

The Robot Goes Skateboarding

Danil Belov¹, Elizaveta Pestova¹, Ilya Osokin¹, Xuan Nguyen¹, Pavel Osinenko^{1*}

Abstract— Quadrupedal robots are less energy-efficient on flat surfaces compared to wheeled platforms. Numerous attempts were made to address this issue, including improving control algorithms, adding serial and parallel springs to the robot, and introducing wheels to the system. The presented work belongs to the latter category and specifically addresses quadrupedal robot skateboarding. A single existing work on quadrupedal skateboarding that has practical outcomes is devoted to moving straight only with no controlled turning. In this work an integrated control framework for autonomous quadrupedal skateboarding is presented that achieves energy-efficient locomotion without any hardware modifications to the robot. The proposed approach relies on pushing with one leg for propulsion and weight re-distribution for standard skateboard maneuvering. The desired acceleration is achieved via modulating the leg velocity during ground contact. The proposed approach was validated on Unitree A1 robot, demonstrating successful autonomous navigation around obstacles with significant energy consumption reduction over walking locomotion baseline. A video demonstration is available at <https://youtu.be/l-RUAfImi1g>. A repository of this project is available at <https://github.com/dancher00/quadrupedal-skateboarding>.

I. INTRODUCTION

Modern legged robots are versatile in terms of locomotion, inherently omnidirectional and capable of obstacle traversal. But this comes at a cost of lower energy efficiency on flat surfaces compared to wheeled platforms [1]–[5]. High-torque actuators that are commonly used in quadrupeds suffer from heat losses up to 76% [6]. Attempts to improve this metric can be generally attributed to the following categories.

First, it is the pure control-based approach with the changes to the algorithms only. It could take the shape of introducing an energy consumption term into the minimization objective for MPC-like approaches, or an additional term in the reward function in RL, or a heuristic, such as trying to strengthen the leg while walking [7]–[10]. The common part is the attempt to decrease the energy consumption via only the control function. While very convenient (the robot is not supposed to be mechanically modified and no additional materials should be used), this approach is limited by the low efficiency of the modern high-torque motors and the necessity to rapidly change the contact points of all the legs, which is the fundamental part of walking.

The second group relies on the introduction of elastic elements to the robot [11]–[13]. Serial elastic elements can



Fig. 1: Unitree A1 during skateboarding.

store energy after the impact and release it afterwards, and parallel elastic elements can reduce the average torque via gravity compensation, thus improving overall energy efficiency. This approach can lead to significant energy consumption reduction. However, the main feature of the flat surface (being flat) still remains underutilized.

The third major group also relies on the introduction of novel elements to the system, but this time in order to provide continuous contact point change. The subcategories in this family of approaches include actuated and not actuated wheels, and the wheels themselves being attached and not attached to the robot. Let us discuss these subcategories and place the proposed method in the context of other approaches.

Actuated wheels that are attached directly to the legs comprise the biggest group, and this is the most straightforward approach [14]. The overall system gains a number of degrees of freedom, but all of them are actuated. Given four wheels, a robotic dog becomes equivalent to a four-wheeled robot with overengineered suspension. Thus, the control algorithms in the simplest scenario are fairly straightforward, both the forward motion and turning can be performed in the same way as in a standard wheeled machine. If more sophisticated control algorithms are used, these wheeled quadrupeds are capable of performing impressive acrobatic stunts, using both legs and wheels. An important factor to be considered while designing these robots is the mechanical complexity. Additional wheels add mass, they should be powered, meaning additional wires and connections to the power board, and legs will deviate further from massless, which is a common assumption while obtaining a simplified dynamical model of the dog, see [15], [16].

Passive wheels add considerably less mass and do not

This work was not supported by any organization

¹The authors are with Artificail Intellegence in Dynamic Action (AIDA) research group, Skoltech, Russia {danil.belov, elizaveta.pestova, ilya.osokin, xuan.nguyen, p.osinenko}@skoltech.ru Pavel Osinenko is the corresponding author.

complicate the build as much. However, the robot's passive dynamics should be explicitly considered, as the robot gains several degrees of underactuation [17]–[19]. And these wheels are supposed to have a mechanism to engage and disengage, or they have to be manually installed each time they are needed. One of the possible ways to switch between legged and wheeled locomotion is to place the wheels eccentrically on the foot and rotate the knees, switching them from pointing forward to pointing backward, see [20], [21]. But the energy consumption reduction with all these modifications to the robot can be substantial. ANYmal robot with passive ice skates [22] achieved 80% reduction in the cost of transport compared to conventional locomotion.

In the presented work an approach is considered that relies on the passive wheels that are not attached to the robot's body, i.e. a skateboard. Skateboarding is a locomotion mode requiring no hardware modifications to the robot, combining versatility of the legs and energy efficiency on the flat surfaces of the wheeled robots. Board [23] demonstrated that longboard skateboarding requires approximately 50% less energy than walking through efficient push-and-glide biomechanics.

II. RELATED WORK

Several recent works have made contributions to robot skateboarding. Below an overview is presented of a number of significant works in the area.

Caltech's group developed LEONARDO [24], a bipedal robot capable of performing skateboarding maneuvers using a combination of legs and propellers. Notably, the authors did not implement the robot's push control. The movement was carried out using the directional thrust from the drone's propellers, but not the legs.

Thibault et al. [25] demonstrated skateboarding for humanoid robots through massively parallel reinforcement learning, achieving an efficient cyclic pushing motion with the REEM-C platform using periodic reward formulation that enables one foot to remain on the skateboard deck while the other performs ground-pushing maneuvers for forward propulsion. While their work represents the first attempt at learning skateboarding behaviors for a full-size humanoid on a realistic skateboard simulation, the approach remains constrained by significant limitations that highlight the preliminary nature of these findings. While their work represents the first attempt at learning skateboarding behaviors for a full-size humanoid on a realistic skateboard simulation, the approach is currently limited to simulation environments with forward-only motion and lacks turning or gliding capabilities.

The most significant and successful recent work introduced Discrete-time Hybrid Automata Learning (DHAL) [26], enabling quadrupedal robot Unitree Go1 to perform skateboarding through on-policy reinforcement learning that identifies and executes appropriate mode-switching behaviors. Their approach used a beta policy distribution with a multi-critic architecture to effectively handle contact-guided motions. However, there is still room for improvement. The authors point out that one of the leg end effectors was rigidly

fixed to the board. That significantly simplified both at the learning stage and at the inference of the model on a real robot.

Complementing these learning-based approaches, Xu et al. [27] developed an optimization-based control pipeline for the Xiaomi CyberDog2 platform to achieve dynamic balance and acceleration on a skateboard. The work remains theoretical, proposing a control strategy for a specific robot without an applied implementation.

The article [28] discusses the issue of autonomously mounting a robot on a skateboard using Reinforcement Learning.

Addressing some of the listed research gaps, the **contributions** of this work are as follows.

- A **method** for pushing off with one leg while maintaining balance on a skateboard was **developed**
- Controlled **turning** on a skateboard via mass displacement was **implemented**
- A number of control algorithms were integrated into a framework, capable of planning the path with the kinematics constraints and following it with autonomous pushing and turning, while avoiding obstacles
- An **experimental performance evaluation on a physical robot** was carried out, suggesting energy efficiency improvement of **44.7% reduction in cost of transportation**

III. PROPOSED ALGORITHMS FOR PUSHING AND MANEUVERING

A. Control System Overview

The proposed control framework employs a hierarchical architecture that divides the autonomous skateboarding problem into distinct planning and control layers, as illustrated in Figure 2. At the highest level, the Global Planner generates a coarse trajectory using standard path planning algorithms from the ROS2 Nav2 navigation stack, taking into account the environmental map and obstacle locations. This module produces a sequence of waypoints that define the desired path from start to goal while avoiding static obstacles.

The Local Planner serves as the intermediate layer, adapting the global trajectory to accommodate the skateboard's nonholonomic constraints and kinematic limitations. Implemented as a Stanley controller designed for car-like robots, this module tracks the reference path while generating appropriate target steering angles (θ) and target velocities (v) based on the current robot state (s). The Stanley controller formulation accounts for both cross-track error and heading error, ensuring smooth path following behavior suitable for the skateboard's bicycle-like kinematics.

The lowest control layer consists of two parallel branches that execute the high-level commands through inverse kinematics and joint-level control. The steering branch employs center-of-mass projection and inverse kinematics shifting to convert target steering angles into joint position commands for lateral weight distribution. Simultaneously, the velocity control branch implements the close-loop pushing

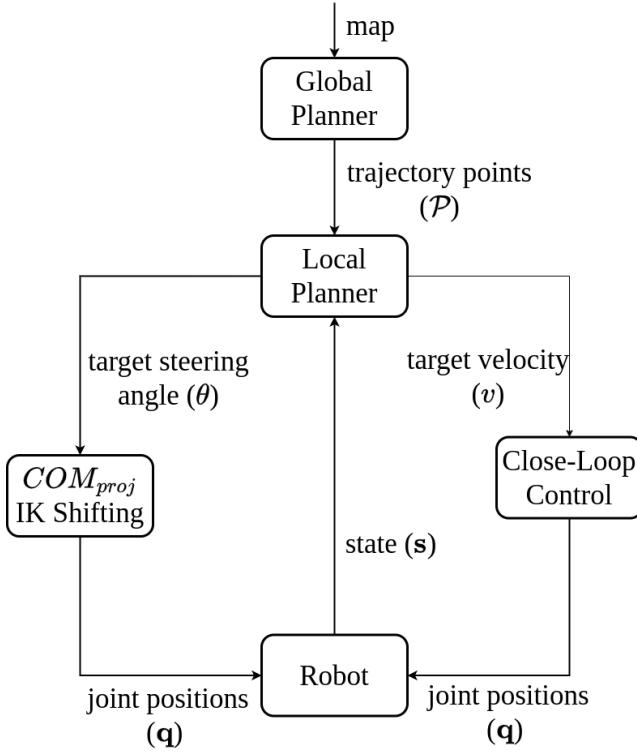


Fig. 2: Path planning system architecture showing the hierarchical control structure. Global planning generates waypoints, local planning adapts them for skateboard kinematics, and the control system executes pushing and steering commands through inverse kinematics and joint-level control.

controller that modulates joint trajectories to achieve the desired forward motion through coordinated leg pushing actions. Both control branches operate at the robot's native control frequency (500 Hz), while the Local Planner updates at a lower rate (50 Hz) suitable for path tracking dynamics. The system maintains feedback through continuous state estimation, enabling real-time adaptation to disturbances and trajectory deviations.

B. Positioning for Push-Off

Before initiating the push-off motion, the robot should transition into a stable configuration that enables ground contact. This involves positioning one leg near the skateboard while maintaining overall balance.

The swing leg is moved along a predefined parabolic trajectory toward the push-off position. The foot trajectory in Cartesian space is defined as:

$$\mathbf{p}_{\text{swing}}(t) = \mathbf{p}_{\text{start}} + \mathbf{v}_0 t + \frac{1}{2} \mathbf{a}_{\text{para}} t^2, \quad (1)$$

where $\mathbf{p}_{\text{start}}$ is the initial position, \mathbf{v}_0 is the initial velocity, and \mathbf{a}_{para} defines the vertical curvature of the trajectory.

In parallel, the robot's CoM executes a multi-stage lateral shift to maintain static stability. Initially, the CoM is centered between the stance legs.

In the first stage, the CoM shifts laterally toward the right side of the board, allowing the front-left leg to be

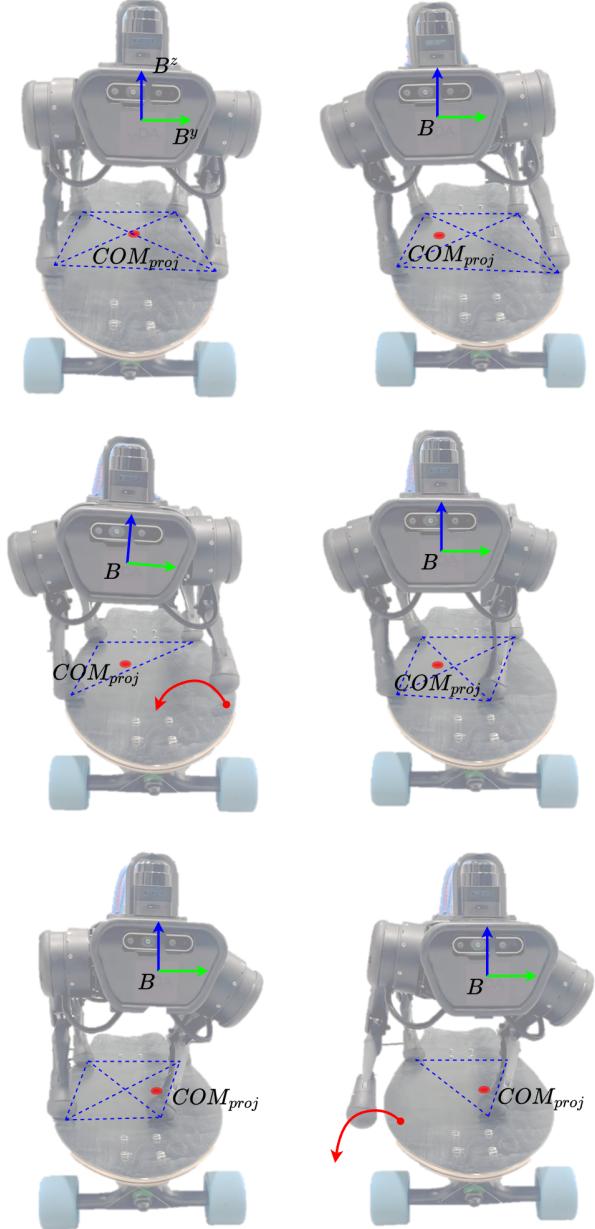


Fig. 3: Sequential frames illustrating the transition from initial stance to push-off preparation. Blue dash lines show support polygon boundaries, and red dot marker represents center of mass projection. Body frame (B), center of mass projection (COM_{proj}) are indicated.

repositioned closer to the center of the skateboard. This adjustment enabling the robot to remain balanced during the later removal of the push-off leg for repositioning.

Before the swing leg lifts off, the torso gradually shifts toward the stance side, increasing normal force under the supporting legs. Once the swing leg approaches the ground, the CoM ensuring that the projection stays within the support polygon throughout the movement (see Fig. 3).

Both the swing leg and torso trajectories are mapped to joint angles using inverse kinematics, ensuring coordinated

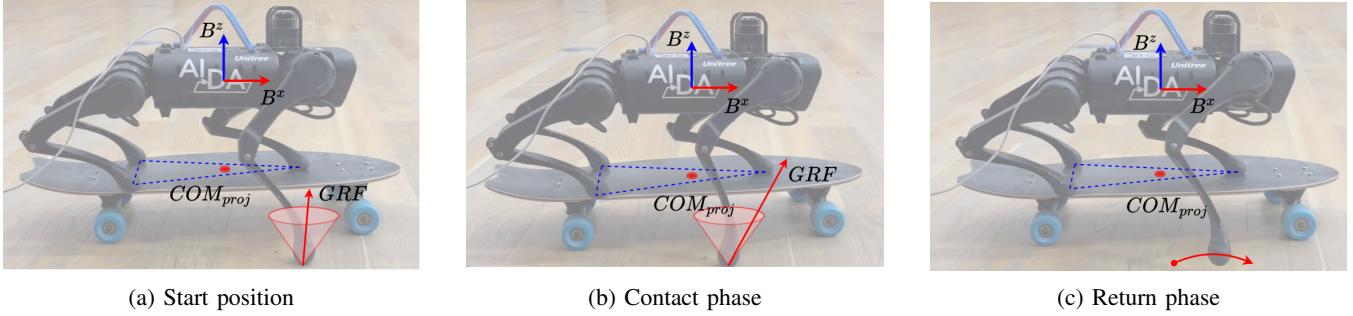


Fig. 4: Sequential frames of robot pushing motion during skateboarding. Body frame (B), center of mass projection (COM_{proj}), and ground reaction force (GRF) are indicated.

full-body motion and stable support during the leg repositioning phase.

C. Velocity-Based Control of Pushing

Forward motion of the quadrupedal robot on the skateboard is achieved through push-off actions performed by a single leg. The control objective during this phase is to maintain a desired velocity of the robot–skateboard system by regulating pushing motion during ground contact.

The leg movement is divided into two distinct phases. During the swing phase, when the leg is not in contact with the ground, its trajectory remains predefined and unaltered. Once ground contact is detected via foot sensors, the system transitions to the stance phase, during which the trajectory is adapted dynamically based on the desired velocity. The core principle of this method is that increased leg speed during contact produces greater impulse, and thus stronger forward acceleration for maintaining the robot’s forward motion.

The control framework for the pushing phase begins with defining a reference velocity profile for the robot’s base, which specifies the desired forward motion over time. A trapezoidal form is used to ensure smooth acceleration and deceleration.

Based on this target velocity, the required foot speed during ground contact is calculated to generate the appropriate push according to the following rule:

$$\mathbf{v}_{\text{push}}(t) = \mathbf{v}_{\text{ref},\text{base}} \cdot (1 + K_v \cdot e(t)), \quad (2)$$

where K_v is the gain, that scales the influence of the velocity error on the foot speed. The error is computed as the difference between the desired and measured foot velocities:

$$e(t) = v_{\text{ref},\text{foot}}(t) - v_{\text{meas}}(t) \quad (3)$$

This foot speed defines a linear trajectory for the leg during the stance phase, where the foot position is updated over time as:

$$\mathbf{p}_{\text{push}}(t) = \mathbf{p}_{\text{push}}(t_0) + \int_{t_0}^t \mathbf{v}_{\text{push}}(\tau) d\tau \quad (4)$$

This Cartesian trajectory is then converted into joint-level commands q_{ref} using inverse kinematics. To prevent

abrupt changes in joint motion and ensure smooth tracking, the desired joint positions are filtered using exponential smoothing:

$$\mathbf{q}_{\text{cmd}}(t) = \alpha \mathbf{q}_{\text{ref}}(t) + (1 - \alpha) \mathbf{q}_{\text{cmd}}(t - \Delta t), \quad (5)$$

where q_{cmd} is commanded joint position and $\alpha \in [0, 1]$ is a smoothing factor.

Finally, these joint trajectories are executed using low-level PD controllers, which track the desired motor positions and ensure smooth and accurate motion during the push-off. For each joint j , the control torque is computed as:

$$\tau_j = K_{p,j} e_j + K_{d,j} \dot{e}_j \quad (6)$$

where $e_j = q_{\text{cmd},j} - q_{\text{act},j}$ is the tracking error, $q_{\text{act},j}$ is the measured joint position from encoders, and $K_{p,j}$, $K_{d,j}$ are the proportional and derivative gains, respectively, which are tuned to provide stable and responsive tracking during dynamic contact with the ground.

D. Steering via CoM Shift

Turning on a skateboard is achieved through lateral shifting of the robot’s center of mass, which creates a turning torque that deflects the trucks and steers the board. The relationship between lateral displacement and turning radius can be derived through torque equilibrium analysis.

When the robot shifts its center of mass laterally by distance l from the skateboard’s centerline, it generates a torque:

$$\tau = mgl \quad (7)$$

where m is the robot mass, g is gravitational acceleration, and l is the lateral displacement.

This torque is balanced by the restoring torque from both truck suspensions:

$$\tau = 2\theta k \quad (8)$$

where θ is the board lean angle, k is the truck stiffness, and the factor of 2 accounts for both front and rear trucks.

From torque equilibrium:

$$mgl = 2\theta k \quad (9)$$

Solving for the lean angle:

$$\theta = \frac{mgl}{2k} \quad (10)$$

The kinematic relationship between lean angle and turning radius, derived from the skateboard geometry with truck mounting angle $\alpha = 45$, is:

$$\rho = \frac{w}{2 \tan(\theta)} \quad (11)$$

where w is the wheelbase.

Substituting the expression for θ :

$$\rho(l) = \frac{w}{2 \tan\left(\frac{mgl}{2k}\right)} \quad (12)$$

For small angles, $\tan(\theta) \approx \theta$, yielding the linearized relationship:

$$\rho(l) \approx \frac{wk}{mgl} \quad (13)$$

This relationship enables direct control of the turning radius through lateral center of mass displacement.

IV. EXPERIMENTAL SETUP AND IMPLEMENTATION

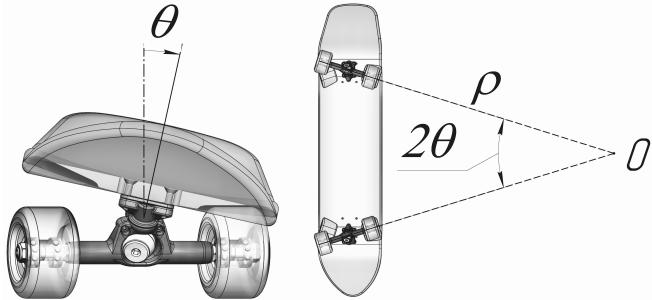


Fig. 5: Linearized Skateboard Model: Rear View (Left) and Bottom View (Right). Parameters: board lean angle (θ), turning radius (ρ), turning center (O).

A. Motion Planning

To ensure balance at all times, the reference torso motion is planned such that the COM of the robot projected onto the ground lies inside the convex hull of the contact points, or support polygon. To apply a pushing force, which accelerates the robot, the normal force of the pushing leg needs to be high enough such that the resultant ground reaction force stays inside the friction cone. The normal force is increased by moving the COM towards the leg that is supposed to start the pushing motion.

Due to the weight of the torso, the normal force increases at the pushing leg. To keep balance after the pushing motion, the COM needs to move towards the next support polygon before the pushing leg lifts off. Quintic polynomial splines are used to define the torso motion [12]. The height of the robot is kept constant during the motion. Since the legs in contact are supporting the motion of the torso, their desired motion is defined by the desired base motion. However, this

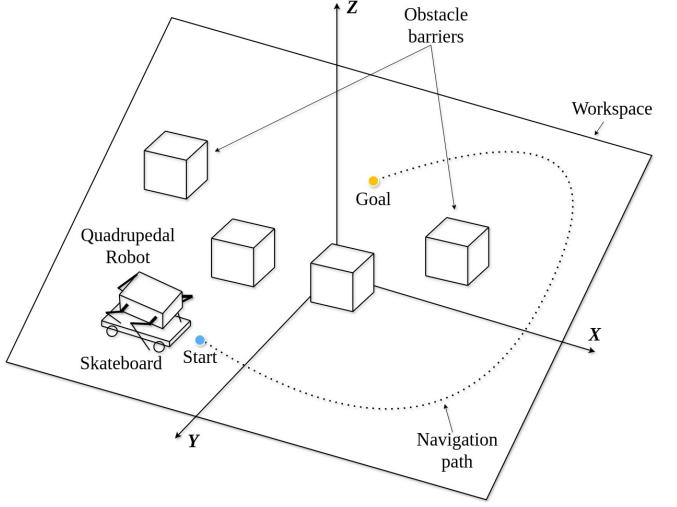


Fig. 6: Problem setup for autonomous quadrupedal skateboard navigation. The robot must navigate from start to goal position while avoiding obstacles (boxes) in the workspace. The path planning system generates feasible trajectories considering skateboard kinematic constraints.

is only valid for the directions in which the skate is able to generate ground reaction forces. In the gliding direction, the skate is not able to generate any ground reaction force which could move the torso in a certain direction. In our case, the robot can only generate forces in the y and z direction of the torso frame B (see Fig. 5).

V. RESULTS

Extensive experiments were conducted to validate the autonomous skateboarding framework using the Unitree A1 quadrupedal robot. The experimental setup includes motion capture for ground truth positioning, force sensors for contact detection, and onboard IMU for real-time state estimation.

A. Pushing Performance

Figure 4 demonstrates the complete pushing motion sequence. The robot successfully maintains balance while executing coordinated leg movements for propulsion. The velocity-based control achieves target speeds with an average tracking error of 0.12 m/s.

The effort analysis in Figure 7 reveals the energy distribution across all four legs during a 30-second continuous pushing motion. The pushing leg (FR) shows periodic high-effort phases during ground contact, while the stance legs maintain consistent moderate effort for stability. The integration of squared effort values enables quantitative energy consumption estimation.

B. Energy Efficiency Comparison

Compared to conventional trotting locomotion at equivalent speeds (0.8 m/s), the skateboarding approach reduces energy consumption by approximately 45%. The energy savings are primarily attributed to the gliding phases between

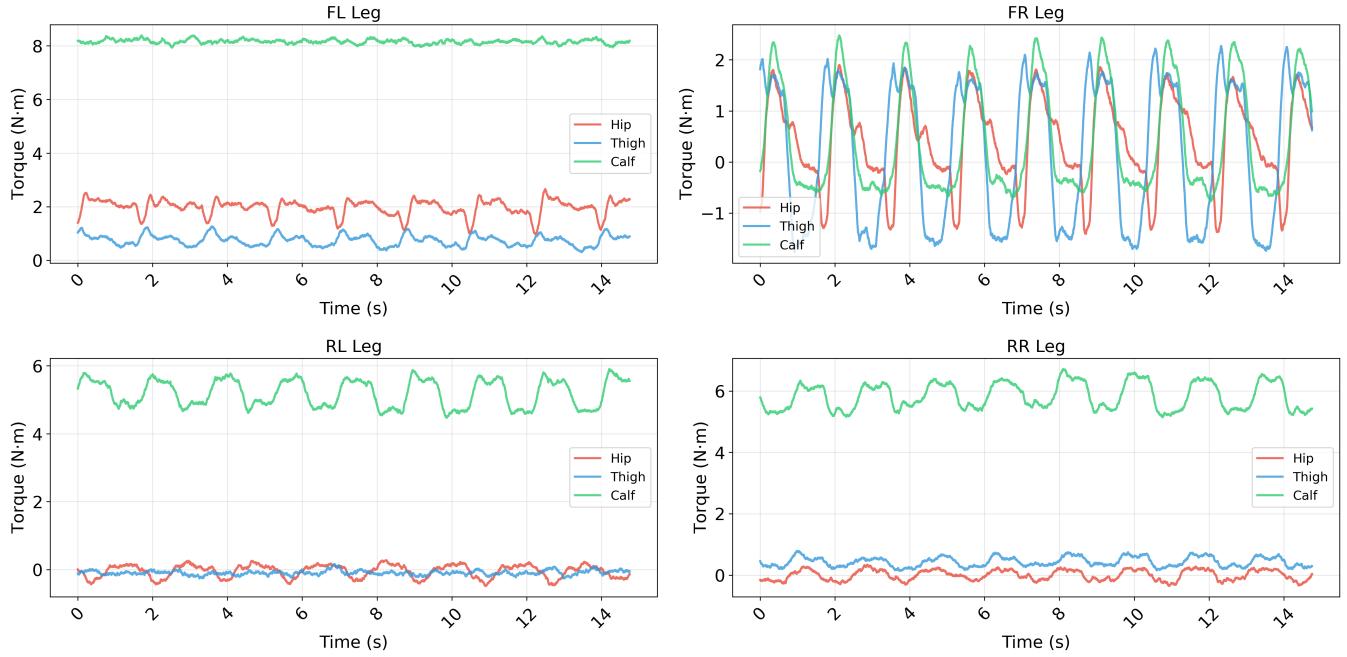


Fig. 7: All Legs - Effort Time Series Comparison. The figure shows effort patterns for hip (red), thigh (blue), and calf (green) joints across all four legs (FL, FR, RL, RR) during the 25-second pushing motion. The time series data enables energy consumption estimation through integration of squared effort values.

push-offs, where the robot maintains forward momentum with minimal actuator effort.

C. Limitations and Challenges

Several limitations affect the current implementation. First, the approach requires relatively smooth surfaces for reliable wheel contact and skateboard stability. Rough terrain or significant surface irregularities can disrupt the pushing motion and compromise balance.

Second, the system's performance depends on accurate contact detection, which can be affected by sensor noise and surface variations. Future work should investigate more robust contact sensing methods, potentially incorporating vision-based ground detection.

Third, the current steering mechanism relies on lateral CoM shifts, which limits the maximum turning rate. Sharp turns require multiple maneuvers or stopping to reposition, reducing overall navigation efficiency.

D. Comparison with Related Work

The proposed approach advances beyond previous skateboarding research in several key aspects. Unlike LEONARDO [24], which relied on propellers for propulsion, the presented method achieves skateboarding through pure leg-based pushing. Compared to the DHAL approach [26], our framework operates without rigid foot attachment to the board, representing a more realistic and versatile solution.

The energy efficiency improvements align with findings from the ANYmal ice skating work [22], confirming that hybrid locomotion modes can significantly reduce transport costs for legged robots on flat surfaces.

VI. CONCLUSION

Several directions emerge for future development. Enhanced surface adaptation could extend the approach to rougher terrains through adaptive contact strategies and improved balance control. Integration of visual perception would enable better obstacle detection and path planning in dynamic environments.

Advanced learning algorithms could optimize the pushing patterns for different skateboard configurations and surface conditions. Multi-robot coordination for skateboarding teams presents interesting challenges for formation control and collaborative navigation.

Long-term autonomy requires investigation of battery management strategies specific to skateboarding locomotion, potentially including energy harvesting during downhill gliding phases.

This work establishes autonomous quadrupedal skateboarding as a viable hybrid locomotion mode, achieving significant energy efficiency improvements without hardware modifications. The integrated control framework successfully combines pushing, steering, and navigation capabilities, demonstrated through comprehensive experimental validation on the Unitree A1 robot.

The results confirm that skateboarding can transform quadrupedal robots into efficient wheeled platforms for energy-critical applications while maintaining the inherent versatility of legged locomotion. This hybrid approach opens new possibilities for adaptive robot mobility in structured environments.

REFERENCES

- [1] S. Hirose, “A study of design and control of a quadruped walking vehicle,” *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 113–133, 1984.
- [2] R. Kurazume, A. Byong-Won, K. Ohta, and T. Hasegawa, “Experimental study on energy efficiency for quadruped walking vehicles,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 1. IEEE, 2003, pp. 613–618.
- [3] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, “Bigdog, the rough-terrain quadruped robot,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10 822–10 825, 2008.
- [4] X. Xiao and W. R. L. Whittaker, “Energy considerations for wheeled mobile robots operating on a single battery discharge,” Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-14-16, August 2014.
- [5] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, “Energy-efficient motion planning for mobile robots,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 5. IEEE, 2004, pp. 4344–4349.
- [6] N. Kau, A. Schultz, N. Ferrante, and P. Slade, “Stanford doggo: An open-source, quasi-direct-drive quadruped,” in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6309–6315.
- [7] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, “Fast and efficient locomotion via learned gait transitions,” in *Conference on Robot Learning*. PMLR, 2022, pp. 773–783.
- [8] W. Zhu and A. Rosendo, “Psto: learning energy-efficient locomotion for quadruped robots,” *Machines*, vol. 10, no. 3, p. 185, 2022.
- [9] R. Zashchitin and D. Dobriborsci, “A study on the energy efficiency of various gaits for quadruped robots: Generation and evaluation.” in *ICINCO (1)*, 2023, pp. 311–319.
- [10] T. Hao, D. Xu, and S. Yan, “Quadrupedal locomotion in an energy-efficient way based on reinforcement learning,” *International Journal of Control, Automation and Systems*, vol. 22, no. 5, pp. 1613–1623, 2024.
- [11] N. Kashiri, A. Abate, S. J. Abram, A. Albu-Schaffer, P. J. Clary, M. Daley, S. Faraji, R. Furnemont, M. Garabini, H. Geyer *et al.*, “An overview on principles for energy efficient robot locomotion,” *Frontiers in Robotics and AI*, vol. 5, p. 129, 2018.
- [12] F. Bjelonic, J. Lee, P. Arm, D. Sako, D. Tateo, J. Peters, and M. Hutter, “Learning-based design and control for quadrupedal robots with parallel-elastic actuators,” *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1611–1618, 2023.
- [13] D. Belov, A. Erkhov, F. Khabibullin, E. Pestova, S. Satsevich, I. Osokin, P. Osinenko, and D. Tsetserukou, “Optimizing energy consumption for legged robot by adapting equilibrium position and stiffness of a parallel torsion spring,” *arXiv preprint arXiv:2411.18295*, 2024.
- [14] M. Bjelonic, C. D. Bellicoso, Y. de Viragh, D. Sako, F. D. Tresoldi, F. Jenelten, and M. Hutter, “Keep rollin’—whole-body motion control and planning for wheeled quadrupedal robots.” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2116–2123, 2019.
- [15] X. He, X. Li, X. Wang, F. Meng, X. Guan, Z. Jiang, L. Yuan, K. Ba, G. Ma, and B. Yu, “Running gait and control of quadruped robot based on slip model,” *Biomimetics*, vol. 9, no. 1, 2024.
- [16] N. Wolf-Sonkin, “Simulation-based analysis of quadrupedal energetics through kinematic and dynamic modeling,” Master’s thesis, The Cooper Union for the Advancement of Science and Art, 2025.
- [17] J. Chen, K. Xu, and X. Ding, “Roller-skating of mammalian quadrupedal robot with passive wheels inspired by human,” *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 3, pp. 1624–1634, 2020.
- [18] J. Chen, R. Qin, L. Huang, Z. He, K. Xu, and X. Ding, “Unlocking versatile locomotion: A novel quadrupedal robot with 4-dofs legs for roller skating,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8037–8043.
- [19] G. Endo and S. Hirose, “Study on roller-walker—improvement of locomotive efficiency of quadruped robots by passive wheels,” *Advanced Robotics*, vol. 26, no. 8-9, pp. 969–988, 2012.
- [20] W. Li, L. Wei, and X. Zhang, “A wheels-on-knees quadruped assistive robot to carry loads.” *Applied Sciences*, vol. 12, no. 18, p. 9239, 2022.
- [21] T. Sun, X. Xiong, Z. Dai, and P. Manooppong, “Small-sized reconfigurable quadruped robot with multiple sensory feedback for studying adaptive and versatile behaviors,” *Frontiers in neurorobotics*, vol. 14, p. 14, 2020.
- [22] M. Bjelonic, C. D. Bellicoso, M. E. Tiriyaki, and M. Hutter, “Skating with a force controlled quadrupedal robot,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7555–7561.
- [23] W. J. Board and R. C. Browning, “Self-selected speeds and metabolic cost of longboard skateboarding,” *European journal of applied physiology*, vol. 114, pp. 2381–2386, 2014.
- [24] K. Kim, P. Spieler, E.-S. Lupu, A. Ramezani, and S.-J. Chung, “A bipedal walking robot that can fly, slackline, and skateboard,” *Science Robotics*, vol. 6, no. 59, p. 8136, 2021.
- [25] W. Thibault, V. Rajendran, W. Melek, and K. Mombaur, “Learning skateboarding for humanoid robots through massively parallel reinforcement learning,” *arXiv preprint arXiv:2409.07846*, 2024.
- [26] H. Liu, S. Teng, B. Liu, W. Zhang, and M. Ghaffari, “Discrete-time hybrid automata learning: Legged locomotion meets skateboarding,” *arXiv preprint arXiv:2503.01842*, 2025.
- [27] Z. Xu, M. Al-Khulaifi, H. Ma, J. Wang, Q. Xin, Y. You, M. Zhou, D. Xiang, and S. Zhang, “Optimization based dynamic skateboarding of quadrupedal robot,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8058–8064.
- [28] B. Danil, E. Artem, P. Elizaveta, O. Ilya, T. Dzmitry, and O. Pavel, “Quadrupedal robot skateboard mounting via reverse curriculum learning,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.06561>