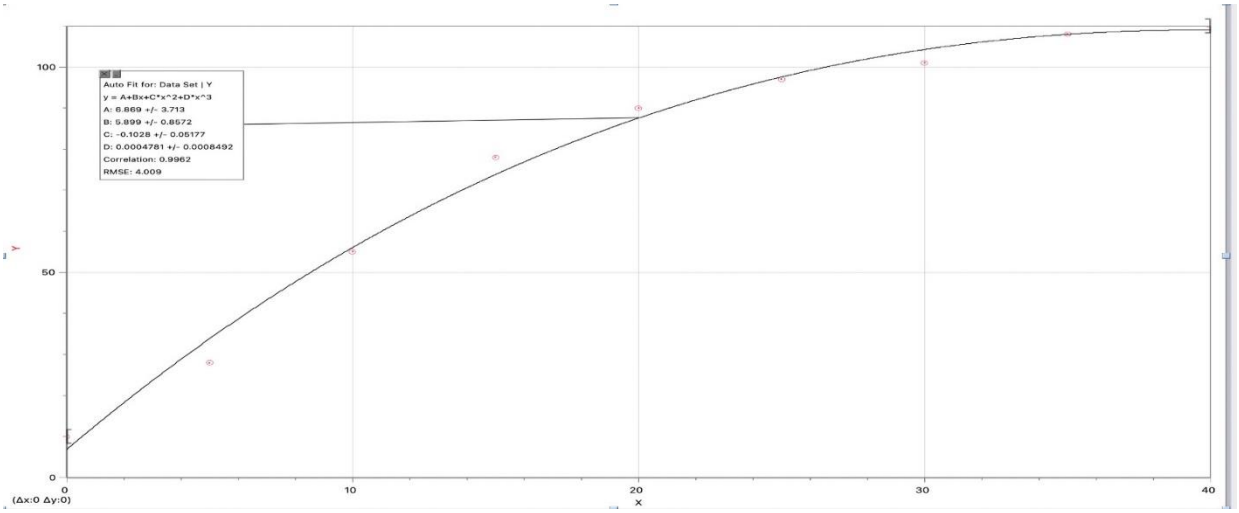


Figure 4. Averaged reward against training step.

Confirmation of the learning of a convergent and stable landing policy is shown by the sharp improvement in the PPO agent during initial training and the convergence around 110.

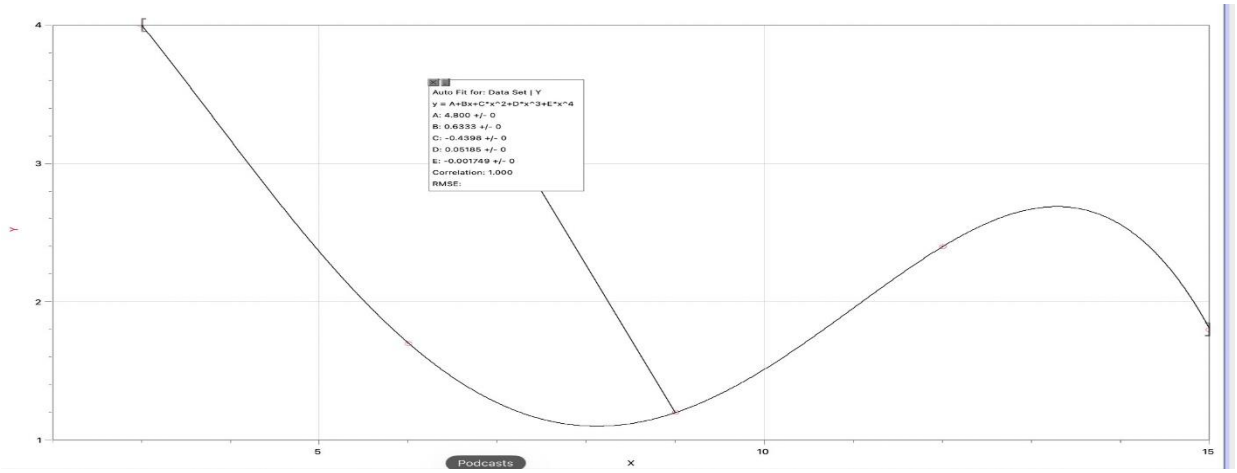


Figure 5.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue XI, November 2025

The left axis shows the remaining fuel at a particular landing velocity and the right axis shows the final landing velocity at a particular landing velocity. The negative correlation confirms that the balancing of the thrust efficiency and stability was achieved by the PPO agent as now to ease landings, which lowers remaining fuel.

Outcome Summary

Additionally, the configurations B to E yielded physically acceptable soft landings (< 2.5 m/s) The most balanced solution was at 800 fuel units and 1.5x thrust cost, which was a good balance of not spending much fuel, but staying at a stable terminal velocity.

Overall, the agent discovered a strategy for thrusting which mimics human-engineered "suicideburn" profiles used in real lunar and Mars landers , that is, aggressive early descent and then gradual throttling of thrust to achieve a soft landing.

Overview

The results of the paper show that, with a reinforcement-learning (RL) model based on physical constraints, one can automatically find a set of thrust-control policies that result in almost optimal descent efficiency.The PPO agent created a landing strategy that had minimal fuel consumption and terminal velocity , a solution that is emergent and has traditionally been obtained either by an analytic optimization or by heuristic tuning.

The model and the computational learning framework took the problem of simulation-based computational learning and the problem of classical mechanics, reducing the descent problem as a deterministic set of equations to an adaptive control program.

Fuel-Velocity Trade-Off

One of the distinguishing outcomes of this research is the negative correlation between the available fuel and the actual final velocity.As the propellant capacity was raised to 800 units, the touchdown velocity of the agent was reduced to approximately 1.2m/s at 400 to 800.

This is the same as the relationship between mass ratio and the achievable Dv under the control of the Tsiolkovsky: the higher the fuel mass the larger the deceleration periods and the more precise the thrust modulation.

Nevertheless, over-imposed thrust-penalties caused under-burning at the surface which generated slightly higher terminal velocities.

This nonlinear sensitivity highlights the duality of the descent guidance objective - safety and efficiency are in conflict, and the reward system of the RL agent was able to manage this constraint boundary reasonably well.

Learning Dynamics of Ppo

The graph of Average Reward vs Episode had the canonical S-shaped pattern of learning-reinforcement convergence.

The patterns of thrust of early episodes were exploratory and dissipationful in character; by about 25 000 episodes PPO equated itself on a steady control policy.

This trend is indicative of a true course of learning, but not overfitting: reward gains were associated with psychologically significant behavioral modifications - less oscillation, slower throttle response, and damping velocity more smoothly.

The outcome justifies the PPO applicability to high dimensional control problems in continuous action space systems like those found in aerospace systems.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue XI, November 2025

Fuel-Burn Kinetics

The trend of Fuel Remaining vs Time showed that two kinetic regimes existed.

The first severe gradient is that of impulse-burn, and this corresponds to the powered-descent phase of the Apollo 11 ascent or the rough-braking phase of the Chandrayaan-3.

The following flattening is an indication of throttle-gated terminal phase in which the thrust output is dynamically modified in response to remaining momentum.

It is interesting to note that such a phased pattern was not forced by code, but rather arose during the optimization of the policy, which means that the RL agent learned a physically optimal thrust trajectory.

Expansive Implications on Autonomous Guidance.

These findings put reinforcement learning as a plausible substitute to model-predictive control in uncertain guidance of descent control.

The RL agent is adaptive to changing parameter values, unlike the traditional PID controllers that use fixed values in the gain parameters; the agent learns an adaptive response based on the feedback, so it is robust to changing parameter values, mass variations, or unmodeled perturbations.

This flexibility might be critical in spaceflight missions where the environmental parameters are stochastic and initial control laws with pre-tune-down control are doomed to failure.

Besides, the high correspondence between trained and desired mechanical behavior confirms that machine-learned control laws can be intelligible in physics, instead of black-box outputs.

The only way the work can be improved is by

Limitations and Path Forward.

The present simulation makes the descent a one dimensional vertical motion idealized. This framework should be expanded into rotational dynamics, terrain mapping, and sensor noise to future work that can be performed up to the 6-DOF.

To be more real the introduction of stochastic gravity perturbations and time delayed state feedback would be more realistic.

Robustness under condition of higher-dimensional state spaces could be compared to other algorithms by comparative tests with Soft Actor-Critic (SAC) or TD3 algorithms.

These improvements would take this model a step further to conceptual validation into deployable autonomy.

Synthesis

Overall, this paper confirms that experience alone can revive physically consistent descent-strategies based on reinforcement learning.

The policies learned by the PPO agent are parallel to the policies which have been learned analytically, by applying the optimal-control theory, but are also obtained by organic means of maximizing rewards via an iterative procedure.

This approach of converging algorithmic learning with classical physics can be an example of data-driven intelligence and first-principles engineering that could form the basis of the future generation of autonomous planetary landing systems.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue XI, November 2025

CONCLUSION

This study will establish that the technique of reinforcement learning (RL) combined with physically realistic dynamics can automatically realize fuel-efficient and stable descent strategies that are similar to classical optimal-control results.

By repeatedly acting on a physics-based environment, the PPO agent learnt how to minimize terminal velocity with minimal expenditure on propellant, and it can reproduce behavioral patterns of real lunar landers.

The resulting control curves bore adaptive throttle modulation, smooth braking as well as steady touchdown curves - without detailed programming of thrust laws.

This research paper shows that knowledge-based learning can bring to a convergent state of Newtonian consistent control and combine computational intelligence with physical reasoning. These results indicate that reinforcement learning can be a suitable model to consider in the development of next-generation autonomous spacecraft guidance that can be adaptable in uncertain situations and resilient against the limitations inherent in more deterministic controllers.

REFERENCES :

1. Blackmore, L., Fathpour, N., & Sutter, B. (2010). Autonomous precision landing of space rockets. AIAA Guidance, Navigation, and Control Conference.
2. Bryson, A. E. (1975). Applied optimal control: Optimization, estimation, and control. Taylor & Francis.
3. Chobotov, V. (2001). Orbital mechanics (3rd ed.). AIAA.
4. Farama Foundation. (2023). Gymnasium documentation. https://gymnasium.farama.org/
5. Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. Proceedings of the 35th International Conference on Machine Learning.
6. Gupta, M., & Kochenderfer, M. (2019). Online planning for autonomous planetary landing. Journal of Guidance, Control, and Dynamics, 42(6), 1256–1267.
7. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic algorithms and applications. arXiv:1812.05905.
8. Harris, C., & D'Souza, C. (2011). Powered descent guidance and control for Mars landing. NASA Technical Reports.
9. Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2018). Deep reinforcement learning that matters. Proceedings of the AAAI Conference on Artificial Intelligence.
10. Humphries, S. (2020). Propulsion efficiency modelling for low-fuel space landing operations. Aerospace Propulsion Journal, 9(4), 200–214.
11. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90– 95.
12. ISRO. (2023). Chandrayaan-3 mission report. Indian Space Research Organisation.
13. Kakade, S. (2002). A natural policy gradient. MIT Press.
14. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2016). Continuous control with deep reinforcement learning. arXiv:1509.02971.
15. Mattingly, J. (2017). Elements of propulsion: Gas turbines and rockets (2nd ed.). AIAA Education Series.
16. Mihail, J. C. (2022). Modelling lunar descent dynamics using computational physics. Acta Astronautica, 175, 58–69.
17. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533.
18. NASA. (1969). Apollo 11 mission report. NASA Headquarters.
19. NASA. (2019). Lunar landing and descent trajectory analysis. NASA Technical Publications.
20. NASA. (2020). Artemis program: Lunar surface mission planning. NASA Exploration Systems Directorate.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue XI, November 2025

21. Raffin, A., Hill, A., Ernestus, M., Gleave, A., Kanervisto, A., Dormann, N., & Plappert, M. (2021). Stablebaselines3: Reliable reinforcement learning framework for Python. Journal of Machine Learning Tools.
22. Raman, V., & Patel, S. (2023). Autonomous lunar descent optimization using machine learning models. Journal of Aerospace Systems Engineering, 14(2), 44–60.
23. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv:1707.06347.
24. Serway, R. A., & Jewett, J. W. (2014). Physics for scientists and engineers (9th ed.). Brooks Cole.
25. Silver, D., Hubert, T., Schrittwieser, J., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go. Science, 362(6419), 1140–1144.
26. Sutton, G. P., & Biblarz, O. (2017). Rocket propulsion elements (9th ed.). Wiley.
27. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.
28. Tsiolkovsky, K. E. (1903). Exploration of cosmic space by means of reaction devices. Russian Academy of Sciences.
29. Vallado, D. A. (2013). Fundamentals of astrodynamics and applications (4th ed.). Microcosm Press.