TOWARD PIXEL-GROUNDED WORLD MODELS FOR POWERED DESCENT: A ROCKET LANDING BENCHMARK AND EXPERT BASELINE

Charles Duong,* Aviral Vaidya,* Aditya Iyer,* Aidan LaBella, Lucas Maes, Randall Balestriero
Department of Computer Science
Brown University
{charles_duong, aviral_vaidya, aditya_iyer}@brown.edu

ABSTRACT

World models aim to learn predictive latent dynamics for planning from high-dimensional observations, yet most closed-loop control demonstrations remain in simplified domains that do not stress realistic actuation limits or touchdown contact dynamics. We introduce **RocketLanding**, a physics-based powered-descent-and-landing benchmark built on `PyFlyt`/PyBullet, designed to stress long-horizon compounding error under constraints that preclude hovering and require precisely timed braking. The task supports pixel observations and goal images for vision-based objectives, while providing a standardized evaluation protocol based on randomized initial-condition perturbations.

We contribute a strong, interpretable state-based **analytical suicide-burn expert** with PD attitude control and gain-scheduled lateral guidance (90% success on medium perturbations and 20% on hard), and outline a JEPA-style pipeline (frozen visual encoder, action-conditioned latent dynamics, goal-conditioned MPC) intended to train on offline trajectories generated by this benchmark.

1 INTRODUCTION

Learning reliable control from high-dimensional observations remains a core challenge for deploying autonomous systems in safety-critical settings. Powered descent and landing is a canonical stress test as dynamics are strongly nonlinear, control authority is limited (throttle and gimbal constraints), and errors compound rapidly near touchdown. Classical guidance pipelines succeed by combining accurate state estimation with carefully engineered objectives and safety margins. In contrast, many learning-based control benchmarks either rely on dense reward shaping, privileged simulator states, or simplified dynamics, making it difficult to assess whether modern *world models* can genuinely support closed-loop decision-making in realistic, contact-rich regimes.

World models offer a clean alternative: learn action-conditioned predictive dynamics in a compact latent space and use the learned model for planning. We outline a JEPA-style pipeline (Fig. 1) with a frozen visual encoder, an action-conditioned latent predictor trained on offline trajectories, and goal-conditioned latent MPC that plans toward a goal image motivated by recent self-supervised visual representation learning. In this paper we do *not* report trained world-model results. Instead, we provide the benchmark, expert trajectories, and evaluation protocol needed to make this pipeline testable.

In this paper, we take a first step toward that agenda for powered descent. We introduce **RocketLanding**, a physics-based benchmark for rocket descent and touchdown that explicitly incorporates a non-hoverable regime (minimum throttle implies $T/W > 1$) and limited gimbal authority, making braking timing and the coupling between attitude and lateral correction crucial for safe landing. We provide a concise task specification in Section 2 and full details in Appendix A. To ground the benchmark with a strong reference point, we implement an interpretable **analytical suicide-burn expert** with PD attitude control and gain-scheduled lateral guidance, and we evaluate it under two

*Equal Contribution.

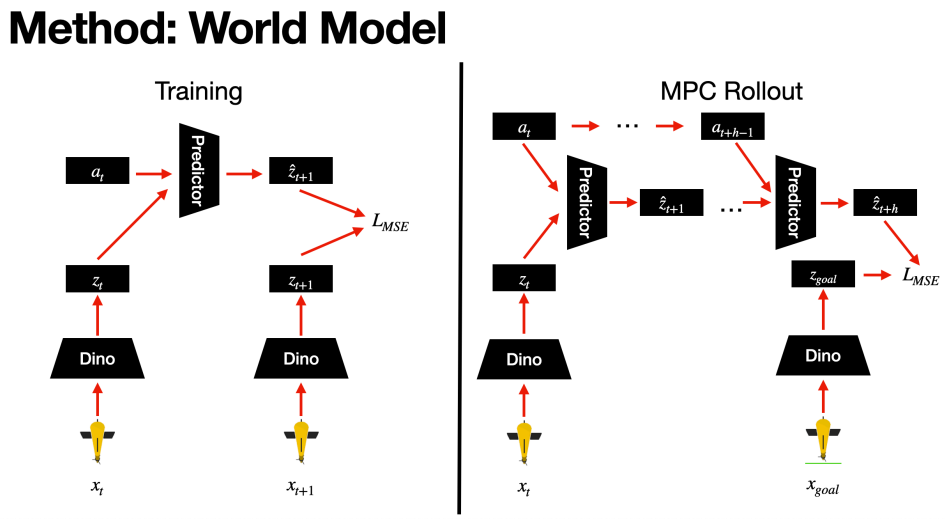

Figure 1: **Proposed JEPA-style pipeline (future work).** (Left) A frozen visual encoder maps images to latents; an action-conditioned predictor is trained to predict next-step latents from offline trajectories. (Right) At test time, goal-conditioned MPC selects actions by rolling out latent dynamics toward a goal latent computed from a goal image.

randomized perturbation regimes. The resulting performance gap between medium and hard settings provides a diagnostic target for learned controllers and highlights the specific regimes where additional predictive capability is likely required.

Contributions.

- **RocketLanding benchmark + protocol:** a physics-based powered-descent-and-landing task with randomized perturbations emphasizing non-hoverable thrust constraints and limited gimbal authority.
- **Analytical expert baseline:** an interpretable suicide-burn guidance and PD attitude/lateral controller that establishes a strong reference point and exposes dominant failure modes under hard perturbations.
- **World-modeling blueprint (future work):** a JEPA-style action-conditioned latent dynamics + goal-conditioned MPC design intended for offline trajectory learning in this domain.

Artifacts. We release a repository containing the RocketLanding environment wrapper, the analytical expert, and scripts to reproduce the perturbation protocol and baseline evaluation tables, as well as a trajectory-generation script for offline world-model training.

2 Related Work

World models and latent planning. Model-based RL has long pursued learning compact dynamics for planning (Ha & Schmidhuber, 2018; Hafner et al., 2019b). Latent world-model agents use learned state-space models with imagination rollouts to solve diverse pixel-control tasks (Hafner et al., 2019a; 2020; 2023), while MuZero-style approaches learn abstract predictive models optimized for planning-relevant targets (Schrittwieser et al., 2019). Visual MPC methods plan toward image/latent goals by rolling out learned predictors under sampling-based MPC (Finn & Levine, 2017; Ebert et al., 2017), and TD-MPC improves long-horizon latent control by combining planning with value learning (Hansen et al., 2022).

Predictive self-supervision and JEPAs. Self-supervised teacher–student encoders (DINO/DINOv2) provide strong, transferable visual representations without labels (Caron

et al., 2021; Oquab et al., 2023). JEPA-style objectives predict targets in representation space (I-JEPA, V-JEPA), motivating our planned action-conditioned latent prediction setup (Fig. 1) (Assran et al., 2023; Bardes et al., 2024).

Rocket landing and powered descent guidance. Convex powered descent guidance with lossless convexification (Açıkmeşe & Ploen, 2007; Blackmore et al., 2010), its real-time variant G-FOLD (Açıkmeşe et al., 2013), and enhancements (Açıkmeşe et al., 2008) are widely used for fuel-optimal landing. In our simulated setting with non-hoverable thrust constraints, we find an analytical suicide-burn controller with PD attitude/lateral guidance to be a robust baseline that exposes clear failure modes under hard perturbations.

3 ENVIRONMENT: ROCKETLANDING (PYFLYT)

We evaluate in a physics-based powered-descent-and-landing task built on `PyFlyt`/PyBullet with a Gymnasium API (Tai et al., 2023; Coumans, 2015; Towers et al., 2024). Episodes run at **40 Hz** for up to 30s and require a soft, near-upright touchdown on a landing pad under randomized initial conditions. The environment supports both state observations and rendered pixels/goal images. Our analytical expert is state-based, while the proposed JEPA pipeline targets the pixel-based variant (Fig. 1). All reported baseline results are obtained with state observations, while the pixel-based setting is reserved for future world-model experiments. **Appendix A** provides full details.

4 BASELINE: ANALYTICAL SUICIDE BURN GUIDANCE

We implement an interpretable powered-descent baseline based on **analytical suicide-burn guidance** with PD attitude control and gain-scheduled lateral steering. This baseline is motivated by the non-hoverable regime ($T/W > 1$ at minimum throttle), which requires a precisely timed braking maneuver and tight coupling between attitude stabilization and lateral correction near touchdown. While convex MPC is a common choice for powered descent (Açıkmeşe & Ploen, 2007; Açıkmeşe et al., 2013), we found this analytical guidance to be more robust to our simulator's physics characteristics.

Approach. The controller runs at 40 Hz and follows a three-phase structure: COAST (engine off until a commit condition), BURN (ignite and regulate throttle to decelerate toward a target descent rate), and FINAL (altitude-scheduled vertical velocity targets for soft touchdown). Attitude and lateral guidance are handled by a PD gimbal controller tracking target roll/pitch angles computed from pad-relative lateral position/velocity errors, with gain scheduling and tilt limits for stability. Full controller details, equations, thresholds, and gain schedules are provided in Appendix B. We use the expert to generate offline trajectories for future pixel-based world-model training and goal-conditioned control.

4.1 EVALUATION

We evaluate the controller under two difficulty levels with randomized initial conditions, as seen in Table 1. The hard regime is intentionally configured so that many failures occur near touchdown, where tight actuation limits and contact dynamics amplify small state-estimation or planning errors.

Table 1: Initial condition perturbations

Parameter	**Medium**	**Hard**
Altitude	80–120 m	80–120 m
Lateral offset	0–12 m	0–35 m
Initial tilt	0–8°	0–25°
Lateral velocity	0–1.5 m/s	0–4 m/s
Vertical velocity	-5 to -12 m/s	-5 to -12 m/s
Angular velocity	± 0.03 rad/s	± 0.15 rad/s

Success criteria. A landing is successful if all of the following hold at touchdown:

$$\|\omega\| < 5 \text{ rad/s}, \qquad \|v\| < 2 \text{ m/s}, \tag{1}$$

$$\|\phi, \theta\| < 0.5 \text{ rad } (\approx 29^\circ), \qquad \|p_{xy}\| < 5 \text{ m}. \tag{2}$$

A *fatal collision* occurs if $\|\omega\| > 10$ rad/s or $\|v\| > 5$ m/s at ground contact.

We report success rate (primary) and touchdown quality metrics including final pad distance, impact speed, max tilt, and max angular rate.

Table 2: Baseline controller performance (20 episodes each)

Difficulty	Success Rate	Fatal Rate	Avg Final Dist
Medium	90% (18/20)	10%	0.52 m
Hard	20% (4/20)	80%	8.4 m

Results. The controller achieves high success on medium perturbations but struggles with hard conditions. Analysis of failures reveals that the combination of large lateral offset (> 25 m), high initial tilt ($> 20^\circ$), and lateral velocity (> 3 m/s) exceeds the 15° gimbal authority's correction capability. Successful hard landings (4/20) occurred when either tilt or lateral velocity was moderate despite large offsets.

5 Conclusion and Future Work

In this paper, we introduced **RocketLanding**, a physics-based powered-descent-and-landing benchmark. We also provided a strong, interpretable analytical suicide-burn expert baseline that performs well under medium perturbations but degrades sharply under hard perturbations. This gap highlights a concrete set of failure modes (limited gimbal authority under large lateral offsets, high initial tilt, and lateral velocity) that motivate predictive models capable of reasoning over longer horizons and coupling attitude stabilization with lateral correction near touchdown. Importantly, the hard regime isolates the contact-critical, actuation-limited corner cases that are often hidden in easier pixel-control benchmarks and that a useful world model must handle.

Planned world-model pipeline. Our next step is to train a JEPA-style world model on offline trajectories generated by the expert and by additional diverse rollouts. As illustrated in Fig. 1, we plan to (i) encode observations into a compact latent state z_t using a frozen self-supervised visual encoder, (ii) train an action-conditioned predictor to model latent dynamics, $\hat{z}_{t+1} = f(z_t, a_t)$, and (iii) deploy the predictor inside a goal-conditioned MPC loop that selects actions by unrolling f over horizon h and minimizing distance to a goal latent z_{goal} obtained from a goal image, avoiding reward engineering for the control objective.

Evaluation plan. We will evaluate learned dynamics along two complementary axes. First, we will measure predictive fidelity via multi-step latent rollout error as a function of horizon and perturbation difficulty (medium vs. hard). Second, we will evaluate closed-loop control by replacing the analytical guidance module with latent MPC and reporting landing success, fatal collision rates, and touchdown quality metrics under the same perturbation protocol used for the expert baseline. We will also ablate key design choices suggested by Fig. 1: representation (Euler vs. quaternion observations vs. pixels), action-conditioning, rollout horizon, and goal specification.

Hypotheses and expected failure modes. We expect the hard regime to stress precisely the challenges that world models must solve in this domain: compounding error under long-horizon rollouts, partial observability when using pixels alone (e.g., estimating pad-relative state), and sensitivity to distribution shift induced by MPC-induced action sequences. We will characterize these effects by comparing open-loop prediction metrics to closed-loop performance and by analyzing failure cases where planning errors amplify near touchdown.

Limitations. Our results are simulation-only and reflect the specific contact and actuation characteristics of the underlying physics engine. The expert is hand-designed and may bias the offline data distribution and thus, learned world models trained on these trajectories may require additional data diversity or on-policy refinement for robust control.

REFERENCES

Behçet Açıkmeşe and Scott R. Ploen. Convex programming approach to powered descent guidance for mars landing. *Journal of Guidance, Control, and Dynamics*, 30(5):1353–1366, 2007. doi: 10.2514/1.27553.

Behçet Açıkmeşe et al. Enhancements on the convex programming based powered descent guidance algorithm for mars landing. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2008. URL https://www.larsblackmore.com/AcikmeseAAS08.pdf.

Behçet Açıkmeşe et al. G-fold: A real-time implementable fuel optimal large divert guidance algorithm for planetary pinpoint landing. In *AIAA Guidance, Navigation, and Control Conference*, 2013. URL https://govindchari.com/assets/pdf/gfold_paper.pdf. Real-time, onboard convex PDG via lossless convexification.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Lars Blackmore, Behçet Açıkmeşe, and Daniel P. Scharf. Minimum-landing-error powered-descent guidance for mars landing using convex optimization. *Journal of Guidance, Control, and Dynamics*, 33(4):1161–1171, 2010. doi: 10.2514/1.47202.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *International Conference on Computer Vision (ICCV)*, 2021.

Erwin Coumans. Bullet physics simulation. In *ACM SIGGRAPH 2015 Courses*, 2015. doi: 10.1145/2776880.2792704.

Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *arXiv preprint arXiv:1710.05268*, 2017.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. doi: 10.1109/ICRA.2017.7989324.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, 2019b.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, and Timothy Lillicrap. Mastering atari, go, chess and shogi by planning with a learned model. *arXiv preprint arXiv:1911.08265*, 2019.

Jun Jet Tai, Jim Wong, Mauro Innocente, Nadjim Horri, James Brusey, and Swee King Phang. Pyflyt – uav simulation environments for reinforcement learning research. *arXiv preprint arXiv:2304.01305*, 2023.

Mark Towers, Ariel Kwiatkowski, Jordan K. Terry, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

A Environment: RocketLanding (PyFlyt)

We study a powered descent and landing task in a physics-based simulator built on the `PyFlyt` rocket landing environment, which uses the Bullet physics engine via PyBullet and exposes a Gymnasium-compatible API (Tai et al., 2023; Coumans, 2015; Towers et al., 2024). The goal is to land an underactuated rocket upright on a landing pad within a bounded flight volume under randomized initial conditions. We use this environment as a physically grounded testbed for representation learning and world modeling from expert demonstrations.

Task Definition Each episode initializes the rocket in free fall above a landing pad. The agent must execute the full descent-to-touchdown maneuver, achieving low velocity and near-upright attitude at pad contact, while avoiding excessive tilt and angular rates. We also expose a pad-contact indicator used for termination and touchdown reward shaping.

Control Frequency and Episode Horizon The simulator runs an agent control loop at **40 Hz**. Episodes are capped by a maximum simulated duration (30 seconds by default), and may terminate earlier due to success or failure conditions.

Action Space The action is a 7-dimensional continuous control vector:

$$a_t = [u_{\text{fin},x}, u_{\text{fin},y}, u_{\text{fin,roll}}, u_{\text{ignite}}, u_{\text{throttle}}, u_{\text{gimbal},x}, u_{\text{gimbal},y}],$$

corresponding to (i) aerodynamic finlet commands, (ii) engine ignition, (iii) throttle, and (iv) two-axis gimbal commands. In our expert baseline, ignition is treated as binary, throttle is normalized, and gimbal commands are saturated by a fixed gimbal-angle limit.

Observation Space The observation is a continuous vector containing (i) angular velocity, (ii) orientation, (iii) linear velocity, (iv) linear position, (v) previous action, (vi) auxiliary state, and (vii) a landing-pad contact flag. Orientation can be represented either in Euler angles or quaternions (configurable). In the moving-pad variant (below), the pad position/velocity are provided via the `info` dictionary rather than appended to the core observation vector.

Initialization and Domain Randomization We define a structured variation space spanning (a) initial conditions and (b) landing-pad motion parameters. Initial conditions randomize the start height (as a fraction of a ceiling height), horizontal offset, and initial attitude/tilt within bounded ranges. The environment also supports optional reset-time perturbations that inject a downward entry velocity and add additional lateral displacement (clipped to prevent extreme starts). This produces a family of start distributions with controllable difficulty.

Reward, Termination, and Success The environment supports sparse and dense reward settings. In dense mode, reward components encourage (i) reducing lateral distance to the pad, (ii) making lateral and vertical progress toward touchdown, (iii) decelerating during descent, and (iv) avoiding

excessive angular velocity or tilt. On pad contact, an additional touchdown term is applied with a penalty proportional to vertical collision speed.

Episodes terminate on fatal contact events (e.g., excessive linear/angular speed at ground contact). Episodes are truncated when the rocket is sufficiently slow, upright, near the pad, and below a low-altitude threshold, at which point a completion bonus is applied.

Rendering and Goal Image When rendering is enabled, the environment constructs a per-episode goal image by temporarily placing the rocket in an ideal landed pose above the pad and calling `render()`, storing the resulting RGB goal in `info["goal"]`. This supports vision-based objectives and goal-conditioned evaluation.

B Analytical Suicide Burn Expert Policy Details

We implement a powered-descent guidance baseline using an **analytical suicide burn** approach with PD attitude control. This method is necessitated by the vehicle's thrust-to-weight ratio ($T/W > 1$ at minimum throttle), which prevents hovering and requires a precisely-timed braking maneuver. Unlike convex MPC approaches (Açıkmeşe & Ploen, 2007; Açıkmeşe et al., 2013), we found that analytical guidance is more robust to the simulator's physics characteristics.

Vehicle Parameters The rocket has the following constraints:

$$T_{\max} = 7607 \text{ N}, \qquad T_{\min} = 0.39 \cdot T_{\max} = 2967 \text{ N}, \tag{3}$$

$$m_{\text{dry}} = 159 \text{ kg}, \qquad m_{\text{fuel}} = 410.9 \text{ kg}, \tag{4}$$

$$\theta_{\text{gimbal}} = \pm 15^\circ, \qquad g = 9.81 \text{ m/s}^2. \tag{5}$$

The minimum thrust constraint means $T_{\min}/m_{\text{dry}} > g$, so the rocket *cannot hover*—it must use a suicide burn trajectory.

B.1 Three-Phase Control Architecture

The controller operates at 40 Hz through three sequential phases:

Phase 1: COAST. Engine off, free fall toward the landing pad. The vehicle uses aerodynamic fin-lets and attitude control to maintain orientation and steer toward the target. The controller computes the commit altitude:

$$h_{\text{commit}} = \frac{v_z^2}{2a_{\text{net}}} \cdot k_{\text{safety}} + 5 \text{ m}, \tag{6}$$

where $a_{\text{net}} = T_{\max}/m - g$ is the net deceleration and $k_{\text{safety}} = 2.0$ provides margin.

Phase 2: BURN. When $h \leq h_{\text{commit}}$ or ($h < 10$ m and $v_z < -5$ m/s), the engine ignites for the braking maneuver. The controller computes a braking altitude h_{brake} based on the distance needed to decelerate from current velocity to the target $v_z^{\text{target}} = -3.0$ m/s:

$$h_{\text{brake}} = \frac{v_z^2 - (v_z^{\text{target}})^2}{2a_{\text{net}}} \cdot 1.2. \tag{7}$$

If descending slower than target ($v_z > v_z^{\text{target}}$), throttle is zero (coast). Otherwise, altitude-proximity takes priority: when close to the braking altitude ($h < h_{\text{brake}}+3$), throttle is computed proportionally as $\tau = \min(1.0, 0.4+0.1\cdot(|v_z|-|v_z^{\text{target}}|))$. When not close, throttle depends on velocity and altitude:

$$\tau = \begin{cases} 1.0, & v_z < -18 \text{ m/s} \\ 0.7, & -18 \leq v_z < -10 \text{ m/s and } h < h_{\text{brake}} + 20 \\ 0.5, & -10 \leq v_z < v_z^{\text{target}} \text{ and } h < h_{\text{brake}} + 15 \\ 0.0, & \text{otherwise (coast to conserve fuel)} \end{cases} \tag{8}$$

Additionally, if tilt exceeds 20°, thrust is cut to allow attitude stabilization; if tilt is between 15° and 20°, a minimum throttle of 0.3 is applied to maintain gimbal authority for attitude correction.

Phase 3: FINAL. When ($h < 20$ m and $|v_z| < 5$ m/s) or ($h < 30$ m and $|v_z| < 2$ m/s), the controller transitions to fine control with altitude-scheduled velocity targets for soft touchdown:

$$v_z^\star(h) = \begin{cases} -2.5 \text{ m/s}, & h > 8 \text{ m} \\ -1.2 \text{ m/s}, & 4 < h \leq 8 \text{ m} \\ -0.6 \text{ m/s}, & 1.5 < h \leq 4 \text{ m} \\ -0.4 \text{ m/s}, & h \leq 1.5 \text{ m} \end{cases} \tag{9}$$

B.2 Attitude Control with Lateral Guidance

At 40 Hz, a PD controller computes gimbal commands to maintain attitude and steer toward the landing pad. Target attitudes are computed from lateral position/velocity errors:

$$\phi_{\text{target}} = +k_p^{\text{lat}} \cdot p_y + k_d^{\text{lat}} \cdot v_y, \tag{10}$$

$$\theta_{\text{target}} = -k_p^{\text{lat}} \cdot p_x - k_d^{\text{lat}} \cdot v_x, \tag{11}$$

where ϕ is roll, θ is pitch, and $p_{x,y}$, $v_{x,y}$ are position/velocity relative to the pad. Gimbal commands track these targets with PD control:

$$\gamma_x = k_p^{\text{att}}(\phi - \phi_{\text{target}}) + k_d^{\text{att}}\omega_x, \tag{12}$$

$$\gamma_y = k_p^{\text{att}}(\theta - \theta_{\text{target}}) + k_d^{\text{att}}\omega_y, \tag{13}$$

where gimbals $\gamma_{x,y}$ are saturated at $\pm 15^\circ$ and rate-limited to 4 rad/s. Gains are scheduled based on time-to-go, altitude, tilt magnitude, and angular velocity:

$$k_p^{\text{lat}} = 0.06 \cdot \min(t_{\text{go}}/3, 3) \cdot f_{\text{alt}}(h) \cdot f_{\text{tilt}}(\|\phi, \theta\|) \cdot f_\omega(\|\omega\|), \tag{14}$$

$$k_d^{\text{lat}} = 0.26 \cdot f_{\text{alt}}(h) \cdot \max(f_{\text{tilt}}, 0.2) \cdot f_\omega(\|\omega\|), \tag{15}$$

where $t_{\text{go}} = \max(h/\max(|v_z|, 1), 1)$ when descending ($v_z < -0.5$ m/s), or $\max(h/5, 2)$ otherwise; $f_{\text{alt}} = \min(h/25, 1)$ scales linearly from 0 to 1 as altitude increases to 25 m; f_{tilt} reduces gains in discrete steps (1.0 below 5°, 0.5 at 5–8°, 0.2 at 8–12°, 0.1 at 12–18°, 0.05 above 18°); and $f_\omega = \max(0.1, 1 - 2(\|\omega\| - 0.5))$ when $\|\omega\| > 0.5$ rad/s, else 1.0. The attitude gains are $k_p^{\text{att}} = 15.0$, $k_d^{\text{att}} = 12.0$.

Aerodynamic finlets. At high speed, finlets provide additional attitude damping:

$$u_{\text{fin}} = -\min\left(\frac{\|v\|}{30}, 1\right) \cdot (k_p^{\text{att}} \cdot \text{euler} + k_d^{\text{att}} \cdot \omega). \tag{16}$$

Guidance tilt limits. Target attitudes from lateral guidance are clamped to altitude-dependent limits to prevent over-correction near the ground:

$$\|\phi_{\text{target}}, \theta_{\text{target}}\| \leq \begin{cases} 3^\circ, & h < 8 \text{ m} \\ 6^\circ, & h < 15 \text{ m} \\ 10^\circ, & h < 30 \text{ m} \\ 15^\circ, & \text{large lateral offset or velocity} \\ 7^\circ\text{–}12^\circ, & \text{otherwise, based on offset} \end{cases} \tag{17}$$

Metrics and failure taxonomy. Our **primary** metric is landing **success rate** under each perturbation regime (Table 1). We additionally report **secondary** metrics that capture touchdown quality and safety margins, including final pad distance ($\|p_{xy}\|$), impact speed ($\|v\|$ at contact), maximum tilt during descent ($\max_t \|\phi_t, \theta_t\|$), and maximum angular rate ($\max_t \|\omega_t\|$). When available, we also track fuel usage and constraint violations (e.g., sustained saturation of gimbal/tilt limits). We categorize failures into: (a) *lateral miss* (large $\|p_{xy}\|$ at contact), (b) *attitude loss* (excessive tilt or angular rate), (c) *hard impact* (excessive $\|v\|$ at contact), and (d) *fuel starvation* (insufficient thrust authority), though the latter is rare in our current settings.