TINTIN: Thrust-Integrated Neural Touchdown with Reinforcement Learning and INertial Navigation

Qilong (Jerry) Cheng[†] N18422055, Juncheng Zhou[†] N16975306

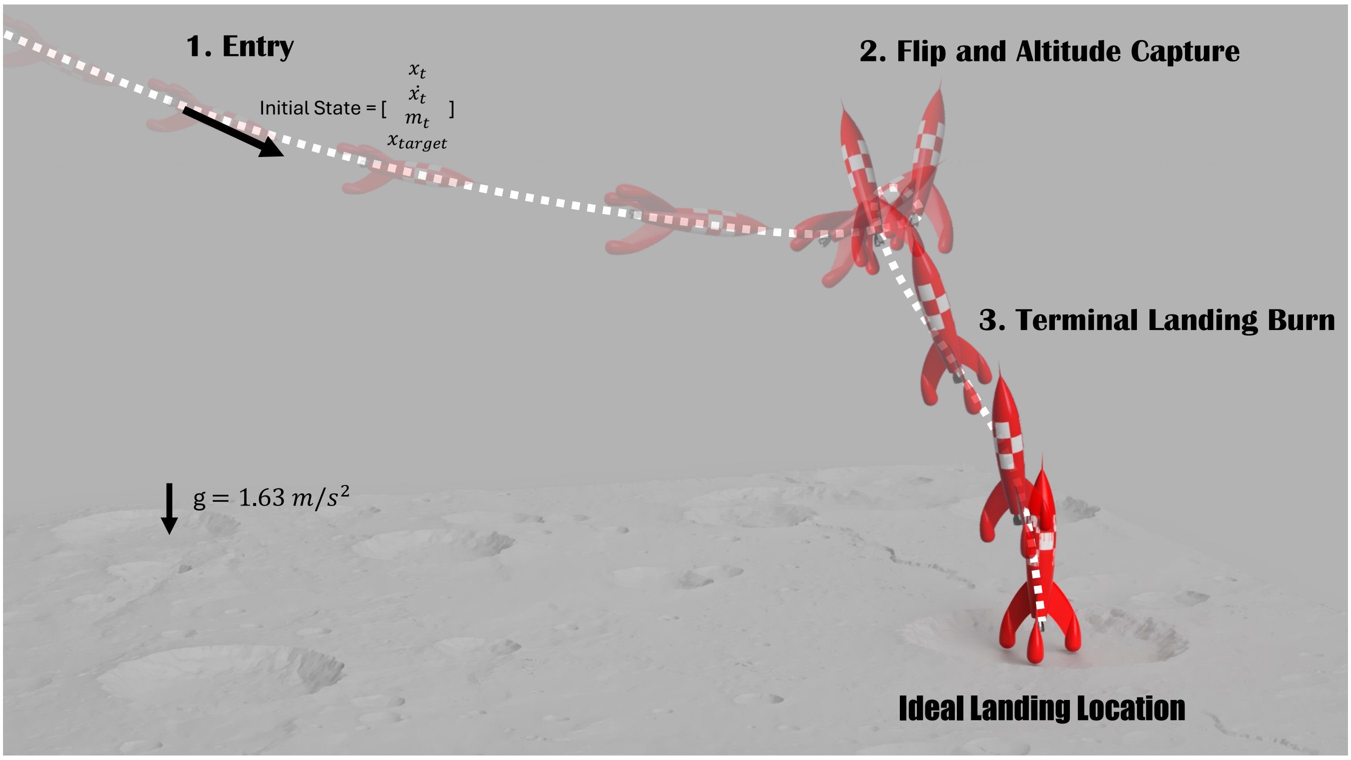
1. Entry
Initial State = [x_t $\dot{x}_t$ m_t x_{target}]
2. Flip and Altitude Capture
3. Terminal Landing Burn
g = 1.63 m/s^2
Ideal Landing Location

Fig. 1. System overview of the lunar rocket-landing pipeline. The trajectory comprises three phases: (1) *Entry*, (2) *Flip & attitude capture*, and (3) *Terminal landing burn* to a pre-specified target pad. An RL policy observes vehicle pose/velocity relative to the target and outputs throttle plus 2-DoF gimbal commands (pitch, yaw) to achieve pose-accurate touchdown while minimizing propellant under lunar gravity ($g \approx 1.62 \text{ m/s}^2$).

I. Introduction

Lunar exploration has been deserted since the 1980s after the Apollo mission due to its high cost, and low profitability. Sustained lunar spacecrafts are to reduce cost and logistics, making lunar explorations possible again. Modern vertical-takeoff–vertical-landing (VTVL) systems on Earth has gained noticeable success in the past decade, notably SpaceX's first-stage returns, illustrate the economic and operational impact of rapid reuse [1]. Translating that capability to the Moon raises distinct challenges. The Moon's environment largerly differs from the Earth, with reduced gravity ($g \approx 1.62 \text{ m/s}^2$), plume–regolith effects, uncertain local slopes and rocks, and sensing constraints under high-contrast illumination. In addition, due to the lack of infrastructure on the Moon, GPS, radar towers are missing, causing the challenge in localization when landing. Those unique challenges on the Moon complicate classical guidance–navigation–control (GNC) stacks built from hand-tuned guidance laws, estimators, and cascaded PID/LQR controllers [2], [3].

Motivated by this mission need, we propose using reinforcement learning (RL) for pose-targeted lunar landings with a two degree-of-freedom (DOF) gimbaled engine. In addition, this project is also largely inspired by the adventures of Tintin's rocket that first sparked our interest in spaceflight. The objective is a soft, on-pad touchdown within tight position and attitude tolerances while minimizing the fuel consumption during landing. In Project 1, we focused on the simulation modeling. We implemented a physics-based model in MuJoCo, featuring a stylized Tintin rocket and Moon-like terrain. This model allows us to simulate the

This work is supported by Professor Benjamin Rivière's course ROB-GY 7863. The document is served as the Project Report 1

† denotes equal contribution.

rocket thrust control, analyzing the rocket's pose, and velocity changes, and the relative position to the target pose. This simulator also provides the environment for RL policy development (Project 2), and it allows us to evaluate dynamical modeling choices and establish baselines for classical versus learned control.

RL is attractive here for three reasons. First, by directly commanding throttle and gimbal angles, a learned policy can co-shape translational and rotational dynamics, mitigating coupling that is cumbersome to tune in cascaded architectures. Second, training with domain randomization (gravity level, terrain heightfields, sensor noise) targets distributional robustness to off-nominal conditions [4], [5]. Third, multi-objective rewards naturally trade pad accuracy, attitude alignment, touchdown softness, and propellant usage without redesigning the control structure.

To sum up, there are three main contirbutions in this project: (i) we developed a MuJoCo-based lunar-landing simulator with a gimbaled-thrust Tintin vehicle and heterogeneous regolith terrain; (ii) defined pose-targeted metrics (position/attitude error, terminal vertical/horizontal velocities) and disturbance suites for validation; and (iii) compare a classical PID control baseline to motivate RL-based control for Project 2. This modeling based simulation study establishes our mission: a reproducible environment framework, and quantitative tests that validate the policy performance for learning-based control.

II. Related Work

A. Physics-Based Methods

Classical powered-descent studies model the vehicle and environment to keep guidance problems tractable rather than exhaustively physical. They normally adopt simple but dominant effects and terminal constraints [6], [7]. Translation is typically reduced to a point-mass with constant gravity, bounded thrust, and a fixed specific-impulse mass-depletion relation [2], [6]. Nonconvex thrust-direction limitations are reshaped as thrust cones and compatible magnitude bounds, enabling efficient solution methods [8], [9]. When six degrees of freedom are retained, the vehicle is modeled as a rigid body with fixed inertia and straightforward gimbal/torque/rate limits, while attitude is decoupled via a fast inner-loop assumption so translation can treat thrust direction as directly commanded [10], [11]. Terrain and contact are often not simulated. Instead, safe landing is enforced via terminal altitude, glide-slope, touchdown-speed, often with a short vertical-braking leg standing in for true contact [2], [11]. Disturbances like the wind or gravity error and sensing noise are simplified as bounded biases or white noise without explicitly modeled as the dynamic model [6], [10]. Overall, ”constraint shaping” yields convex or near-convex programs—lossless convexification, G-FOLD, and successive convexification—supporting real-time implementable guidance [8]–[12].

B. Learning-Based Methods

Learning-based guidance and control approaches have gained increasing attention in recent years, focusing on autonomous policy learning for planetary powered descent and landing. These methods often employ simplified dynamic environments for tractable training. Typically, the target reference frame is defined as an inertial frame centered at the landing site with constant gravity. The terrain is not explicitly modeled as a mesh or image, instead, touchdown is represented by an altitude condition ($h = 0$) with glideslope and attitude constraints to emulate safe terminal descent. A short vertical descent phase (below $\sim$15 m) is commonly enforced to approximate pre-contact behavior.

In terms of control formulation, both imitation policies (trained on optimal trajectories) and deep reinforcement learning (DRL) policies have been explored. Gaudet *et al.* [13], [14] demonstrated a PPO-based integrated guidance-and-control (IGC) policy in a full 6-DOF simulation environment, achieving robust Monte Carlo performance and competitive fuel efficiency. However, their environment used simplified vehicle and actuator models, assumed full-state observability, and excluded explicit touchdown contact or regolith physics—leaving open issues in sim-to-real transfer and constraint enforcement. The study primarily focused on realistic thrust–torque–mass–attitude coupling and dynamic uncertainty modeling.

Scorsoglio *et al.* [15] proposed a vision-based meta-RL framework for autonomous lunar landing that integrates image-based navigation and intelligent guidance. The environment employed a real lunar DTM from the LROC Apollo 16 site rendered via Blender/Cycles with physically based path tracing, using the DTM as a displacement map to generate realistic grayscale images. The policy learned image-to-thrust mappings via PPO and recurrent networks, showing promising adaptation to uncertain dynamics. Nevertheless, the dynamics were limited to 3-DOF without explicit ground contact or plume effects, making transfer under illumination changes, dust obscuration, and sensor latency still challenging.

III. Methods

A. Analytical Modeling

We first model a rigid single-body TINTIN rocket with four frames: the world/inertial frame $\mathcal{F}_W$, the rocket body frame $\mathcal{F}_B$ fixed at the center of mass (CoM), the thrust/nozzle frame $\mathcal{F}_T$ attached at the bottom of the physical nozzle, and a target frame $\mathcal{F}_{\text{tar}}$ fixed to the desired landing location.

We define the model's states and controls as

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ q \\ \boldsymbol{\omega} \\ m \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} T \\ \theta_p \\ \theta_y \end{bmatrix}. \tag{1}$$

Here $\mathbf{p} = [x\ y\ z]^\top$ (m) is the CoM position in the world frame $\mathcal{F}_W$; $\mathbf{v} = [v_x\ v_y\ v_z]^\top$ (m/s) is the CoM linear velocity in $\mathcal{F}_W$; $q = [w\ x\ y\ z]^\top$ is the unit quaternion mapping

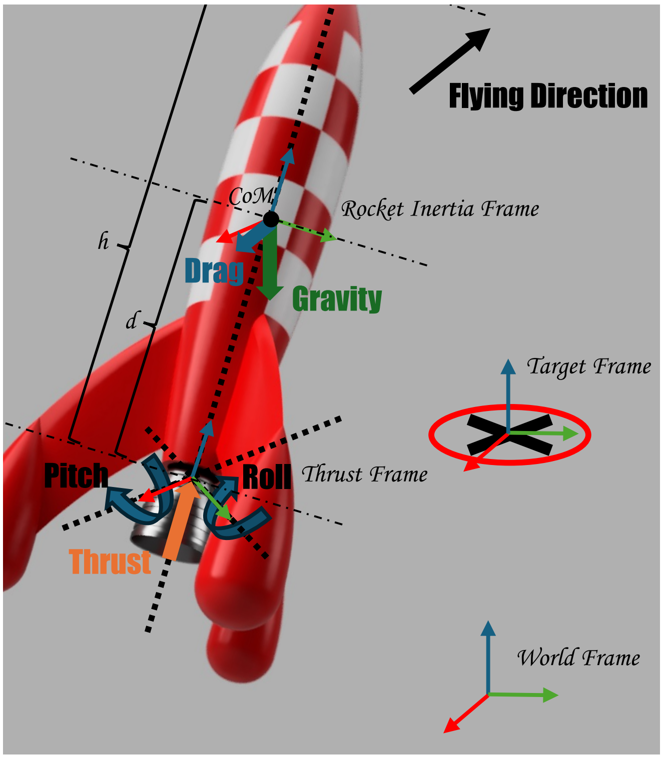

Figure 2: Coordinate frames and external forces for the rocket model: world frame, rocket inertial/body frame at the CoM, thrust frame at the gimbaled engine, and a target/landing frame for guidance. Thrust is commanded in the thrust frame and transformed to the body frame via pitch–yaw gimbal angles; drag opposes the flying direction and gravity acts in the world frame.

$\mathcal{F}_B \to \mathcal{F}_W$ and induces $\mathbf{R}_{WB}(q) \in \mathrm{SO}(3)$; $\boldsymbol{\omega} = [\omega_x \; \omega_y \; \omega_z]^\top$ (rad/s) is the body angular velocity expressed in $\mathcal{F}_B$; $m > 0$ (kg) is the vehicle mass. The control $\mathbf{u}$ consists of thrust magnitude $T \geq 0$ (N) and gimbal angles (θ_p, θ_y) (rad) defining $\mathbf{R}_{BT}(\theta_p, \theta_y) \in \mathrm{SO}(3)$.

Variable mass: We decompose the total mass as

$$m(t) = m_{\text{dry}} + m_{\text{prop}}(t), \qquad m_{\text{prop}}(t) \geq 0,$$

where m_{dry} is the fixed dry mass (structure/avionics/engine) and $m_{\text{prop}}(t)$ is the remaining propellant. With CoM fixed, propellant burns according to

$$\dot{m}_{\text{prop}} \;=\; \dot{m} \;=\; -\frac{T}{I_{sp}\, g_0},$$

and we clamp at burnout:

$$m(t) \;\leftarrow\; \max\big(m_{\text{dry}},\, m(t)\big).$$

Here T [N] is the commanded net thrust magnitude, I_{sp} [s] is the engine specific impulse, and $g_0 = 9.81 \ \mathrm{m/s^2}$ is the standard gravity constant.

For the later stability of the RL training, we maintain the inertia constant at its initial value and neglect $\dot{\mathbf{J}}$:

$$\mathbf{J} \approx \mathbf{J}(m_0) = \mathrm{diag}(J_x^0, J_y^0, J_z^0) \tag{2}$$

$$\mathbf{J}\,\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega}) = \boldsymbol{\tau}. \tag{3}$$

Kinematics modeling: The vehicle has height h, and the nozzle origin is offset from the CoM along $-\mathbf{e}_3^b$ by a distance d. Based on this kinematics relationship we can get the thrust force relative to the rocket body as

$$\mathbf{r}_{TB}^{B} = \begin{bmatrix} 0 \\ 0 \\ -d \end{bmatrix}, \tag{4}$$

$$\mathbf{R}_{WB}(q) \in \mathrm{SO}(3), \tag{5}$$

$$\mathbf{R}_{BT}(\theta_p, \theta_y) = \mathbf{R}_x(\theta_p)\mathbf{R}_y(\theta_y) \in \mathrm{SO}(3). \tag{6}$$

Here $\mathbf{r}_{tb}^{b}$ is the translation from the body origin (CoM) to the thrust-frame origin, expressed in $\mathcal{F}_b$. The height of the rocket h is used for geometry modeling, and stays constant. The nozzle position in the world is

$$\mathbf{p}_{\text{thr}}^{W} = \mathbf{p}^{W} + \mathbf{R}_{WB}(q)\,\mathbf{r}_{BT}^{B}.$$

With $\hat{\mathbf{t}}_T = \mathbf{e}_3 = [0 \;\; 0 \;\; 1]^\top$ the thrust axis in $\mathcal{F}_T$, the thrust directions in body and world are

$$\hat{\mathbf{t}}_B = \mathbf{R}_{BT}(\theta_p, \theta_y)\,\hat{\mathbf{t}}_T, \tag{8}$$

$$\hat{\mathbf{t}}_W = \mathbf{R}_{WB}(q)\,\hat{\mathbf{t}}_B. \tag{9}$$

Dynamics modeling: Under Moon environment, vacuum descent, the aerodynamics drag can be ignored. Thus the only modeled external forces are gravity applied at the CoM, thrust applied at the bottom of the throttle, and any other contact impulses generated by MuJoCo. The thrust force and its torque about the CoM are

$$\mathbf{F}_T^W = T\,\hat{\mathbf{t}}^W, \tag{10}$$

$$\boldsymbol{\tau}_T^B = \mathbf{r}_{TB} \times \big(T\,\hat{\mathbf{t}}^B\big), \tag{11}$$

$$\boldsymbol{\tau}_T^W = \mathbf{R}_{WB}(q)\,\boldsymbol{\tau}_T^B. \tag{12}$$

Let $\mathbf{F}_c^W$ and $\boldsymbol{\tau}_c^W$ denote net contact force/torque at the CoM produced by MuJoCo's constraint solver (zero when in free flight). The continuous-time equations of motion are

$$\dot{\mathbf{p}} = \mathbf{v}, \tag{13}$$

$$m\,\dot{\mathbf{v}} = m\,\mathbf{g} + \mathbf{F}_T^W + \mathbf{F}_c^W, \tag{14}$$

$$\dot{q} = \tfrac{1}{2}\,\Omega(\boldsymbol{\omega})\,q, \tag{15}$$

$$\mathbf{J}\,\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega}) = \boldsymbol{\tau}_T^B + \mathbf{R}_{BW}(q)\,\boldsymbol{\tau}_c^W, \tag{16}$$

$$\dot{m} = -\frac{T}{I_{sp}\, g_0}, \tag{17}$$

where I_{sp} is specific impulse (s), $g_0 = 9.81 \ \mathrm{m/s^2}$, and

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}. \tag{18}$$

Equations (10)–(16) implement a gimbaled thrust applied at the physical nozzle location (torque via the lever arm $\mathbf{r}_{TB}$), with gravity acting through the CoM and no aerodynamic forces. Further, we can stack the equations to get the rocket's full body dynamics as $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, where

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{v} \\ \mathbf{g} + \dfrac{T}{m}\mathbf{R}_{WB}(q)\,\mathbf{R}_{BT}(\theta_p, \theta_y)\,\mathbf{e}_3 + \dfrac{1}{m}\mathbf{F}_c^W \\ \tfrac{1}{2}\,\Omega(\boldsymbol{\omega})\,q \\ \mathbf{J}^{-1}\big(\mathbf{r}_{BT}^B \times (T\,\mathbf{R}_{BT}(\theta_p, \theta_y)\,\mathbf{e}_3) + \mathbf{R}_{BW}(q)\,\boldsymbol{\tau}_c^W - \boldsymbol{\omega} \times (\mathbf{J}\,\boldsymbol{\omega})\big) \\ -\dfrac{T}{I_{sp}\,g_0} \end{bmatrix}. \tag{19}$$

MATLAB simulation verification: We verified the 6–DoF dynamics by simulating a Moon–gravity transfer from $\mathbf{p}_A$ to $\mathbf{p}_B$ using the dynamics model above.The designed experiment objective is to transfer the rocket body from pose $\mathcal{A}$ to pose $\mathcal{B}$ in 3D. For testing, we use a simple *world-frame position PID* to generate a desired acceleration $\mathbf{a}_{\text{cmd}}$ and allocate thrust by $\mathbf{F}_T^W = T_{\text{cmd}}\hat{\mathbf{t}}_W$ with magnitude saturation $T_{\min} \le T_{\text{cmd}} \le T_{\max}$ and direction clamped by the gimbal cone. The induced lever-arm torque $\boldsymbol{\tau}_T^B = \mathbf{r}_{BT}^B \times (T_{\text{cmd}}\hat{\mathbf{t}}_B)$ excites attitude dynamics; no attitude setpoint is commanded.

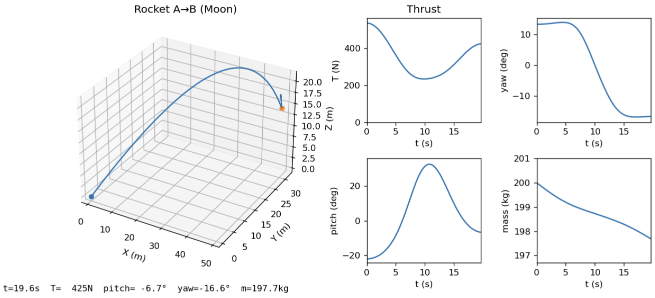

Figure 3: Closed-loop Moon transfer $\mathcal{A} \to \mathcal{B}$ using a position–PID thrust–vectoring controller (gimbal-limited, offset nozzle). Left: 3-D CoM trajectory with target. Right: time histories of thrust magnitude, yaw and pitch gimbal angles, and mass burn; the vehicle reaches $\mathcal{B}$ in ∼19.6 s within thrust and gimbal bounds.

(i) Frame and wrench consistency: applying thrust at the physical nozzle and observing the induced body torque confirms the $\mathcal{F}_W \leftrightarrow \mathcal{F}_B$ transforms and the cross-product torque mapping. (ii) Mass–thrust coupling: monotonic mass decay and the resulting acceleration trends match $\dot{m} = -T/(I_{sp}g_0)$ and $m\dot{\mathbf{v}} = m\mathbf{g} + \mathbf{F}_T^W$, validating the variable-mass terms. (iii) Actuator limits: the closed-loop motion respects $T_{\max}$ and the gimbal cone, indicating correct saturation and cone projection. (iv) Trajectory feasibility: the 3-D path converges to $\mathbf{p}_B$ within position/velocity tolerances, showing a dynamically feasible transfer without active attitude regulation.

We logged mass, 3-D trajectory, and thrust control angles over time for dynamics analysis. In the experiment run, mass decreases smoothly, the path bends as expected from offset-thrust torques, and speed evolves consistently with thrust saturation.

B. MuJoCo Simulation Modeling

To mitigate the computational cost of evaluating full analytic dynamics at high rates, we use MuJoCo as the physics integrator and contact solver. The Tintin rocket mesh is exported from STEP to OBJ in Blender. In MuJoCo the vehicle is a single rigid body with a `<freejoint/>`. The gimbaled engine is modeled as a thrust wrench applied at a body-fixed nozzle frame. Ground contact uses simple MuJoCo geoms. Lunar gravity is set to $g_{\text{moon}} = 1.62~\text{m/s}^2$. Contact parameters are tuned (friction, rolling friction, restitution) to achieve realistic touchdown deceleration without excessive rebound. At runtime we compute only the thrust wrench (force plus equivalent CoM torque) from the control $\mathbf{u}$ and the current pose, and inject it as an external wrench on the lander body.

Algorithm 1: TINTIN rocket MuJoCo per-step computation process pseudocode

```
Input: $\mathbf{x}_k = [\mathbf{p}_k, \mathbf{v}_k, q_k, \boldsymbol{\omega}_k, m_k]$, $\mathbf{u}_k = [T_k, \theta_{p,k}, \theta_{y,k}]$, $\mathbf{r}_{BT}^B$, $\Delta t$
Output: Applied wrench $(\mathbf{F}_{T,k}^W, \boldsymbol{\tau}_{T,k}^W)$, updated mass $m_{k+1}$
    1) Pose: read $\mathbf{R}_{WB}(q_k)$, $\mathbf{p}_k$; set $\mathbf{p}_{\text{thr},k}^W \leftarrow \mathbf{p}_k + \mathbf{R}_{WB}(q_k)\,\mathbf{r}_{BT}^B$.
    2) Gimbal: $\mathbf{R}_{BT,k} \leftarrow \mathbf{R}_x(\theta_{p,k})\,\mathbf{R}_y(\theta_{y,k})$;
       $\hat{\mathbf{t}}_k^W \leftarrow \mathbf{R}_{WB}(q_k)\,\mathbf{R}_{BT,k}\,\mathbf{e}_3$.
    3) Wrench: $\mathbf{F}_{T,k}^W \leftarrow T_k\,\hat{\mathbf{t}}_k^W$; $\boldsymbol{\tau}_{T,k}^W \leftarrow (\mathbf{p}_{\text{thr},k}^W - \mathbf{p}_k) \times \mathbf{F}_{T,k}^W$.
    4) Apply: data.xfrc_applied[bid,:3] $\leftarrow \mathbf{F}_{T,k}^W$;
       data.xfrc_applied[bid,3:] $\leftarrow \boldsymbol{\tau}_{T,k}^W$.
    5) Mass: $m_{k+1} \leftarrow \max\big(m_k - \frac{T_k}{I_{sp}g_0}\Delta t,\ m_{\text{dry}}\big)$.
    6) Step: mj_step; set $m \leftarrow m_{k+1}$ and read back $\mathbf{x}_{k+1}$.
```

In short, the method enforces a minimal, physically grounded lander model: rigid-body dynamics with fixed principal inertia, gravity, and a gimbaled thrust applied at the nozzle location. MuJoCo supplies contact impulses and integrates the equations of motion; our implementation computes only the thrust wrench from the control and updates the rocket's mass by the rocket equation (17). This separation keeps the simulator faithful yet lightweight for large-scale policy training.

References

[1] K. Dresia, S. Stappert, S. Karl, and M. Sippel, "Multidisciplinary design optimization of reusable launch vehicles," *Journal of Spacecraft and Rockets*, vol. 58, no. 5, pp. 1193–1209, 2021.

[2] S. R. Ploen, B. Açıkmeşe, and A. A. Wolf, "A comparison of powered descent guidance laws for mars pinpoint landing," in *AIAA Guidance, Navigation, and Control Conference*, 2006.

[3] L. Blackmore, B. Açıkmeşe, and D. P. Scharf, "Minimum-landing-error powered-descent guidance for mars landing using convex optimization," *Journal of Spacecraft and Rockets*, vol. 47, no. 5, pp. 737–746, 2010.

[4] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," *arXiv preprint*, 2017. arXiv:1703.06907.

[5] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, 2018.

[6] B. Açıkmeşe and S. R. Ploen, "Convex programming approach to powered descent guidance for mars landing," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, 2007.

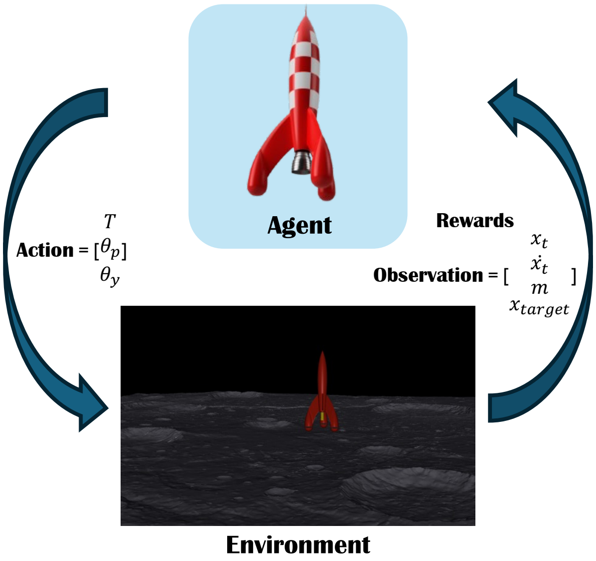
Agent
Rewards
Action = [T θ_p θ_y]
Observation = [x_t $\dot{x}_t$ m x_{target}]
Environment

Figure 4: Agent–environment loop for learning thrust–vector control for rocket landing. The policy (**Agent**) outputs actions $a_t = [\,T,\ \theta_p,\ \theta_y\,]$. The **Environment** simulates a 6-DoF variable-mass rocket on lunar terrain and returns observations $o_t = [\,x_t,\ \dot{x}_t,\ m_t,\ x_{\text{target}}\,]$ together with rewards that encourage reaching the target and soft landing while discouraging high fuel use and large gimbal movements.

[7] L. Blackmore, B. Açıkmeşe, and D. P. Scharf, "Minimum-landing-error powered-descent guidance for Mars landing using convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 4, pp. 1161–1171, 2010.

[8] J. M. C. III, B. Açıkmeşe, and L. Blackmore, "Lossless convexification of powered-descent guidance with nonconvex thrust bound and pointing constraints," in *Proc. American Control Conference (ACC)*, pp. 4224–4231, 2011.

[9] B. Açıkmeşe, J. M. C. III, and L. Blackmore, "Lossless convexification of nonconvex control bound and pointing constraints of the soft-landing optimal control problem," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2104–2113, 2013.

[10] M. Szmuk and B. Açıkmeşe, "Successive convexification for Mars 6-dof powered descent landing guidance," in *AIAA SciTech Forum, Guidance, Navigation, and Control Conference*, 2017.

[11] M. Szmuk, T. P. Reynolds, and B. Açıkmeşe, "Successive convexification for real-time 6-dof powered descent guidance with state-triggered constraints," *arXiv preprint*, 2018. arXiv:1811.10803.

[12] B. Açıkmeşe, A. A. Aňg, *et al.*, "G-fold: A real-time implementable fuel-optimal large-divert guidance algorithm for planetary pinpoint landing," in *Lunar and Planetary Science Conference (LPSC) Abstracts*, no. 4193, 2012.

[13] B. J. Gaudet, D. Furfaro, R. Linares, M. D. Lizia, and D. J. Scheeres, "Deep reinforcement learning for six degree-of-freedom planetary powered descent and landing," *arXiv preprint*, 2018. arXiv:1810.08719.

[14] B. J. Gaudet, D. Furfaro, R. Linares, M. D. Lizia, and D. J. Scheeres, "Deep reinforcement learning for six degree-of-freedom planetary landing," *Acta Astronautica*, vol. 171, pp. 24–36, 2020.

[15] A. Scorsoglio, D. Furfaro, R. Linares, *et al.*, "Image-based deep reinforcement learning for autonomous lunar landing," in *AIAA SciTech*, 2020.