Unified Planning and Reinforcement Learning Control for Quadruped Robots in Off-Road Environments

Ji Ma
Department of Mechanical Engineering
The University of Hong Kong
Hong Kong, SAR, China
maji@connect.hku.hk

Zeren Luo
Department of Mechanical Engineering
The University of Hong Kong
Hong Kong, SAR, China
zerluo@connect.hku.hk

Peng Lu*
Department of Mechanical Engineering
The University of Hong Kong
Hong Kong, SAR, China
lupeng@hku.hk

***Abstract*—Trajectory planning for quadrupedal robots in complex unknown environments is an extremely challenging task due to the need to maintain balance, stability, and safe interaction with unstructured terrain while navigating efficiently. Existing methods often decouple planning and control, relying on computationally expensive environmental representations, or struggling with non-convergence in intricate scenarios. This paper presents a novel reinforcement learning(RL)-based approach that tightly integrates spatio-temporal trajectory planning and control for quadrupedal robots. The proposed method is validated on a RL-based locomotion controller which is tailored to challenging terrains. By unifying optimization-based planning and RL-based control in a unified framework, the quadrupedal robot can execute tasks intelligently while making real-time adjustments based on environmental feedback, resulting in improved overall performance and robustness. The proposed framework paves the way for robust and efficient navigation of legged robots in complex, unstructured, and off-road environments.**

***Index Terms*—Off-road Autonomy; Trajectory Planning; Reinforcement Learning**

I. Introduction

For quadrupedal robots, trajectory planning in unknown complex environments is an extremely challenging task. Compared to aerial robots such as unmanned aerial vehicles (UAVs), quadrupedal robots not only need to satisfy balance and stability constraints during motion but also must consider the interaction and contact with complex terrains, avoiding collisions or instability. Therefore, quadrupedal robots need to plan an optimal trajectory that can reach the target point while ensuring stability, to improve navigation efficiency and energy utilization.

In recent years, researchers have proposed numerous trajectory-planning algorithms for robot navigation, aiming to generate optimal trajectories. However, most existing algorithms are more suitable for robot systems that do not require ground contact, such as UAVs. Although these algorithms can be simply modified for application to quadrupedal robots, the resultant planning results often exhibit abnormalities or sub-optimal performance, failing to meet the special requirements of quadrupedal robots.

*Corresponding author.

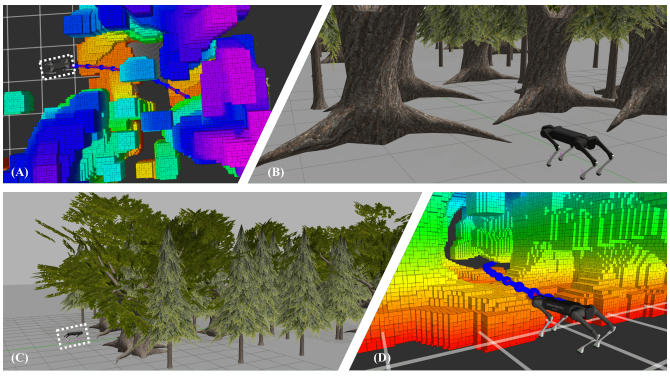

Fig. 1. The snapshot of a planning procedure in a random simulation environment of massive trees

Therefore, developing a trajectory planning algorithm specifically designed for quadrupedal robots has significant theoretical value and application prospects.

A. Vision-based Obstacle Avoidance

Vision-based obstacle avoidance control algorithms play a crucial role in autonomous navigation and path planning. These algorithms aim to generate safe and efficient paths by leveraging visual information, enabling robots to achieve autonomous navigation capabilities in complex environments [1]–[3]. However, many existing trajectory planning algorithms construct Euclidean Signed Distance Fields (ESDF) maps [4] or safe corridors [5] based on visual perception to constrain the optimization of trajectory generation, which typically consumes a significant amount of planning time. To achieve real-time obstacle avoidance, some researchers attempt to eliminate these additional operations by directly incorporating the information as constraints in a gradient-free manner [6], improving planning speed and efficiency with satisfactory results. Nevertheless, these gradient-based

planners may face non-convergence or planning failures when dealing with complex environments.

B. Learning-based Planner

In recent years, with the flourishing development of reinforcement learning(RL), utilizing learning-based planners to solve path-planning problems has become a popular research direction. RL is a learning method based on Markov decision processes, where an intelligent agent observes states and outputs behaviors that maximize the total expected reward. Numerous research efforts focus on combining RL algorithms with classical global or local motion planning modules to achieve relatively optimal results [7]–[10]. These algorithms are typically used as supplements to classical algorithms but share the same limitation of insufficient robustness. Simultaneously, some end-to-end approaches directly take sensor data as input and control the robot to reach its destination. These works often employ deep learning techniques but still face significant challenges when deployed in the real world.

C. Gap between Planning and Control

To handle complex environments and tasks, many works discretize the robot's state space. This approach simplifies the problem but results in discontinuous and non-smooth planned paths. This means that the robot may exhibit disjointed actions when executing the planned path, affecting its performance and effectiveness [11].

Furthermore, previous works often decouple planning and control execution. While this decoupling reduces system complexity, it also induces sub-optimal executed trajectory. This implies that the robot may not achieve the expected performance level in the actual environment, leading to a sim2real gap [12]. In the summary, the contribution of this work can be listed as follows:

- Proposed a novel optimization-based approach for spatio-temporal planning of trajectories in unknown, off-road environments. This approach leverages the power of an RL-based pre-existing locomotion controller to generate optimized trajectories without relying on prior knowledge of the environment.
- By utilizing a RL control policy, the trajectory planning process becomes more efficient and effective, leading to improved performance in terms of speed and accuracy, especially in challenging off-road scenarios.
- Integrated planning and control in a unified framework, enabling the robot to execute tasks in unknown environments, resulting in improved overall performance and robustness in off-road autonomy.

II. Method

A. Overview

An overview of our system is illustrated in Fig 2. Given a pre-established low-level locomotion policy, we develop a planner that generates high-frequency velocity commands to be tracked in a hierarchical structure. We first obtain sensor information from the real robot, which can be accessed in

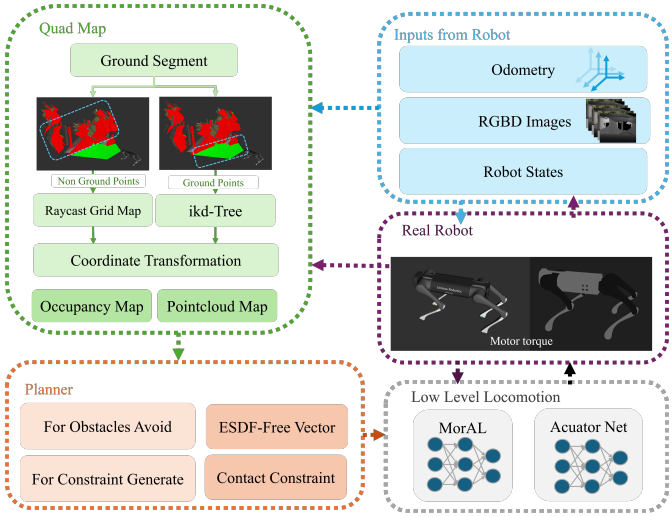

Fig. 2. Overview of the proposed pipeline

both simulation and real-world platforms. Assuming at the i-th step t_i, we can obtain the odometry messages $\mathcal{O}_i$ and camera messages $\mathcal{C}_i$. Based on this information, we can acquire the point cloud transformed by the depth camera. Then, we apply ground segmentation algorithms, specifically the Patchwork++ algorithm [13], to obtain two concerned point cloud classes $\mathcal{C}_a = \{\mathcal{C}_g, \mathcal{C}_n\}$, where $\mathcal{C}_g$ indicates the ground cloud used to build a point cloud map using an ikd-tree [14], and $\mathcal{C}_n$ indicates the non-ground segment used to build a grid map. We then pass the output to the planner module, which generates trajectories as discussed in the next section. The RL controller subsequently generates the torque commands $\boldsymbol{\tau}$ to drive the motors. After completing this pipeline, we obtain the desired outcome.

B. Trajectory Definition

In robot motion planning, trajectory generation is a critical step to ensure smooth robot motion. For a quadruped robot, we simplify the planning problem by focusing on the trajectory planning of its base in the $x, y, z,$ yaw state space, without considering obstacle avoidance for the time being. The initial trajectory is generated using the minimum jerk method [2]. The resulting trajectory is a piecewise four-dimensional fifth-order polynomial with M segments, and the i-th segment P_i can be represented as:

$$P_i = C_i^T \boldsymbol{\beta}(t - t_{i-1}), \quad t \in [t_{i-1}, t_i] \tag{1}$$

where $\boldsymbol{\beta}(t)$ is the vector of fifth-order basis functions, and C_i is the coefficient matrix for the i-th segment. The minimum jerk trajectory ensures smooth continuity at the boundary conditions while minimizing the trajectory curvature change, thereby avoiding abrupt robot motions.

To further optimize the trajectory, we introduce the $MINCO$ [15] parameterization method. By turning the trajectory coefficient matrices C and segment durations T =

$[T_1, \ldots, T_M]^T$ to optimization variables q, T, we can formulate the following unconstrained optimization problem:

$$\mathbf{c} = M(\mathbf{q}, \mathbf{T}) \tag{2}$$

where $\mathbf{c} = [\mathbf{c}_1^T, ..., \mathbf{c}_M^T]^T$ collects the coefficients of all segments, q represents the intermediate waypoints, and each segment's time cost is T, the MINCO optimization problem can be formulated as:

$$\min_{q,T} J = \int_0^{T_t} \|\ddot{\boldsymbol{p}}(t)\|^2 \,\mathrm{d}t + \rho T_t \quad T_t = \sum_{l=1}^{M} T_l \tag{3}$$

We use a method similar to [3] to model the obstacle avoidance penalty J_o as:

$$Jo = \sum_{i=0}^{\kappa} \max\{(C_o - d_o(p(t_i))), 0\}^3 \tag{4}$$

where $d_o(\cdot)$ is a distance function that depends on the trajectory $p(t)$, and therefore J_o is a function of the MINCO trajectory. C_o represents the clearance threshold for obstacle avoidance in the grid map.

We also incorporate additional penalties J_* and solve this optimization problem efficiently using unconstrained optimization algorithms such as L-BFGS.

C. Deep Reinforcement Learning(DRL)-based Controller

In our previous work [16], we trained a general DRL-based control policy that uniformly considers the robot's structural variation and terrain diversity. The model takes the observation $\mathcal{O}_t \in \mathbb{R}^{45}$ as input, which includes the body angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3$, projected gravity $\boldsymbol{g}_t \in \mathbb{R}^3$, body linear velocity command $\boldsymbol{v}_t^* \in \mathbb{R}^3$, joint angles $\boldsymbol{q}_t \in \mathbb{R}^{12}$, joint velocities $\dot{\boldsymbol{q}}_t \in \mathbb{R}^{12}$, and the action from the previous time step $\boldsymbol{a}_{t-1} \in \mathbb{R}^{12}$, which can be written as:

$$\mathcal{O}_t = [\boldsymbol{\omega}_t, \boldsymbol{g}_t, \boldsymbol{v}_t^*, \boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \boldsymbol{a}_{t-1}] \tag{5}$$

The output of the DRL policy is then directly applied to the joint-level PD controller of the joint-level actuation module, i.e., $\boldsymbol{\tau} = \boldsymbol{K}_p \cdot (\boldsymbol{q}_t^* - \boldsymbol{q}_t) + \boldsymbol{K}_d \cdot (-\dot{\boldsymbol{q}}_t)$, where $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are the proportional and derivative gains, respectively.

The DRL controller can handle various velocity commands, while the classical planner can generate smooth trajectories with high responsiveness. By combining these two methods, we can obtain a highly novel planner for navigation tasks that leverages the strengths of both approaches. The DRL controller provides robust and adaptive control policies, while the the planner ensures smooth and responsive trajectory generation.

III. Experiment

A. Experimental Setup

We utilized IsaacGym as the simulator environment for training our model. Recent work has already generated a diverse set of environments with varying tracks and obstacles [16]. The model was trained on a desktop computer equipped with an Nvidia RTX 4070 GPU for more than 6000 episodes, allowing for extensive training iterations.

To evaluate the performance of our algorithm, we generated multiple testing environments using the Gazebo simulator. The first testing scenario was a random forest environment comprising 70 pine trees and 30 bark obstacles within a 15m $\times$ 50m area. This environment aimed to assess the algorithm's capability to navigate through cluttered and unstructured environments. Additionally, we have environments with a 15m $\times$ 50m random forest with 100, 150, and 200 pine trees.

The second testing scenario was derived from a highly challenging competition [17], where our algorithm was evaluated on a 6m $\times$ 10.32m runway. This scenario provided a controlled and structured environment to test the algorithm's performance under constrained conditions, mimicking real-world applications such as autonomous navigation in urban or industrial settings.

By leveraging diverse testing environments, including both unstructured and structured scenarios, we aimed to comprehensively evaluate the robustness, adaptability, and generalization capabilities of our algorithm across a wide range of potential real-world applications.

B. Experiment Results

Based on our algorithm, we first tested our planner's performance with both the Model Predictive Control (MPC) controller and the DRL controller. Subsequently, we evaluated our entire planner's performance against some widely used planners.

1) Controller Comparisons: In the pine tree forest environment, we evaluated the performance of our method against two controllers: the MPC controller and the DRL controller shown in Fig 3, 4. Our method demonstrated superior capabilities in traversing exposed roots without slipping, resulting in higher mobility and reduced time consumption compared to the other controllers. Additionally, our method exhibited improved velocity trackability, enabling more accurate and responsive control.

2) Plan Quality Evaluation: We assessed the plan quality of our method by testing it in both the random forest and pine forest environments which are shown in Fig 5. The results include metrics related to trackability and trajectory smoothness. These evaluations provide insights into the algorithm's capability to generate high-quality plans that adhere to the desired trajectories while navigating through diverse and challenging terrain conditions.

IV. Conclusion

In this work, we presented a novel framework for integrated spatio-temporal trajectory planning and control tailored for

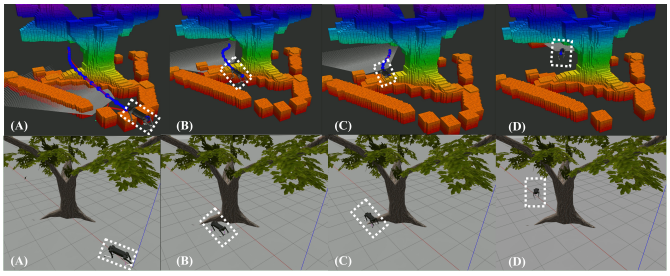

Fig. 3. Collision-free navigation of a quadruped robot to the goal using the proposed method

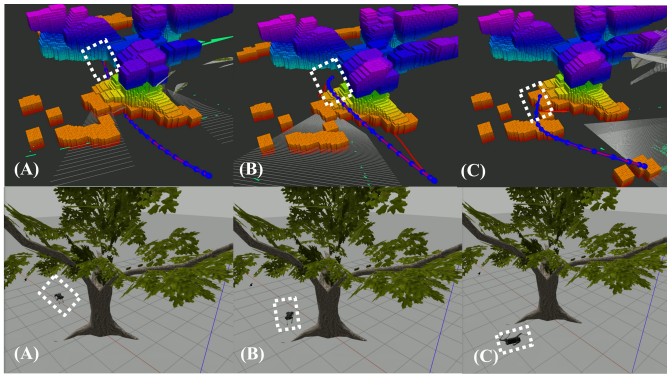

Fig. 4. Quadruped robot navigation using MPC fails due to execution error

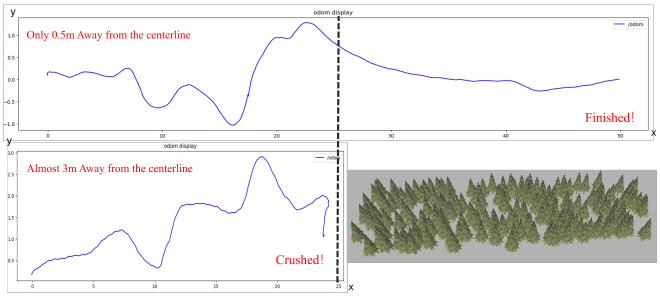

Fig. 5. Long-distance tracking performance - proposed method completed 50m without off-tracking, while MPC failed before 25m

quadrupedal robots in complex, unstructured off-road environments. By leveraging the power of RL techniques, our approach generates optimized normalized trajectories that enable dynamic adaptation to environmental constraints while ensuring balance, stability, and safe terrain interaction.

Future research directions include incorporating more sophisticated perception modules, extending to multi-robot coordination scenarios, and exploring applications in challenging domains.

REFERENCES

[1] F. Gao, W. Wu, Y. Lin, and S. Shen, "Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 344–351.

[2] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2520–2525.

[3] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu, and F. Gao, "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm5954, 2022. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abm5954

[4] F. Gao, Y. Lin, and S. Shen, "Gradient-based online safe trajectory generation for quadrotor flight in complex environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3681–3688.

[5] J. Chen, T. Liu, and S. Shen, "Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1476–1483.

[6] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2021.

[7] Z. Luo, E. Xiao, and P. Lu, "Ft-net: Learning failure recovery and fault-tolerant locomotion for quadruped robots," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8414–8421, 2023.

[8] U. Patel, N. K. S. Kumar, A. J. Sathyamoorthy, and D. Manocha, "Dwa-rl: Dynamically feasible deep reinforcement learning policy for robot navigation among mobile obstacles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6057–6063.

[9] P. Rousseas, C. Bechlioulis, and K. J. Kyriakopoulos, "Harmonic-based optimal motion planning in constrained workspaces using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2005–2011, 2021.

[10] L. Chang, L. Shan, C. Jiang, and Y. Dai, "Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment," *Auton. Robots*, vol. 45, no. 1, p. 51–76, jan 2021. [Online]. Available: https://doi.org/10.1007/s10514-020-09947-4

[11] J. Ye, D. Batra, A. Das, and E. Wijmans, "Auxiliary tasks and exploration enable objectgoal navigation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 16 097–16 106.

[12] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot, "Navigating to objects in the real world," *Science Robotics*, vol. 8, no. 79, p. eadf6991, 2023. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.adf6991

[13] S. Lee, H. Lim, and H. Myung, "Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3d point cloud," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 276–13 283.

[14] Y. Cai, W. Xu, and F. Zhang, "ikd-tree: An incremental K-D tree for robotic applications," *CoRR*, vol. abs/2102.10808, 2021. [Online]. Available: https://arxiv.org/abs/2102.10808

[15] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," 2022.

[16] Z. Luo, Y. Dong, X. Li, R. Huang, Z. Shu, E. Xiao, and P. Lu, "Moral: Learning morphologically adaptive locomotion controller for quadrupedal robots on challenging terrains," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4019–4026, 2024.

[17] J. Jeon, "Icra2024 quadruped robot challenges," 2024. [Online]. Available: https://github.com/teamgrit-lab/ICRA2024_Quadruped_Robot_Challenges