Learning Impact-Rich Rotational Maneuvers via Centroidal Velocity Rewards and Sim-to-Real Techniques: A One-Leg Hopper Flip Case Study

Dongyun Kang[*1], **Gijeong Kim**[*1], **JongHun Choe**[*1], **Hajun Kim**[1], **and Hae-Won Park**[†1]

[1]Korea Advanced Institute of Science and Technology (KAIST)

Figure 1: Snapshots of the first successful deployment of the learned front flip on the real one-leg hopper platform. The policy was trained with a centroidal velocity–based reward and sim-to-real techniques to achieve robust, impact-rich rotation on hardware.

Abstract: Dynamic rotational maneuvers, such as front flips, inherently involve large angular momentum generation and intense impact forces, presenting major challenges for reinforcement learning and sim-to-real transfer. In this work, we propose a general framework for learning and deploying impact-rich, rotation-intensive behaviors through centroidal velocity-based rewards and actuator-aware sim-to-real techniques. We identify that conventional link-level reward formulations fail to induce true whole-body rotation and introduce a centroidal angular velocity reward that accurately captures system-wide rotational dynamics. To bridge the sim-to-real gap under extreme conditions, we model motor operating regions (MOR) and apply transmission load regularization to ensure realistic torque commands and mechanical robustness. Using the one-leg hopper front flip as a representative case study, we demonstrate the first successful hardware realization of a full front flip. Our results highlight that incorporating centroidal dynamics and actuator constraints is critical for reliably executing highly dynamic motions. A supplementary video is available at: https://youtu.be/atMAVI4s1RY

Keywords: Reinforcement Learning, Sim-to-Real Transfer, One-Leg Hopper

1 Introduction

Legged robots have recently demonstrated an impressive range of dynamic behaviors, from agile locomotion to acrobatic maneuvers [1, 2, 3, 4, 5, 6, 7, 8]. These capabilities push beyond low-speed locomotion and begin to exploit the full dynamics of the system. Critically, such dynamic behaviors inherently involve the generation and dissipation of momentum through strong impacts [7, 8], the modulation of body inertia through posture changes [4, 5], and the precise control of rotational motion during flight or contact transitions [6]. Managing these phenomena—momentum exchange and

[*]These authors contributed equally.

[†]Corresponding author. haewonpark@kaist.ac.kr

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

inertial shaping—is a hallmark of dynamic legged systems, distinguishing them from other robotic platforms [9]. However, robotic motions that demand large body rotations and endure strong impacts pose fundamental challenges for reinforcement learning (RL) and sim-to-real transfer. Policies must not only coordinate whole-body dynamics to achieve significant rotational maneuvers [10, 11], but must also respect tight hardware limits on torque production [11, 2], and impact tolerance [12]. Small mismatches between simulation and reality can easily result in catastrophic failures during such high-energy motions.

In this work, we propose a framework for learning and deploying impact-rich, rotation-intensive behaviors through centroidal velocity–based rewards and actuator-aware sim-to-real techniques. We identify that conventional link-level reward formulations often fail to induce true whole-body rotation, as they can be satisfied by exploiting internal joint motions without achieving meaningful global movement. To address this, we introduce a centroidal angular velocity reward that accurately captures system-wide rotational dynamics and drives the discovery of flipping behaviors. To bridge the sim-to-real gap under extreme actuation demands, we incorporate Motor Operating Region (MOR) modeling into the simulation. By accurately reflecting the torque-speed limitations of real actuators, this ensures that policies remain within feasible motor capabilities throughout training. To protect vulnerable hardware components during high-impact contacts, we apply transmission load regularization, penalizing excessive joint stresses and promoting safer landing strategies.

As a concrete demonstration, we select the front flip on a one-leg hopper, a minimal legged system with a single foot, a small support polygon, and no rotational authority from auxiliary limbs. Performing a front flip on such a platform is challenging due to the highly nonlinear coupling between leg actuation and body rotation, the need for precise impulse generation during takeoff, and the demanding contact dynamics at landing. We validate our framework through the first successful hardware realization of a front flip on this minimal system, demonstrating that impact-rich, rotation-intensive behaviors can be learned and deployed reliably under severe actuation and structural constraints. This task epitomizes the challenges of executing large aerial rotations and absorbing high-magnitude impacts, making it an ideal benchmark for evaluating our methodology. Moreover, mastering this maneuver provides broader insights into controlling underactuated robotic systems, offering a transferable recipe for robust sim-to-real learning beyond the specific case of flipping.

In summary, our contributions are threefold:

- We propose a centroidal angular velocity reward that accurately captures whole-body rotation and guides the discovery of dynamic maneuvers.
- We introduce sim-to-real transfer techniques, including realistic Motor Operating Region (MOR) modeling and transmission load regularization, and analyze their critical role in achieving realistic, and hardware-resilient deployment in impact-rich rotational tasks.
- We demonstrate the first successful hardware implementation of a front flip on a one-leg hopper, validating the effectiveness of our approach on a minimal and challenging platform.

2 Related Work

Dynamic Motion in Legged Robots. In the legged robot fields, many studies have explored dynamic motions such as fast running [13, 1, 2], bucking [14], parkour [15, 16, 3]. In particular, some studies have demonstrated agile rotational maneuvers, such as cartwheels [6], barrel rolls [4], and flips [5, 17, 7, 18, 19, 8, 20]. These challenging maneuvers involve high impacts and require consideration of complex dynamics during both takeoff and landing.

Motion Planning with Centroidal Momemtum. The centroidal momentum model has been extensively studied in model-based motion generation approaches [21, 22, 10, 11, 23], as it effectively captures the whole-body dynamics of legged robots [24]. Orin et al. [22] analyzed the Centroidal Momentum Matrix (CMM) and characterized the concept of centroidal velocity. Dai et al. [10] and Zhou et al. [11] proposed motion planning frameworks that incorporate centroidal momentum for acrobatic motion in humanoid robots. In learning-based methods, base angular velocity rewards

have been widely used to encourage rotational motion [7, 25]. More recently, several studies have begun to utilize centroidal momentum to better capture whole-body dynamics [9, 26, 27, 28]. In particular, Xie et al. [28] introduced rewards based on centroidal angular momentum to account for the contribution of leg-induced momentum. However, these approaches have primarily focused on momentum regularization for balance control, rather than on actively generating dynamic rotational behaviors. In contrast, our work leverages the concept of centroidal velocity to explicitly guide policy learning toward rotation-intensive motions.

Sim-to-Real Transfer in Dynamic and Impact-Rich Tasks. Applying learned motions to real legged robots remains highly challenging due to the sim-to-real gap [9]. To address this, various methods have been proposed to mitigate this gap, including domain randomization [29, 30, 31, 32], noise injection [31, 32, 33], and online system identification using real-world data [34, 35]. As the field progresses, and legged robots advance toward more dynamic behaviors, recent studies increasingly focus on handling high impacts and actuator constraints. For instance, Shin et al. [2] adopted MOR constraints to consider the actuator's physical limits during training, while Ostyn et al. [12] proposed an impact-aware planner that accounts for transmission-level load propagation. Building on these efforts, in this work, we investigate sim-to-real transfer techniques that explicitly incorporate actuator constraints, aiming to enable reliable execution of dynamic and impact-intensive motions on real legged robots.

3 Hardware Platform

Our target platform is a compact, custom-designed 3-DOF one-leg hopper system inspired by human lower-limb kinematics. It consists of a single knee pitch joint and a two-axis ankle (pitch and roll), terminating in a flat foot. The platform is capable of executing forward, backward, and lateral command-tracking locomotion, and can recover from external pushes with no additional support. Key specifications and structural features are as follows:

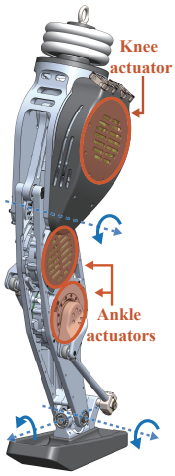


Figure 2: One-leg hopper used in this study.

Physical Dimensions. The robot weighs 12.45 kg, and when its knee is fully extended, it reaches 890 mm in height. The thigh is 0.35 m long and weighs 7.60 kg, the calf is 0.41 m long and weighs 3.89 kg, and the foot is 0.08 m long in height and weighs 0.96 kg. These measurements closely match human lower-limb proportions. The flat foot measures approximately 200 mm in length and 100 mm in width. And a small toe section is integrated into the foot plate to increase contact area and improve ground reaction stability.

Closed-Loop Ankle Mechanism. Ankle pitch and roll motions are produced by two parallel actuators coupled through a closed-loop mechanism using universal and ball joints. This enforces kinematic constraints between the actuators and the foot. The closed-loop structure is simulated in RaiSim [36], which provides pin constraints in the kinematic tree to enforce coupling consistency.

4 Method

4.1 Centroidal Velocity based Reward

Maximizing base angular velocity (BAV) is a common reward design for rotational motions such as flips [7, 25]. However, we found that such link-level rewards can fail to induce true whole-body rotation. In a one-leg hopper, rewarding the base (i.e., the thigh) pitch angular velocity leads the robot to generate rapid internal motion between the thigh and calf without net body rotation (see Sec. 5.1). Because the internal angular momenta of the thigh and calf cancel, the robot achieves high base angular velocity while producing negligible net angular momentum. An alternative is to maximize centroidal angular momentum (CAM). While CAM-based rewards encourage momentum generation, we observed that they fail to convert it into sufficient angular velocity for completing the

flip. This is because increasing rotation speed requires not only generating momentum but also reducing inertia, yet CAM rewards provide no incentive for inertia reduction. As a result, policies trained with CAM rewards converge to inefficient motions that do not leverage inertia modulation to amplify rotation speed. In this work, we instead use a reward based on the centroidal angular velocity (CAV), which inherently accounts for both angular momentum and moment of inertia.

Centroidal dynamics describe the motion of a multibody system in terms of translation and rotation about its center of mass (COM). According to [22], the centroidal momentum $\mathbf{h_G} = [\mathbf{p_{com}}^T,\ \mathbf{L_{com}}^T]^T \in \mathbb{R}^6$ satisfies the relation $\mathbf{h_G} = \mathbf{I_G}\mathbf{v_G}$, where $\mathbf{v_G} = [\mathbf{v_{com}}^T,\ \mathbf{w_{com}}^T]^T \in \mathbb{R}^6$ is the system's 6D twist, and $\mathbf{I_G} \in \mathbb{R}^{6\times 6}$ is the centroidal composite rigid body inertia. $\mathbf{I_G}$ represents the total inertia of the system when all joints are locked, effectively modeling it as a rigid body. When $\mathbf{I_G}$ is defined at the COM, it takes a block-diagonal form: $\mathbf{I_G} = \mathrm{blockdiag}(m\mathbf{I_3}, \mathbf{I_{com}})$, where m is the total mass and $\mathbf{I_{com}} \in \mathbb{R}^{3\times 3}$ is the composite rigid body inertia about the COM. The angular momentum component therefore reduces to $\mathbf{L_{com}} = \mathbf{I_{com}}\mathbf{w_{com}}$, and the centroidal angular velocity (CAV), which can be interpreted as the system's overall rotation rate, is given by $\mathbf{w_{com}} = \mathbf{I_{com}^{-1}}\mathbf{L_{com}} \in \mathbb{R}^3$ [22]. We use both the centroidal angular momentum (CAM) $\mathbf{L_{com}}$ and the centroidal angular velocity (CAV) $\mathbf{w_{com}}$ to formulate our flip reward. The two-second motion is divided into three phases: a takeoff phase (0–0.5 seconds), an aerial phase (0.5–1.05 seconds), and a landing phase (1.05–2.0 seconds). The aerial phase duration can be adjusted for different tasks, allowing for longer airborne motions by extending flight time. During the aerial phase, we encourage angular velocity aligned with the desired rotational axis $\boldsymbol{\alpha}$, and during landing, we penalize residual angular momentum. For front flips, we set $\boldsymbol{\alpha} = [0,\ 1,\ 0]^T$ to target pitch rotation. The reward is defined as follows: $r_{\mathrm{CAV}} = 0$ during takeoff, $r_{\mathrm{CAV}} = \max(\min(\boldsymbol{\alpha}^T\mathbf{w}_{\mathrm{com}}, 10), -0.1)$ during the aerial phase, and $r_{\mathrm{CAV}} = -0.5\ \min(\|\mathbf{L}_{\mathrm{com}}\|,\ 2.5)$ during landing.

4.2 Sim-to-Real Transfer and Impact Mitigation Techniques

In this section we outline two key techniques that enable reliable transfer of the learned flip motion from simulation to the physical one-leg hopper. Additional techniques used to facilitate sim-to-real transfer are described in Appendix A.4.

Motor Operating Region. Dynamic, impact-rich maneuvers demand torque performance at the very edge of actuator capabilities. In particular, front flips on a one leg hopper require rapid takeoff impulses that push the knee and ankle motors into high-speed, high-torque regimes. Therefore, accurately modeling the available Motor Operating Region (MOR) [2, 11] and constraining commands within this envelope during simulation is critical for reliable policy transfer.

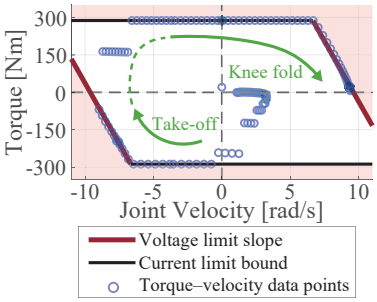


Figure 3: Motor Operating Region (MOR) at the knee actuator. Simulation data were collected within ± 0.25 seconds during the flip motion around take-off, with take-off and knee fold events also noted in the figure. Red regions denote areas beyond the MOR.

Many reinforcement learning frameworks adopt a simple box-type MOR, with constant torque and velocity bounds. Such fixed bounds fail to capture the speed-dependent characteristics of real motors, where available torque drops with increasing rotor speed due to back-EMF. Fig. 3 illustrates a more realistic MOR [2], shown as a trapezoidal region in the torque–speed plane. This envelope is bounded by two critical lines. The first boundary, marked in red, is the *voltage limit slope*. As rotor speed increases, the back-EMF reduces the available voltage, causing a linear falloff in maximum torque. The second boundary, shown in black, is the *soft current limit*. Even if the voltage margin allows greater torque, a current limit is enforced to prevent overheating. During training, we clip commanded torques to remain within this envelope. Without this constraint, the policy exploits torque–speed combinations that are unattainable on the real hardware. The blue points in Fig. 3 represent torque–speed pairs commanded during the flip motion. Notably, the policy naturally drives the actuator near the MOR boundary, fully utilizing the available region to generate sufficient takeoff impulse. If a simplified box-type MOR were used instead, the policy would learn

in an unrealistic region (red region in Fig. 3). On real hardware, this mismatch may cause incorrect contact impulses at takeoff, leading to poor transfer and failed flips.

Transmission Load Regularization. To protect vulnerable transmission components, we introduce a regularization term that penalizes large joint loads induced by contact impacts. The one-leg hopper experiences high-impact forces on a single limb, and early hardware tests revealed sun gear failure due to excessive impulse-induced loads. Notably, minimizing contact impulses alone does not guarantee reduced actuator loads in the closed-loop ankle linkage, since impact forces can redistribute through the mechanism. We estimate the instantaneous transmission load from external contacts using the contact impulse and jacobian. These external torques are then averaged over the control interval to compute the effective joint load: $\boldsymbol{\tau}_{\text{inst}} = \sum_{\text{contacts}} \frac{\mathbf{J}^{\mathsf{T}} \boldsymbol{\lambda}}{\Delta t_{\text{sim}}}, \quad \boldsymbol{\tau}_{\text{load}} = \frac{1}{\Delta t_{\text{control}}/\Delta t_{\text{sim}}} \sum \left| \boldsymbol{\tau}_{\text{inst}} \right|$ where $\mathbf{J}$ is the contact Jacobian, $\boldsymbol{\lambda}$ is the contact impulse, Δt_{sim} is the simulation timestep, and $\Delta t_{\text{control}}$ is the control interval. The operator $|\cdot|$ denotes component-wise absolute value. We employ the relaxed log barrier function [33] to constrain the peak transmission load. This regularization term is added to the barrier reward to guide the policy towards landing strategies that reduce extreme transmission stress. Furthermore, we impose episode termination for extreme transmission overload. Although the barrier penalty discourages high loads, the brief duration of impact can lead policies to tolerate occasional violations in pursuit of long-term return. To reflect irreversible hardware damage, we terminate the episode with 50% probability whenever $|\boldsymbol{\tau}_{\text{load}}|$ exceeds a critical threshold. This probabilistic termination preserves exploration during early training while imposing a realistic cost for severe overloads that would otherwise disable the real hardware.

4.3 Implementation Details

Reward Composition. The total reward is composed of two parts: *motion rewards*, which guide behavior discovery, and *barrier rewards*, which enforce hardware constraints. The motion reward includes the centroidal angular velocity term (Sec. 4.1) and regularization penalties for stabilizing behavior. The barrier reward employs relaxed log barriers [33] applied to joint limits and transmission load (Sec. 4.2). These are combined as scalar values via $r_{\text{motion}} = (1 + r_{\text{CAV}}) \sum_i w_i r_i$ and $r_{\text{barrier}} = \sum_j w_j b_j$ where r_i and w_i denote the ith motion reward term and its weight, and b_j and w_j denote the jth barrier term and its weight. Details of individual terms are listed in Appendix A.1.

Policy Architecture. The policy is trained using Proximal Policy Optimization (PPO) with an asymmetric actor–critic structure. The actor receives proprioceptive observations, short action history, and phase. The critic is given privileged information, including unactuated joint states and body linear velocity. To bridge the observation gap, we adopt a concurrent estimation framework [37] that augments the policy input with predicted privileged states. See Appendix A.3 for further details.

5 Results

5.1 Effect of Centroidal Angular Velocity based Reward

To compare the effects of different reward formulations on rotational motion, we train policies to maximize base angular velocity (BAV), centroidal angular momentum (CAM), and centroidal angular velocity (CAV, proposed). Each reward replaces the aerial-phase term (0.5–1.05 seconds) in r_{CAV} to induce a front flip. Fig. 4 illustrates the resulting motion for each policy.

The BAV-maximizing policy executes a rapid knee flexion–extension motion between a fully extended singular pose and a folded configuration. Base angular velocity is generated through knee extension from the folded posture, reaching 8.7 rad/s (Fig. 4c). Despite this high base rotational speed, the robot fails to achieve take-off (Fig. 4b) or produce net whole-body rotation (Fig. 4a). In contrast to the BAV-maximizing policy, the CAM-maximizing policy initiates take-off (Fig. 4b) and generates nonzero net rotation. However, the rotation is insufficient to complete a full flip (Fig. 4a). The key limitation arises from maintaining a large composite inertia through an extended leg configuration, to achieve high angular momentum despite low rotational speed. This allows the policy to reach a centroidal angular momentum of 4.3N·s, which is comparable to that of the proposed

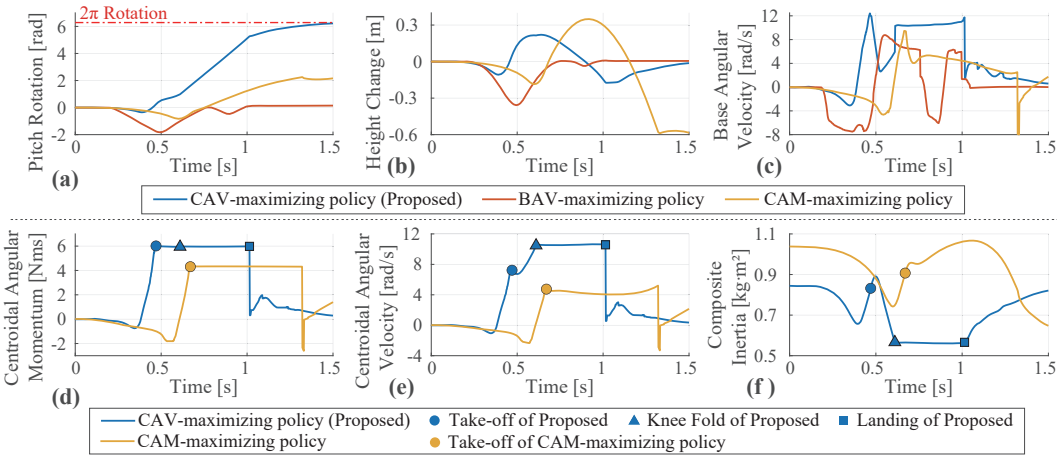


Figure 4: Comparison of policies trained with different aerial-phase rewards (0.5–1.05 s) for front-flip motion. (a)–(b) show base pitch rotation and vertical height (relative to initial pose); (c) shows base angular velocity about the pitch axis. (d)–(f) present centroidal angular momentum (CAM), centroidal angular velocity (CAV), and composite pitch-axis inertia for CAM- and CAV-maximizing policies. Key events (take-off, knee fold, landing) are marked in (d)–(f) for both policies.

policy (6.0N·s; Fig.4d). However, the resulting centroidal angular velocity remains substantially lower (4.8rad/s vs. 10.6rad/s; Fig.4e). Since only the magnitude of angular momentum is rewarded, the policy does not attempt to accelerate its rotation mid-flight by reducing inertia. As a result, the composite inertia remains large throughout the aerial phase (Fig. 4f), and no post-take-off configuration change occurs that would increase rotational speed. The proposed CAV-maximizing policy completes a full front flip (Fig.4a). Prior to take-off, the robot extends the knee while applying force through the heel, generating substantial centroidal angular momentum (Fig.4d). Once airborne, this momentum is conserved in the absence of external torques. The policy then rapidly flexes the knee to reduce the composite rotational inertia (Fig.4f), resulting in a 43% increase in centroidal angular velocity, from 7.4rad/s at take-off to a peak of 10.6rad/s during flight (Fig.4e). This mid-air acceleration enables the robot to complete a full 360-degree rotation before landing (Fig.4a-b). These results indicate that optimizing centroidal angular velocity enables both momentum utilization and inertia modulation, which are critical for executing dynamic rotational maneuvers.

5.2 One-Leg Hopper Front Flip Results and MOR

We deployed the proposed policy on the one-leg hopper hardware, achieving a full front flip (Fig. 5). To evaluate the role of MOR constraints in enabling such dynamic behavior, we compare four cases: The proposed policy, trained with MOR constraints, is evaluated in (1) hardware and (2) simulation with MOR constraints. A second policy, trained without MOR, is evaluated in simulation (3) without and (4) with MOR constraints. In the MOR-unconstrained setup, only static torque and velocity limits from the URDF are enforced, forming a box-shaped constraint in the torque–velocity space.

The policy trained without MOR constraints fails to complete a front flip when evaluated under MOR constraints, despite succeeding in the MOR-unconstrained simulation used for training. As shown in Fig.5a–c, the robot generates insufficient rotational velocity and momentum, resulting in landing before completing a full rotation. This failure arises from torque commands that exceed the feasible MOR, particularly during take-off and mid-air inertia modulation (Fig.5d). Along the MOR, the actuator's available torque decreases at high joint velocities, when torque and velocity are aligned (i.e., during positive mechanical work). These limits are critical during take-off, when the policy attempts to extend the knee and push off with the ankle at high speed, and again after take-off when the knee is rapidly flexed to reduce composite inertia. In both phases, Fig. 5d show a reduction in allowable torque, which the policy, trained under static box-shaped torque limits, fails to respect. This mismatch produces unrealizable torque commands, resulting in reduced actual torque

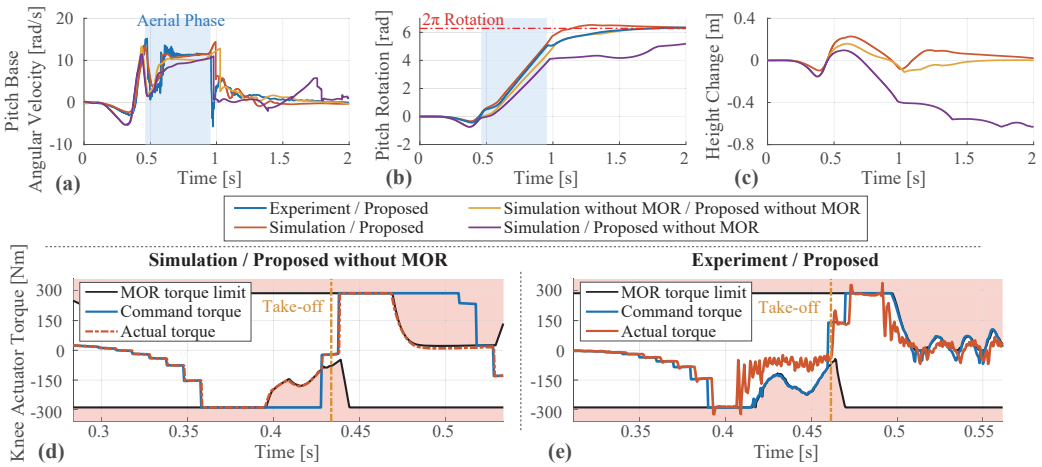


Figure 5: Comparison of simulation and hardware results for policies with and without MOR constraints. (a–c) show base pitch angular velocity, pitch rotation, and vertical height (simulation only), respectively. (d)–(e) show knee torque around take-off from policies trained without MOR (simulation) and with MOR (hardware experiment), respectively. Shaded regions indicate MOR violations.

(Fig. 5d), insufficient rotational velocity (Fig. 5a), and ultimately failure to complete a full rotation (Fig. 5b). In contrast, the proposed method incorporates MOR constraints during training, allowing the policy to adapt to the reduced torque capacity near joint speed extremes observed during take-off and mid-air reconfiguration. As a result, in real-world experiments, the commanded torques remain within the actuator's feasible envelope and yield measured torques that follow the intended profiles (Fig.5e). By generating executable torque commands, the proposed method reproduces the simulated angular velocity profile (Fig.5a) and completes a full front flip on hardware (Fig.5b).

5.3 Effect of Transmission Load Regularization

External Torque at Landing. In hardware trials without transmission load regularization, the robot completed one front flip, but the ankle actuator's sun gear fractured upon landing during the second trial, halting further experimentation. Fig. 6 shows the estimated external torque during landing, computed as the difference between input torque and net torque, with the latter derived from measured joint acceleration. As contact forces are not directly measurable on hardware, this provides an indirect estimate of the load on the actuator. During the initial impact phase (10 ms after contact, 0.02–0.03 s in Fig. 6), a substantial external load was transmitted to the actuator. Although the policy included contact impulse regularization, it failed to mitigate the external load during impact, resulting in gear fracture without transmission load regularization.

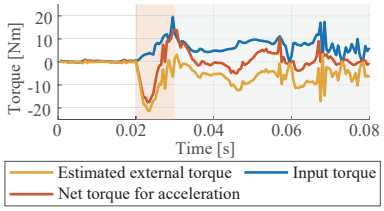


Figure 6: Experimental torque profiles at the ankle left motor during landing. Motor-side torque (without gear reduction) is shown.

Simulation and Hardware Evaluation. We evaluate whether transmission load regularization reduces stress on vulnerable transmission components, specifically the sun gear of the ankle actuator, by comparing policies trained with and without this regularization. Fig. 7 shows the estimated external torque at the ankle motor-side (i.e., on the sun gear), from simulation and hardware experiments. In the simulation results (Fig. 7a–b), solid lines show the mean external torque, and shaded regions indicate the min–max range across runs.

The policy without transmission load regularization exhibits a broader min–max range of external torque across simulation runs (Fig. 7a–b). This arises from flat-foot landings, where slight deviations in foot orientation cause the foot to skid or rotate unpredictably upon impact, resulting in irregular and asymmetric load transmission through the ankle. The expanded external torque range can im-

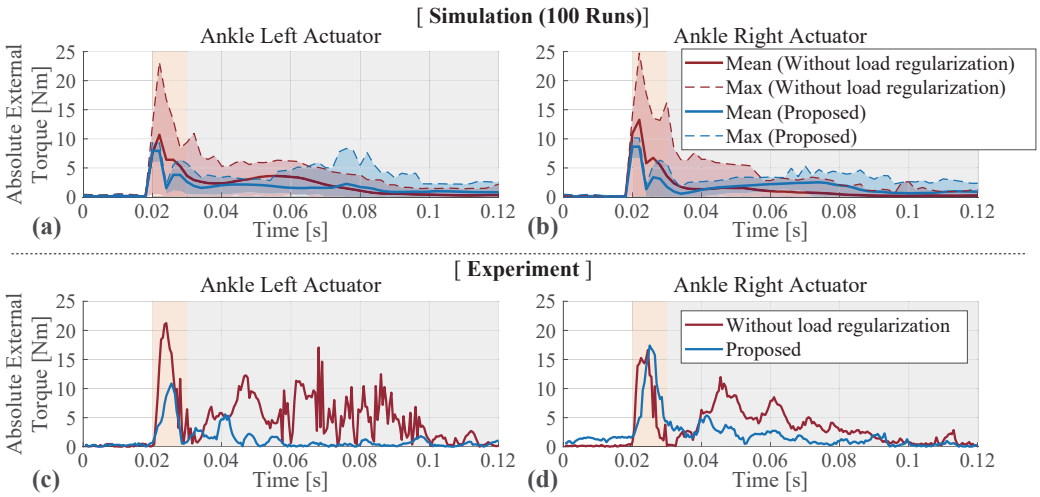


Figure 7: External torques on the ankle actuator's sun gear during landing, for policies trained without vs. with transmission load regularization. (a), (b) are simulated; (c), (d) are estimated from hardware experiments. Time was aligned such that landing starts at 0.02 s in all cases.

pose excessive mechanical load on transmission components. In contrast, the policy trained with regularization yields lower and more consistent external torque trajectories. The learned strategy adjusts ankle orientation to initiate contact at a designated foot corner, followed by a rolling motion through pitch and roll after impact. This extends the contact duration and distributes impact forces over time. Although the mean external torque is comparable, Fig. 7a–b show that the regularized policy reduces variability and yields lower peak external torques. This improvement can reduce the likelihood of excessive peak stress on transmission components during high-impact landings.

The effect of transmission load regularization is further validated through hardware experiments. As shown in Fig.7c–d, the estimated peak external torque decreases from approximately 21 N·m to 17 N·m with regularization, and the peak occurs later in time, indicating more compliant or delayed impact absorption. This reduction in transmission stress translates to improved hardware robustness. Fig. 8 shows pitch rotation over repeated trials: the non-regularized policy caused sun gear fracture on the second attempt, whereas the regularized policy completed eight trials without mechanical failure. These results demonstrate that transmission load regularization not only reduces peak stress but also improves consistency across varied conditions, yielding substantial gains in durability for impact-rich maneuvers.

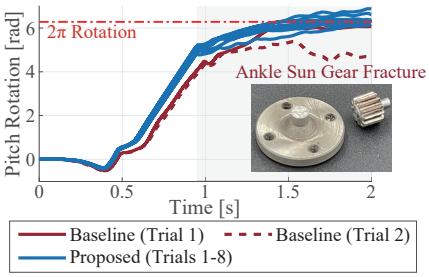


Figure 8: Pitch rotation across repeated hardware flip trials. The baseline (without load regularization) failed on the second trial due to sun gear fracture (inset photo); the regularized policy completed eight trials without failure.

6 Conclusion

This paper presents a framework for learning and deploying impact-rich, rotation-intensive behaviors in legged robots using centroidal velocity–based rewards and actuator-aware sim-to-real techniques. The proposed centroidal angular velocity reward enables whole-body rotations beyond the limitations of conventional link-level rewards. Incorporating Motor Operating Region (MOR) modeling and transmission load regularization enhances the sim-to-real reliability of learned policies by respecting real-world actuator constraints and improving structural robustness. We validated the proposed framework by achieving the first hardware front flip on a one-leg hopper, a minimal yet highly challenging platform. This task, demanding large aerial rotations and high-magnitude impact absorption, served as a representative benchmark for testing the robustness of our approach.

Limitations

The proposed framework has been evaluated primarily on a single hardware platform—a custom-designed one-leg hopper—and focused on a specific dynamic maneuver, namely the front flip. While this setting serves as an effective benchmark for testing impact-rich rotational motion, it does not fully represent the diversity of behaviors or morphologies found in more complex legged systems. Furthermore, our reward formulation and control strategies have not yet been validated on other motion types such as spins, twists, or sequential acrobatic actions. Nonetheless, the underlying principles introduced in this work are not limited to the one-leg hopper scenario. These components are broadly applicable to other legged platforms and dynamic tasks, and we believe they offer promising building blocks for future applications in agile locomotion and acrobatic robotics.

Acknowledgments

This work was supported by the Technology Innovation Program(or Industrial Strategic Technology Development Program-Robot Industry Technology Development)(RS-2024-00427719, Dexterous and Agile Humanoid Robots for Industrial Applications) funded By the Ministry of Trade Industry & Energy(MOTIE, Korea)

References

[1] Y. Jin, X. Liu, Y. Shao, H. Wang, and W. Yang. High-speed quadrupedal locomotion by imitation-relaxation reinforcement learning. *Nature Machine Intelligence*, 4(12):1198–1208, 2022.

[2] Y.-H. Shin, T.-G. Song, G. Ji, and H.-W. Park. Reinforcement learning for high-speed quadrupedal locomotion with motor operating region constraints: Mitigating motor model discrepancies through torque clipping in realistic motor operating region. *IEEE Robotics & Automation Magazine*, 2024.

[3] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024.

[4] B. Dynamics. Atlas — partners in parkour. URL https://www.youtube.com/watch?v=tF4DML7FIWk&ab_channel=BostonDynamics.

[5] U. Robotics. World's first side-flipping humanoid robot: Unitree g1. URL https://www.youtube.com/watch?v=29xLWhqME2Q&ab_channel=UnitreeRobotics.

[6] B. Dynamics. Walk, run, crawl, rl fun — boston dynamics — atlas. URL https://www.youtube.com/watch?v=I44_zbEwz_w&ab_channel=BostonDynamics.

[7] D. Kim, H. Kwon, J. Kim, G. Lee, and S. Oh. Stage-wise reward shaping for acrobatic robots: A constrained multi-objective reinforcement learning approach. *arXiv preprint arXiv:2409.15755*, 2024.

[8] H. Li and P. M. Wensing. Cafe-mpc: A cascaded-fidelity model predictive control framework with tuning-free whole-body control. *IEEE Transactions on Robotics*, 2024.

[9] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv. Learning-based legged locomotion: State of the art and future perspectives. *The International Journal of Robotics Research*, page 02783649241312698, 2024.

[10] H. Dai, A. Valenzuela, and R. Tedrake. Whole-body motion planning with centroidal dynamics and full kinematics. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 295–302. IEEE, 2014.

[11] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim. The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2021.

[12] F. Ostyn, B. Vanderborght, and G. Crevecoeur. Improving the collision tolerance of high-speed industrial robots via impact-aware path planning and series clutched actuation. *IEEE Transactions on Robotics*, 2024.

[13] H.-W. Park, P. M. Wensing, and S. Kim. High-speed bounding with the mit cheetah 2: Control design and experiments. *The International Journal of Robotics Research*, 36(2):167–192, 2017.

[14] G. Kim, D. Kang, J.-H. Kim, S. Hong, and H.-W. Park. Contact-implicit model predictive control: Controlling diverse quadruped motions without pre-planned contact modes or trajectories. *The International Journal of Robotics Research*, 44(3):486–510, 2025.

[15] K. Caluwaerts, A. Iscen, J. C. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang, et al. Barkour: Benchmarking animal-level agility with quadruped robots. *arXiv preprint arXiv:2305.14654*, 2023.

[16] Z. Zhuang, Z. Fu, J. Wang, C. G. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. In *Conference on Robot Learning*, pages 73–92. PMLR, 2023.

[17] B. Dynamics. What's new, atlas? URL https://www.youtube.com/watch?v=fRj34o4hN4I&ab_channel=BostonDynamics.

[18] C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimminger, and G. Martius. Learning agile skills via adversarial imitation of rough partial demonstrations. In *Conference on Robot Learning*, pages 342–352. PMLR, 2023.

[19] Y. Fuchioka, Z. Xie, and M. Van de Panne. Opt-mimic: Imitation of optimized trajectories for dynamic quadruped behaviors. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5092–5098. IEEE, 2023.

[20] M. I. of Technology (MIT). Backflipping mit mini cheetah. URL https://www.youtube.com/watch?v=xNeZWP5Mx9s&ab_channel=MassachusettsInstituteofTechnology%28MIT%29.

[21] S.-H. Lee and A. Goswami. Reaction mass pendulum (rmp): An explicit model for centroidal angular momentum of humanoid robots. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 4667–4672. IEEE, 2007.

[22] D. E. Orin and A. Goswami. Centroidal momentum matrix of a humanoid robot: Structure and properties. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 653–659. IEEE, 2008.

[23] Z. Zhou, B. Wingo, N. Boyd, S. Hutchinson, and Y. Zhao. Momentum-aware trajectory optimization and control for agile quadrupedal locomotion. *IEEE Robotics and Automation Letters*, 7(3):7755–7762, 2022.

[24] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete. Optimization-based control for dynamic legged robots. *IEEE Transactions on Robotics*, 40:43–63, 2023.

[25] F. Vezzi, J. Ding, A. Raffin, J. Kober, and C. Della Santina. Two-stage learning of highly dynamic motions with rigid and articulated soft quadrupeds. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9720–9726. IEEE, 2024.

[26] D. Ferigo, R. Camoriano, P. M. Viceconte, D. Calandriello, S. Traversaro, L. Rosasco, and D. Pucci. On the emergence of whole-body strategies from humanoid robot push-recovery learning. *IEEE Robotics and Automation Letters*, 6(4):8561–8568, 2021.

[27] X. Zhang, X. Wang, L. Zhang, G. Guo, X. Shen, and W. Zhang. Achieving stable high-speed locomotion for humanoid robots with deep reinforcement learning. *arXiv preprint arXiv:2409.16611*, 2024.

[28] W. Xie, C. Bai, J. Shi, J. Yang, Y. Ge, W. Zhang, and X. Li. Humanoid whole-body locomotion on narrow terrain via dynamic balance and reinforcement learning. *arXiv preprint arXiv:2502.17219*, 2025.

[29] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*, 2018.

[30] X. Bin Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. *Robotics: Science and Systems (RSS), Virtual Event/Corvalis, July*, pages 12–16, 2020.

[31] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[32] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62):eabk2822, 2022.

[33] G. Kim, Y.-H. Lee, and H.-W. Park. A learning framework for diverse legged robot locomotion using barrier-based style rewards. *arXiv preprint arXiv:2409.15780*, 2024.

[34] J. Chen, J. Frey, R. Zhou, T. Miki, G. Martius, and M. Hutter. Identifying terrain physical parameters from vision-towards physical-parameter-aware locomotion and navigation. *IEEE Robotics and Automation Letters*, 2024.

[35] H. Kim, D. Kang, M.-G. Kim, G. Kim, and H.-W. Park. Online friction coefficient identification for legged robots on slippery terrain using smoothed contact gradients. *IEEE Robotics and Automation Letters*, 2025.

[36] J. Hwangbo, J. Lee, and M. Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 3(2):895–902, 2018.

[37] G. Ji, J. Mun, H. Kim, and J. Hwangbo. Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robotics and Automation Letters*, 7(2):4630–4637, 2022.

A Appendix

A.1 Reward Formulations

Table 1: Reward terms with expressions and variable definitions.

Term	Expression	Key variables and description
Motion Reward Terms		
r_{CAV}	$3 \cdot \begin{cases} 0, & 0 \le \phi < \phi_{\text{jump}} \\ \max(\min(\boldsymbol{\alpha}^T \mathbf{w}_{\text{com}}, 10), -0.1), & \phi_{\text{jump}} \le \phi < \phi_{\text{land}} \\ -0.5\,\min(\lVert \mathbf{L}_{\text{com}} \rVert, 2.5), & \phi_{\text{land}} \le \phi < 2.0 \end{cases}$	$\boldsymbol{\alpha}$: target flip axis $\mathbf{w}_{\text{com}}$: centroidal angular velocity $\mathbf{L}_{\text{com}}$: angular momentum ϕ: motion phase
r_{lin}	$0.10\exp(-\lVert \mathbf{v}_{xy} \rVert^2)$	$\mathbf{v}_{xy}$: horizontal base velocity
r_τ	$0.10\exp(-5\times10^{-3}\lVert \mathbf{W}_\tau \boldsymbol{\tau} \rVert)$	$\boldsymbol{\tau}$: joint torque
r_p	$0.10\exp(-5\times10^{-2}\lVert \mathbf{W}_p(\theta-\theta_0) \rVert^2)$	θ: joint positions; θ_0: reference pose
r_v	$0.20\exp(-5\times10^{-3}\lVert \mathbf{W}_v \dot{\theta} \rVert^2)$	$\dot{\theta}$: joint velocities
r_a	$0.20\exp(-1\times10^{-3}\lVert \mathbf{W}_a(\dot{\theta}-\dot{\theta}_{\text{prev}}) \rVert^2)$	$\dot{\theta}_{\text{prev}}$: previous joint velocity
r_{slip}	$0.10\exp(-0.2\lVert \mathbf{v}_{\text{foot}} \rVert^2)$	$\mathbf{v}_{\text{foot}}$: foot velocity (xy plane)
r_{act}	$0.20\exp(-0.18\lVert \Delta a \rVert^2)$	$\Delta a = a_t - 2a_{t-1} + a_{t-2}$: second-order action difference
r_{cs}	$0.10\exp(-\lVert c_t - c_{t-1} \rVert^2)$	c_t: contact state vector
r_{ci}	$0.20\exp(-100\lVert \lambda \rVert^2)$	λ: contact impulse vector
Barrier Reward Terms (see Appendix A.2)		
b_{pos}	$\sum_j b(\theta_j; \underline{\theta}_j, \overline{\theta}_j, 0.08)$	θ_j: joint positions with limits
b_{vel}	$\sum_j b(\dot{\theta}_j; \underline{\dot{\theta}}_j, \overline{\dot{\theta}}_j, 2.0)$	$\dot{\theta}_j$: joint velocities with limits
b_{load}	$b(\tau_{\text{load}}; -30, 30, 1)$	τ_{load}: joint load from contact (see Sec. 4.2)

Note. $\mathbf{W}_\tau$, $\mathbf{W}_p$, $\mathbf{W}_v$, $\mathbf{W}_a$ are diagonal weight matrices for the 3 actuated joints: knee, ankle pitch, and ankle roll. Specifically, $\mathbf{W}_\tau = \text{diag}(0.5,\ 2.0,\ 2.0)$, $\mathbf{W}_p = \text{diag}(0.5,\ 1.0,\ 1.0)$, $\mathbf{W}_v = \text{diag}(0.5,\ 2.0,\ 2.0)$, $\mathbf{W}_a = \text{diag}(0.5,\ 2.0,\ 2.0)$. Joint limits used in the barrier rewards are $\underline{\theta} = [-0.4,\ -0.873,\ -0.4]$, $\overline{\theta} = [2.15,\ 0.698,\ 0.4]$, $\underline{\dot{\theta}} = [-8,\ -12,\ -12]$, $\overline{\dot{\theta}} = [8,\ 12,\ 12]$.

Our total reward consists of two parts: *motion rewards*, which guide the discovery and regularization of the motion, and *barrier rewards*, which enforce safety and hardware limits. The motion reward combines the centroidal angular velocity term from Sec. 4.1 with a collection of regularization penalties that suppress excessive or unstable behaviors. The barrier reward uses a relaxed log barrier [33] to sharply penalize violations of joint, velocity, and transmission limits.

The overall scalar rewards are defined as:

$$r_{\text{motion}} = \big(1 + r_{\text{CAV}}\big) \sum_i w_i\, r_i, \qquad r_{\text{barrier}} = \sum_j w_j\, b_j,$$

where r_{CAV} is the centroidal angular velocity term, r_i are the auxiliary motion-regularization rewards with weights w_i, and b_j are barrier terms defined via relaxed log barrier function $b(\cdot)$ (Appendix A.2) with weights w_j.

The key terms are summarized below:

- r_{CAV}: centroidal angular velocity(CAV) reward (Sec. 4.1).
- r_{lin}: horizontal linear velocity reward.
- r_τ: torque penalty to encourage energy-efficient motions.
- r_p, r_v, r_a: joint position, velocity, and acceleration regularizations for smooth trajectories.
- r_{slip}: foot slip penalty to minimize contact sliding.
- r_{act}: action smoothness penalty on consecutive commands.

- r_{cs}: contact smoothness penalty to avoid abrupt contact mode changes.
- r_{ci}: contact impulse penalty to discourage reliance on large simulated impulses.
- b_{pos}: joint position barrier to enforce position limits.
- b_{vel}: joint velocity barrier to enforce velocity limits.
- b_{load}: transmission load barrier (Sec. 4.2).

Detailed expressions for each reward term are provided in Table 1.

A.2 Relaxed Log Barrier Function

The relaxed log barrier function is used to softly enforce inequality constraints while retaining differentiability near the boundary. This specific formulation is adapted from [33], which introduces a smooth approximation to the standard logarithmic barrier.

The relaxed log barrier term is defined for a scalar x with lower and upper limits $\alpha_{\min}, \alpha_{\max}$ and relaxation parameter δ as

$$b(x; \alpha_{\min}, \alpha_{\max}, \delta) = f(x - \alpha_{\min}) + f(\alpha_{\max} - x),$$

where

$$f(z) = \begin{cases} \log \delta - \frac{1}{2}\left(\left(\frac{z-2\delta}{\delta}\right)^2 - 1\right), & z < \delta, \\ \log z, & z \geq \delta. \end{cases}$$

A.3 Network Architecture and Training Details.

The policy network consists of three fully connected hidden layers of size 256, 128, and 64, each followed by a LeakyReLU activation. It is trained using Proximal Policy Optimization (PPO) with an asymmetric actor–critic formulation. The actor's observation includes the gravity vector in the body frame, the body's angular velocity from the IMU, the current angles and angular velocities of the left and right ankle motors, and an action-history buffer of the two most recent commands. To capture temporal context, we also include the joint position error and joint velocity history over the past 0.18 s as well as the sine and cosine of the phase variable. The actor outputs target joint positions, which are passed to a low-level proportional-derivative (PD) controller to generate torque commands according to $\tau = K_p(q^* - q) - K_d\dot{q}$, where q^* is the target joint position given by the policy, q is the current joint position, and $\dot{q}$ is the joint velocity. This torque is then applied to the simulated or real robot. The critic receives both the actor's observation and privileged data, including the unactuated ankle's pitch and roll joint positions and velocities, and the body's linear velocity. Its value function is realized as a dual-headed MLP, where each head consists of three hidden layers with sizes 256, 128, and 64. One head predicts the return associated with the primary motion reward, while the other predicts the return for the barrier penalty. We compute two advantage functions, one from each head, and form the clipped surrogate loss using their sum with equal weighting. This formulation yields more accurate return estimates and provides stronger guidance to the actor. To bridge the information gap, we leverage a concurrent estimation framework. An auxiliary estimator network, implemented as a two-layer MLP with hidden sizes 256 and 128, ingests the actor's observations and predicts the privileged states used by the critic. These estimated states are concatenated with the original observations and fed into the policy network, relieving the actor from modeling the full closed-loop dynamics from scratch.

A.4 Auxiliary Methods for Sim-to-Real Transfer

In addition to our core sim-to-real techniques, we also adopt auxiliary strategies during training to further enhance transferability to the real hardware.

Domain Randomization and Noise Injection. We apply domain randomization and noise injection to improve robustness against modeling errors and hardware variability. At the start of each episode,

we randomize key physical properties (mass, inertia, center of mass), environmental parameters (friction, restitution), actuator properties, and sensor noise. These perturbations expose the policy to diverse conditions and improve transfer stability. Detailed randomization and noise parameters are listed in Table 2 and Table 3.

Table 2: Domain randomization settings used during training.

Parameter	**Distribution / Range**
Total mass	Uniform scaling in [0.85, 1.15]
Link inertias	Uniform scaling in [0.85, 1.15]
COM position	Uniform shift in [-3, 3]cm
Joint PD gains	Uniform scaling in [0.9, 1.1]
Joint friction	Uniform in [0, 0.7] Nm (knee), [0, 1.0] Nm (ankle)
Ground friction coefficient	Uniform in [0.5, 0.9]
Restitution coefficient	Uniform in [0, 0.2]

Table 3: Observation noise settings (uniform noise applied with specified scales).

Observation Type	**Noise Scale**
Gravity vector (body frame)	0.07
Body angular velocity	0.10 rad/s
Joint position	0.05 rad
Joint velocity	0.50 rad/s
Joint position history	0.05 rad
Joint velocity history	0.10 rad/s

Contact Mode and Contact Impulse Regularization. To mitigate reliance on inaccurate impact modeling, we penalize excessive contact impulses during takeoff and landing. Simulated contact forces are governed by simplified restitution and friction models that often diverge from real-world behavior. As a result, policies that exploit large simulated impulses can fail on hardware, particularly when real-world discrepancies cause contact events to be mistimed or weakened. In particular, a policy may learn to rely on a single high-impulse contact occurring over a very short time window. In reality, even a slight timing mismatch can cause that critical contact to be missed, leading to failure of the entire maneuver. To prevent this, we penalize both the magnitude of each contact impulse and abrupt changes across timesteps, reducing high-frequency spikes that are unlikely to be realizable on physical hardware.

A.5 Validation of External Torque Estimation.

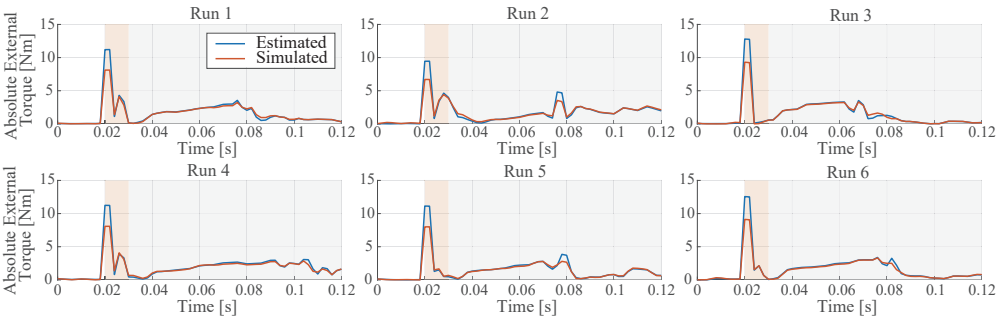


Figure 9: Comparison of estimated and simulated external torque on the ankle actuator across six representative trials during impact.

We estimate the external torque as the difference between the input torque and the torque required to accelerate the effective inertia, computed as: $\boldsymbol{\tau}_{\text{ext}} = \boldsymbol{\tau}_{\text{input}} - \mathbf{I}_{\text{eff}}\, \dot{\boldsymbol{\omega}}_{\text{rotor}}$. Since the ankle actuator

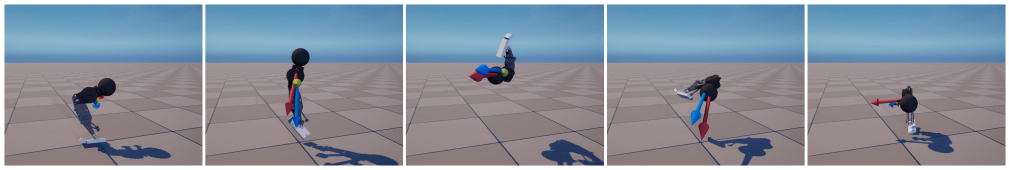
is connected to the foot through a closed-loop mechanism, we approximate the effective inertia as the sum of the rotor inertia and half of the foot link's inertia : $\mathbf{I}_{\text{eff}} = \mathbf{I}_{\text{rotor}} + \mathbf{I}_{\text{foot}}/2$.

To evaluate the accuracy of this estimation method, we compared the estimated torques against ground-truth values computed from contact forces in simulation (see Sec. 4.2). Figure 9 illustrates six representative trials, overlaying the estimated and simulated torques. The shaded orange region indicates the initial impact phase (0.02–0.03 s), and the gray region marks the post-impact phase (0.03–0.12 s). Across these trials, the estimated torques closely match the simulated ground truth in both timing and magnitude.

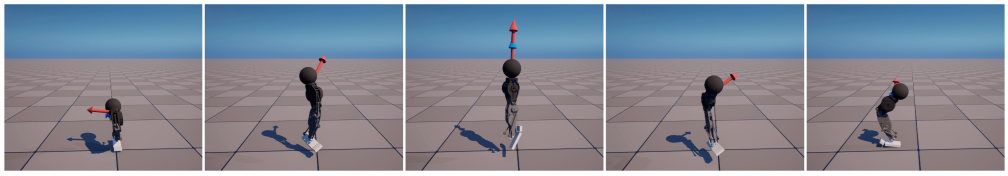
To further quantify accuracy, we compute root-mean-square error (RMSE) within each phase, averaged across 100 trials and both ankle actuators. The resulting RMSE is 1.783 Nm during the initial impact phase and 0.297 Nm during the post-impact phase, indicating reasonable alignment between estimated and simulated torques throughout the contact event.

A.6 Applicability to Other Platforms and Tasks.

To evaluate the generality of our framework, we extended its application beyond the front flip on a one-leg hopper to other tasks and a more complex robotic platform.



(a) Barrel roll



(b) Yaw-spin

Figure 10: Additional maneuvers learned by the one-leg hopper using the same reward framework.

First, we tasked the one-leg hopper, now operating without its closed-loop ankle mechanism, with learning two additional aerial maneuvers: a yaw-spin and a barrel roll. As shown in Fig. 10, both motions were successfully learned by merely adjusting the desired rotational axis vector in the reward function. This demonstrates the flexibility of our centroidal angular velocity reward, as no other changes to the reward formulation or learning pipeline were necessary.

Figure 11: A backflip trained on the Unitree Go1 quadruped.

Furthermore, we applied our framework to a morphologically distinct platform, the Unitree Go1 quadruped, to train a backflip policy in simulation (Fig. 11). The policy was successfully acquired using the identical centroidal angular velocity reward structure and actuator-aware constraints (MOR modeling) from the hopper experiments. This result indicates that our approach generalizes effectively across platforms with different morphologies and actuation structures.

These additional results confirm that our method is capable of scaling to a wide range of dynamic, impact-rich maneuvers across various legged systems.