REBot: Reflexive Evasion Robot for Instantaneous Dynamic Obstacle Avoidance

Zihao Xu[1*], Ce Hao[1], Chunzheng Wang[2], Kuankuan Sima[2], and Qiaojun Yu[3]

***Abstract*— Dynamic obstacle avoidance (DOA) is critical for quadrupedal robots operating in environments with moving obstacles or humans. Existing approaches typically rely on navigation-based trajectory replanning, which assumes sufficient reaction time and leading to fails when obstacles approach rapidly. In such scenarios, quadrupedal robots require reflexive evasion capabilities to perform instantaneous, low-latency maneuvers. This paper introduces Reflexive Evasion Robot (REBot), a control framework that enables quadrupedal robots to achieve real-time reflexive obstacle avoidance. REBot integrates an avoidance policy and a recovery policy within a finite-state machine. With carefully designed learning curricula and by incorporating regularization and adaptive rewards, REBot achieves robust evasion and rapid stabilization in instantaneous DOA tasks. We validate REBot through extensive simulations and real-world experiments, demonstrating notable improvements in avoidance success rates, energy efficiency, and robustness to fast-moving obstacles. Videos and appendix are available on https://rebot-2025.github.io/.**

I. Introduction

Ensuring robot safety during task execution is crucial [1]. In dynamic obstacle avoidance (DOA), when obstacles are slow (reaction time >2 s), a robot can stop and replan a collision-free trajectory using navigation methods [2]–[4]. For legged platforms, this couples high-level planning with low-level locomotion control [5], [6].

When obstacles approach rapidly (reaction time <1.5 s), replanning often fails due to actuation and latency limits [7], [8]. Inspired by vertebrate spinal reflexes, we advocate instantaneous, locality-driven evasive behaviors that bypass slow deliberation [9].

We present the *Reflexive Evasion Robot* (REBot) for quadrupeds, demonstrated on Unitree Go2 [10], [11]. REBot uses a three-stage controller (Normal $\rightarrow$ Avoidance $\rightarrow$ Recovery): a PPO-trained RL policy [12]–[14] executes rapid, safe evasions, and a recovery policy quickly restabilizes the robot and resumes normal tasks.

Trained in Isaac Gym [15] and deployed on hardware, REBot achieves higher avoidance and recovery success than alternatives in both static and dynamic scenarios, while reducing peak joint power and path deviation. Performance varies with approach direction/speed; the platform is particularly effective against frontal threats due to favorable backward maneuvers. Ablations show the necessity of the recovery policy, a two-stage curriculum, and adaptive rewards. Real-world experiments validate real-time, sub-1.5 s evasion and inform safety-system design.

[1] School of Computing, National University of Singapore, Singapore.
[2] Department of Electrical and Computer Engineering, National University of Singapore, Singapore. [3] Shanghai AI Lab.
[*] corresponding to zihao.xu@u.nus.edu

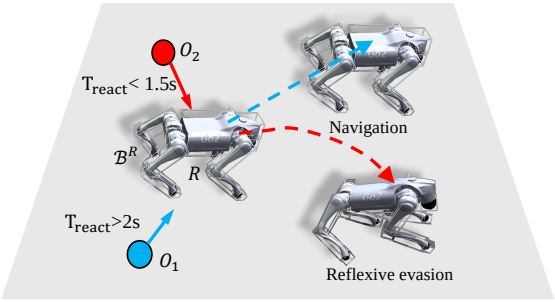


Fig. 1: DOA regimes by reaction time. Red: reflexive evasion at short reaction time; blue: navigation-based avoidance at long reaction time.

Contributions.

- Formalization of reflexive evasion for DOA under constrained reaction time in quadrupeds.
- A three-stage REBot system integrating avoidance and recovery policies for real-time reflexive maneuvers.
- Comprehensive simulation-to-real evaluation with analyses across obstacle directions and speeds.

II. Preliminary

Problem formulation. We consider dynamic obstacle avoidance (DOA) between a dynamic obstacle O and a quadruped robot R (Fig. 1). Obstacles are modeled as spheres with radius r^O and state (p_t^O, v_t^O) in 3D. The robot is a 12-DoF articulated system with state $s_t^R = \{p_t^R, \theta_t^R, v_t^R, \omega_t^R, q_t^R, \dot{q}_t^R, \tau_t^R, f_t^R\}$, and action $a_t^R \in \mathbb{R}^{12}$ (joint targets). A trial is *successful* if no collision occurs over time; we declare collision when the signed distance from obstacle center to the robot's oriented bounding box $\mathcal{B}^R$ satisfies $d(p_t^O, \mathcal{B}^R) < r^O$. Experiments use Unitree Go2.

Avoidance regimes. The controller observes $o_t = \{p_t^R, \omega_t^R, q_t^R, \dot{q}_t^R, \tau_t^R, f_t^R, p_t^O, v_t^O, r^O\}$ and outputs a_t^R to avoid O while maintaining balance. For long reaction time ($T_{\text{react}} > 2$ s), the robot can stop and replan a navigation trajectory (*navigation avoidance*). For short reaction time ($T_{\text{react}} < 1.5$ s), it must execute immediate evasive motions (*reflexive evasion*, Fig. 1). We target the reflexive regime with REBot, and later analyze success, peak joint power, and path deviation in simulation and real-world tests.

III. Method

We propose the **REBot** system for reflexive DOA (Fig. 2). REBot is a three-stage finite-state controller with (i) Normal, (ii) Avoidance, and (iii) Recovery stages. Below we outline stage transitions (Sec. III-A), policies (Secs. III-B–III-C), and training/deployment (Sec. III-D).

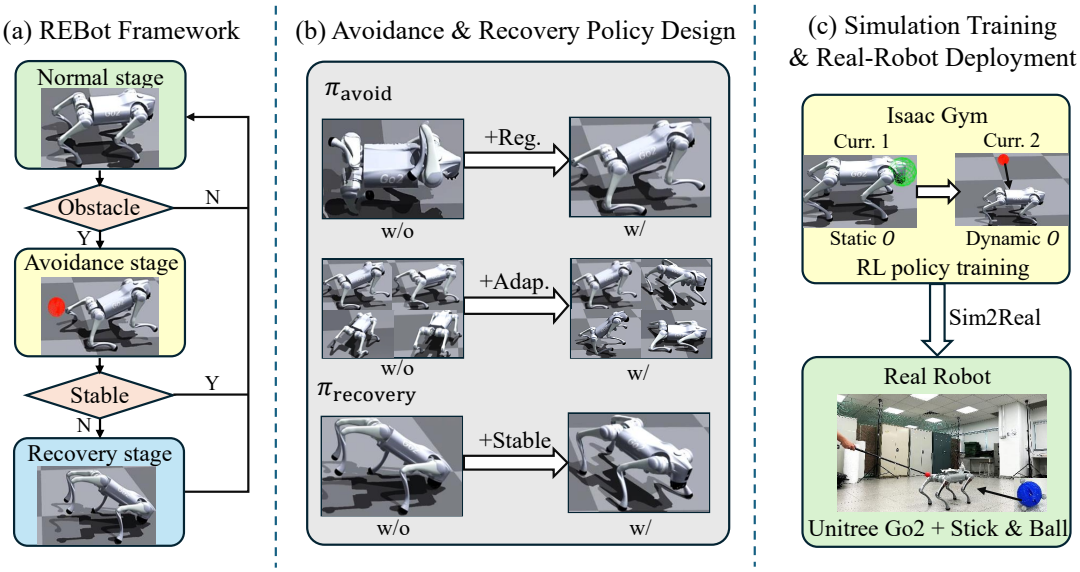


Fig. 2: **(a) REBot framework.** FSM over {Normal, Avoidance, Recovery} governs reflexive evasion against incoming obstacles. **(b) Policy design.** Avoidance uses safety, regularization, and adaptive terms; Recovery stabilizes posture and motion. **(c) Training & deployment.** Two-stage curriculum in Isaac Gym and real-robot tests on Unitree Go2.

A. REBot Stages and Transition Criteria

Normal. The robot maintains a stable posture (standing with a PD controller) while monitoring obstacles.

Avoidance. When an obstacle approaches (e.g., $\langle v_t^O,\ p_t^R - p_t^O\rangle > 0$), REBot switches to *Avoidance* and executes reflexive maneuvers under tight reaction time.

Recovery. If evasive motion induces instability, REBot switches to *Recovery*. Instability is detected if any holds: $\|\theta_t^R\| > \theta_{\text{th}}^R$, $\|\dot{q}_t^R\| > \dot{q}_{\text{th}}^R$, or $h_t^R < h_{\text{th}}^R$. Recovery runs until the robot re-enters the safe set and then returns to *Normal*.

B. Avoidance Policy

We train an RL policy (PPO) for rapid, stable, and energy-aware evasion under constrained reaction time, with reward $r = r_{\text{avoid}} + r_{\text{reg}} + r_{\text{adapt}}$.

Avoidance encourages clearance and penalizes contact, using $r_{\text{avoid}} = -\exp\big(-(d(p_t^O, \mathcal{B}^R) - r^O)\big)$.

Regularization keeps motions natural and efficient: $r_{\text{reg}} = r_{\text{walk}} + r_{\text{energy}} + r_{\text{contact}}$, where r_{walk} rewards diagonal phase consistency (trot-like contacts), $r_{\text{energy}} = -\sum_i |\tau_t^{R,i}\, \dot{q}_t^{R,i}|$ reduces power, and $r_{\text{contact}} = -\sum_i (f_t^{R,i,z} - f_{t-1}^{R,i,z})^2$ suppresses contact jitter.

Adaptive terms promote diversity, speed modulation, and directional efficiency: $r_{\text{adapt}} = r_{\text{div}} + r_{\text{threat}} + r_{\text{dir}}$, where r_{div} encourages varied actions (e.g., action-variance/entropy), $r_{\text{threat}} = -\|v_t^R - v_t^{R,\text{safe}}\|$ with $v_t^{R,\text{safe}} = v_t^{R,\text{cmd}} + \lambda e^{-\eta T_{\text{reaction}}}$, and $r_{\text{dir}} = -\langle v_t^R,\ p_t^O - p_t^R\rangle$ discourages motion to the threat.

C. Recovery Policy

The recovery policy drives the robot back to the safe set and smooth posture, with reward $r = r_{\text{ori}} + r_{\text{stab}} + r_{\text{pos}} + r_{\text{aux}}$, where $r_{\text{ori}} = -\sum_i (\theta_t^{R,i} - \theta_0^{R,i})^2$ penalizes tilt, $r_{\text{stab}} = \sum_i e^{-|\dot{q}_t^{R,i}|}$ encourages low joint speeds, $r_{\text{pos}} = -\|p_t^R - p_0^R\|^2$ penalizes large base deviation, and r_{aux} adds torque and action-smoothness penalties to avoid abrupt motions.

D. Training in Simulation and Real-Robot Deployment

We implement REBot on Unitree Go2 and train policies in Isaac Gym [16] with PPO [17]–[19]. The avoidance and recovery policies are trained with the above curriculum and randomization, then deployed to hardware. For real tests, a motion-capture system provides real-time positions of robot and obstacle; a lightweight ball on a rod serves as a repeatable dynamic-obstacle proxy with reflective markers for tracking.

IV. Simulation Experiments

We validate REBot in simulation and address: **Q1** success under instantaneous DOA (Sec. IV-B); **Q2** reactions under different obstacle conditions (Sec. IV-C); **Q3** the impact of reward design and recovery (Sec. IV-D).

A. Experiment Settings

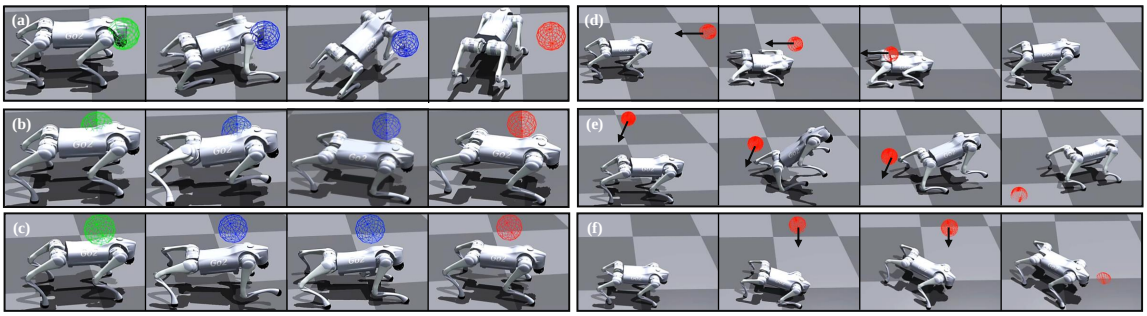
Tasks. We evaluate in Isaac Gym [16]. Obstacles approach within a 180° arc in the robot body frame across XZ/ YZ/ XY planes (frontal, lateral, overhead, ground-level; Fig. 6), with reaction time $T_{\text{react}} \in [0.1, 4.0]$ s to cover instantaneous and delayed regimes.

Metrics. We report five metrics: avoidance success rate $ASR = N_{\text{avoid}}/N_{\text{total}}$; recovery stability $RSR = N_{\text{recover}}/N_{\text{avoid}}$; maximum joint power (MJP); avoidance moving distance (AMD; base displacement between start and end); and gait diversity index $GDI = \mathbb{E}_{s^R\sim\mathcal{D}(s^R)}\big[\text{Var}_{a^R\sim\pi(\cdot|s^R)}(a^R)\big]$.

Baselines. *ABS* [6]: high-speed navigation with static-obstacle robustness, no active DOA; *RRL* [20]: reactive policy for UAV dynamics, not tailored to legged whole-body control.

B. Main Experimental Results

REBot trained with both curricula produces appropriate evasions under varying conditions (Fig. 3): frontal/side threats often trigger jump-away behaviors, while overhead threats elicit crouching.



Curriculum 1: static obstacle Curriculum 2: moving obstacle

Fig. 3: Simulation curricula. **Curr. 1**: static obstacle appears after a delay; **Curr. 2**: moving obstacle with randomized speed/direction. ● Normal, ● Avoidance, ● Obstacle.

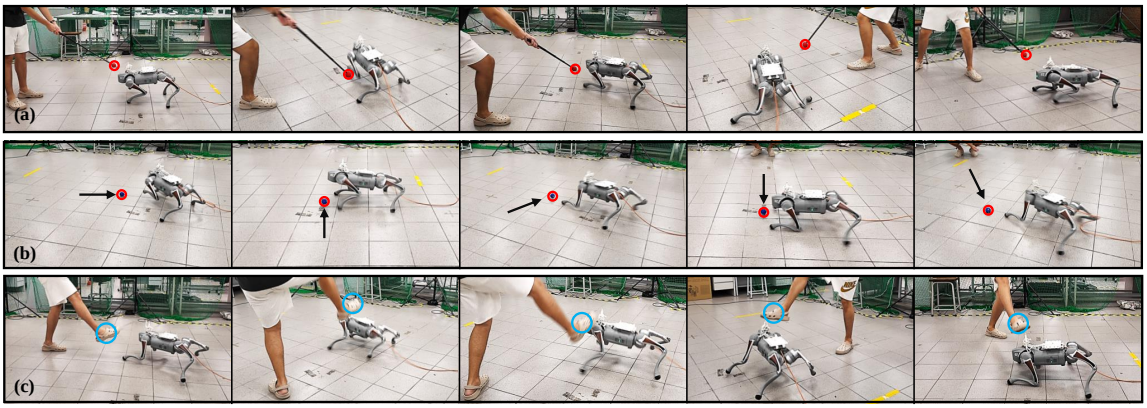


Fig. 4: REBot system real-robot demonstrations on Unitree Go2 Robot (See video). (a) the robot is poked from different directions using a stick; (b) a ball is launched towards the robot from different directions; (c) the robot is subjected to intentional kicks from different directions.

TABLE I: Simulation results across reaction-time ranges.

T_{react} / s	Metric	ABS$^{\diamond}$	RRL$^{\diamond}$	REBot
0.1 ∼ 0.5	ASR↑*	0.00	0.00	**0.05**
	RSR↑*	0.00	0.00	**0.03**
	MJP↓*	0.51	0.52	**0.50**
	AMD↓*	0.84	0.85	**0.82**
0.5 ∼ 1.5	ASR↑	0.11	0.09	**0.65**
	RSR↑	0.06	0.05	**0.59**
	MJP↓	0.52	0.51	**0.49**
	AMD↓	0.80	0.86	**0.47**
1.5 ∼ 4.0	ASR↑	0.51	0.41	**0.81**
	RSR↑	0.42	0.32	**0.74**
	MJP↓	0.40	0.45	**0.34**
	AMD↓	0.60	0.70	**0.26**

* ASR: avoidance success rate; RSR: recovery success rate; MJP: max joint power; AMD: avoidance moving distance.
$^{\diamond}$ ABS: Agile But Safe [6]; RRL: Reactive RL [20].

We analyze three reaction-time bands (Table I, Fig. 5a,b): for $0.1 \sim 0.5$ s, all methods have near-zero ASR/RSR due to insufficient time; for $0.5 \sim 1.5$ s, REBot exhibits reflexive evasion and clearly outperforms ABS/RRL in ASR and RSR; for $1.5 \sim 4.0$ s, all improve but REBot remains best as baselines are not specialized for active DOA. Trends in Fig. 5c,d show that extremely short T_{react} yields high

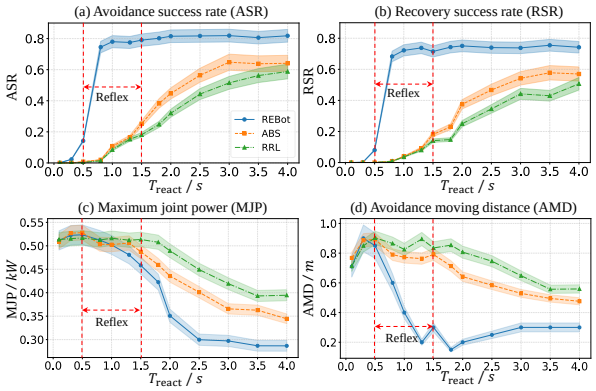


Fig. 5: Performance vs. reaction time. Red dashed region: reflexive regime; long reaction: navigation regime.

MJP (>500 W) and large AMD (jump-like evasions); with moderate T_{react}, both decrease (crouch-like responses); with long T_{react}, values converge lower as navigation behaviors dominate.

C. Analysis of Avoidance Ability

We partition behavior into three regions by ASR/MJP thresholds (boundary I–II: ASR $> 30\%$; boundary II–III: MJP < 300 W). In XZ/XY (Fig. 6a,c), frontal threats are easier, requiring shorter T_{react} to reach navigation; rear threats need longer time. In YZ (Fig. 6b), lateral threats are easier

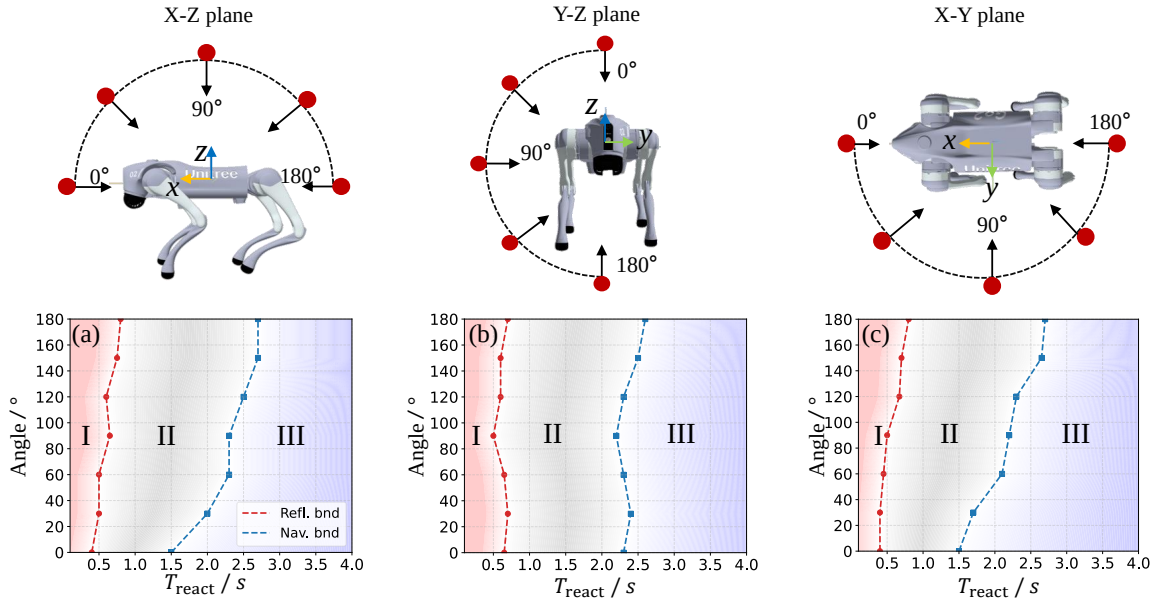


Fig. 6: Directional effects across planes (top: approach directions; bottom: behaviors). Regions: I failure, II reflexive evasion, III navigation avoidance.

TABLE II: Ablations (mid/long T_{react}).

T_{react} / s	Metric	w/o rcv.[1]	w/o curr.[2]	w/o adp.[3]	REBot
0.5∼1.5	ASR↑*	0.63	0.48	0.59	**0.65**
	RSR↑*	0.31	0.39	0.51	**0.59**
	GDI↑*	2.46	2.41	1.43	**2.51**
1.5∼4.0	ASR	0.80	0.71	0.78	**0.81**
	RSR	0.63	0.60	0.69	**0.74**
	GDI	2.06	2.24	1.36	**2.13**

* ASR, RSR, GDI as defined above. [1] no recovery stage; [2] no curriculum-1; [3] no adaptive reward.

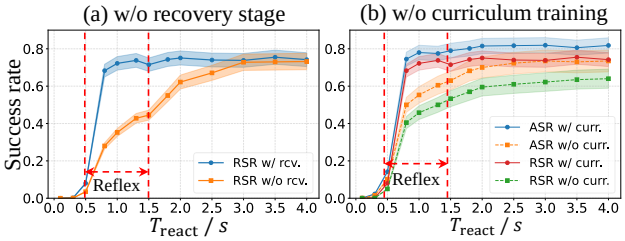


Fig. 7: Ablation success rates vs. reaction time (red dashed: reflexive regime).

than top/bottom. The asymmetry stems from Unitree Go2 mechanics—backward motions are favored over forward jumps.

D. Ablation Studies of REBot System

Recovery. Removing recovery reduces reflex-region success by about 20% (Table II); its influence wanes as T_{react} grows and navigation dominates. We also observe more low-height/fall events and stronger contact-force oscillations, consistent with the stabilization terms in Sec. III-C.

Curriculum. Skipping Curr.1 (directly training on moving obstacles) causes ∼5% ASR/RSR drops (Table II), indicating the staged progression provides better, more stable starts. Without Curr.1, policies overuse aggressive leaps, yielding larger AMD and delayed triggers under fast threats.

Adaptive reward. Removing it reduces GDI by ∼40% and moderately lowers ASR/RSR (Table II), showing that diversity boosts robustness. Failures concentrate on lateral/overhead approaches where non-backward gaits are most beneficial.

V. Real-Robot Demonstration

We deploy REBot on a Unitree Go2 and track robot/obstacle poses with OptiTrack. We evaluate three interaction types—stick poke (Fig. 4a), thrown ball (Fig. 4b), and kick (Fig. 4c)—from multiple directions (front, left/right, left-front, right-front). When an obstacle approaches, REBot enters avoidance to execute reflexive maneuvers (jump-away, crouch), then switches to recovery to restabilize. For slower pokes, the robot often adopts navigation-style avoidance due to the longer available reaction time. Under real-world tests, REBot attains *ASR* 56% and *RSR* 53%. The gap to simulation mainly arises from Sim2Real factors—unmodeled actuator dynamics, control latency, and surface-friction variability—which especially impact fast reflexes requiring precise torque delivery.

VI. Conclusion

We introduce reflexive evasion for quadrupedal DOA, where navigation-based replanning fails under tight reaction times. We present REBot, a unified controller that couples rapid avoidance with stability recovery via reinforcement learning and a staged curriculum. Simulation and hardware results show reliable, adaptive evasions and reveal how morphology and obstacle dynamics shape reflexes, pointing to more agile and safety-aware legged robots.

Limitations remain: (i) we assume precise obstacle pose; integrating on-board perception is future work; (ii) hardware induces a backward-jump bias; (iii) a Sim2Real gap persists—simulation uses joint-velocity commands that miss torque-servo dynamics and ground-friction variability during fast maneuvers.

REFERENCES

[1] F. Shi, C. Zhang, T. Miki, J. Lee, M. Hutter, and S. Coros, "Rethinking robustness assessment: Adversarial attacks on learning-based quadrupedal locomotion controllers," *arXiv preprint arXiv:2405.12424*, 2024.

[2] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter, "Learning a state representation and navigation in cluttered and dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5081–5088, 2021.

[3] T. Dudzik, M. Chignoli, G. Bledt, B. Lim, A. Miller, D. Kim, and S. Kim, "Robust autonomous navigation of a small-scale quadruped robot in real-world environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3664–3671.

[4] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang, "Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers," *arXiv preprint arXiv:2107.03996*, 2021.

[5] X. Huang, Z. Li, Y. Xiang, Y. Ni, Y. Chi, Y. Li, L. Yang, X. B. Peng, and K. Sreenath, "Creating a dynamic quadrupedal robotic goalkeeper with reinforcement learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 2715–2722.

[6] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi, "Agile but safe: Learning collision-free high-speed legged locomotion," *arXiv preprint arXiv:2401.17583*, 2024.

[7] M. Lu, X. Fan, H. Chen, and P. Lu, "Fapp: Fast and adaptive perception and planning for uavs in dynamic cluttered environments," *IEEE Transactions on Robotics*, 2024.

[8] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Robust and versatile bipedal jumping control through reinforcement learning," *arXiv preprint arXiv:2302.09450*, 2023.

[9] T. Umeda, O. Yokoyama, M. Suzuki, M. Kaneshige, T. Isa, and Y. Nishimura, "Future spinal reflex is embedded in primary motor cortex output," *Science Advances*, vol. 10, no. 51, p. eadq4194, 2024.

[10] M. Liu, J. Xiao, and Z. Li, "Deployment of whole-body locomotion and manipulation algorithm based on nmpc onto unitree go2quadruped robot," in *2024 6th International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, 2024, pp. 1–6.

[11] F. Xiao, T. Chen, and Y. Li, "Egocentric visual locomotion in a quadruped robot," in *Proceedings of the 2024 8th International Conference on Electronic Information Technology and Computer Engineering*, 2024, pp. 172–177.

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[13] M. Gurram, P. K. Uttam, and S. S. Ohol, "Reinforcement learning for quadrupedal locomotion: Current advancements and future perspectives," in *2025 9th International Conference on Mechanical Engineering and Robotics Research (ICMERR)*. IEEE, 2025, pp. 28–38.

[14] Z. Xu, A. H. Raj, X. Xiao, and P. Stone, "Dexterous legged locomotion in confined 3d spaces with reinforcement learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 474–11 480.

[15] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[16] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.

[17] X. Han and M. Zhao, "Learning quadrupedal high-speed running on uneven terrain," *Biomimetics*, vol. 9, no. 1, p. 37, 2024.

[18] Y. Zhao, T. Wu, Y. Zhu, X. Lu, J. Wang, H. Bou-Ammar, X. Zhang, and P. Du, "Zsl-rppo: Zero-shot learning for quadrupedal locomotion in challenging terrains using recurrent proximal policy optimization," *arXiv preprint arXiv:2403.01928*, 2024.

[19] J. W. Mock and S. S. Muknahallipatna, "A comparison of ppo, td3 and sac reinforcement algorithms for quadruped walking gait generation," *Journal of Intelligent Learning Systems and Applications*, vol. 15, no. 1, pp. 36–56, 2023.

[20] X. Fan, M. Lu, B. Xu, and P. Lu, "Flying in highly dynamic environments with end-to-end learning approach," *IEEE Robotics and Automation Letters*, vol. 10, no. 4, p. 3851–3858, Apr. 2025. [Online]. Available: http://dx.doi.org/10.1109/LRA.2025.3547306