2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3503-7770-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/IROS58592.2024.10802606

A Fast Motion and Foothold Planning Framework for Legged Robots on Discrete Terrain

Jiyu Yu[1], Dongqi Wang[2], Zhenghan Chen[1], Ci Chen[1], Shuangpeng Wu[1], Yue Wang[1], and Rong Xiong[1]

***Abstract*— Legged robot proved their capability to cross complex terrain in recent research, yet the autonomy of robots on discrete terrain still needs to be enhanced since it requires a full stack framework. This paper introduces a real-time motion and foothold planning framework tailored for legged robots navigating uneven terrains, such as stepping stones. Our approach addresses the critical challenges of determining feasible global paths and local footholds to enhance autonomous mobility across complex landscapes. By using a sampling-based global path planner integrated with terrain segmentation and the robot's kinematic model, our framework swiftly generates viable navigation paths. Concurrently, it utilizes a Mixed Integer Programming (MIP) methodology for real-time foothold optimization, ensuring the robot's stability and safety through dynamic terrain interaction. Finally, an execution layer including Model Predictive Control (MPC) and Whole-Body Control (WBC) generates the robots' motion. Simulation and real-world experiments demonstrate that our framework improves legged robots' adaptability on discrete terrains.**

I. INTRODUCTION

Legged robots, unlike wheeled robots, inherently possess the capability to navigate discrete terrains due to the biomimetic design of their leg structures. This feature establishes a significant relationship between the base pose and foothold position, especially when traversing through environments such as stepping stones. Dealing with this scenario involves solving two primary issues: determining the global path ("Where to go?") and identifying local footholds ("Where to step?").

Specifically, the question "Where to go?" entails generating a collision-free global path, enabling the legged robot to identify feasible steps along the path [1]. The primary goal is for the robot to traverse this path and safely reach its destination. However, due to the numerous Degrees of Freedom (DoFs) involved, the planning process often results in lengthy computation times or poses challenges in finding solutions [2]. Also, assessing the traversability of a path, especially in challenging terrain, remains an open issue as it is closely tied to the robot's structure and control strategy.

On the other hand, "Where to step?" focuses on determining feasible footholds locally. Employing heuristic footholds is one strategy [3], [4], though it introduces no additional computational complexity. However, due to the absence of comprehensive consideration, these footholds often fall short of optimality and might even precipitate balance problems. An alternative strategy encompasses an optimization-based approach [5], [6], The key challenge here lies in effectively integrating terrain data and the robot's kinematics model to identify foothold points that ensure both safety and stability. In addition, real-time planning is crucial for real-world applications, this requires the algorithm to run as fast as possible.

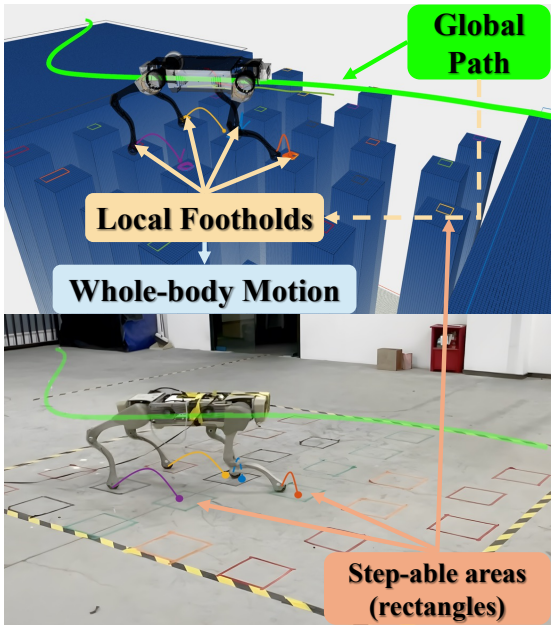


Fig. 1. Legged robot X20 trots across the stepping stones scenario. In the above figure, we send pre-build virtual terrain to the robot and the robot generates global path, local footholds, and whole-body motion autonomously. The following figure shows the real-world experiment, step-able areas are marked with colored rectangles.

Moreover, the above-mentioned studies [5], [4], [7] have primarily concentrated on tackling the challenge of "where to step". However, the lack of efficient solutions for the "where to go" problem hampers the autonomous mobility of legged robots, particularly in complex and discrete terrains where legged robots are indispensable. In reality, these two questions are intricately connected and demand a comprehensive approach.

To address both of these challenges simultaneously, we introduce a fast and real-time motion planning framework for legged robots designed for navigating through discrete terrains. Specifically, the framework incorporates a sampling-

This work was supported in part by the National Science and Technology Major Project of China under Grant 2021ZD0114504 and in part by the National Nature Science Foundation of China under Grant 62373322.

[1]Jiyu Yu, Zhenghan Chen, Chen Ci, Shuangpeng Wu, Yue Wang, and Rong Xiong are with the State Key Laboratory of Industrial Control and Technology, Zhejiang University, Hangzhou 310027, China. Yue Wang is the corresponding author (e-mail: wangyue@iipc.zju.edu.cn).

[2]Dongqi Wang is with the Department of Engineering Mechanics, Zhejiang University, Hangzhou 310027, China.

based global path planner that employs the robot's kinematic model and segmented terrain data. This planner swiftly generates a global path utilizing the terrain information segmentation. Moreover, operating under the quasi-static assumption, the planner also checks the path states to ensure feasible solutions during the foothold planning phase. Subsequently, a MIP-based local foothold planner, guided by the global path, optimizes foothold positions by formulating a disjunctive constraints problem. The execution of motion is then handled by the MPC and WBC, as outlined in our previous work [8]. This integrated methodology boosts the overall efficiency and adaptability of legged robots dealing with online navigation in environments featuring discrete terrain elements (Fig. 1).

Our contributions can be summarized as follows:

- We extend the legged robot global path planner to discrete terrain, including challenging scenarios such as stepping stones, and achieve a higher success rate.
- We propose a MIP-based foothold local planner. It comprehensively considers the kinematic relationship between the four legs and the center of the base and provides the optimal foothold.
- By integrating the global path with heuristic footholds as an initial guess, we limit the MIP problem's scope, thus running faster and can be deployed in real-world robot systems.
- Comprehensive simulation and hardware experiments in multiple scenarios have verified the effectiveness of our framework.

II. Related Works

A. Global path planning

The field of robotics has extensively explored motion planning for legged robots. The Trajectory Optimization (TO) method, as demonstrated in [9] and [10], initially showcased impressive outcomes by formulating a complex nonlinear programming problem that spans from the starting state to the end, leading to the determination of the global trajectory and footholds. Despite the reliability and elegance of the optimization outcomes, practical implementation is hindered by lengthy optimization durations, often extending to several minutes or even hours.

In contrast, Sampling-Based Planners (SBPs) offer a notable advantage in terms of time efficiency. For instance, [11] introduces a framework for scoring traversability and planning paths for legged robots, emphasizing the evaluation of terrain attributes like slope, roughness, and steps. Building upon this concept, [12] develops a rapid global path planner using filtered terrain data and a simplified robot model. However, as terrains become increasingly intricate, the effectiveness of simplistic terrain scoring approaches diminishes. There arises a need to integrate the robot's kinematics to account for foothold safety, reachability, and stability in a comprehensive evaluation of path traversability.

In this work, SBP is selected for path planning due to its effectiveness in navigating non-convex environments. To tackle the high DoFs challenge, the approach involves reducing the sampled DoFs through terrain data and employing the quasi-static assumption. This allows for a meticulous assessment of feasibility along the path and facilitates rapid global path generation. Recent work [2] shares similarities with the framework we propose in terms of the path planner. However, the main difference is that they integrate path planning and foothold planning into one planner for a long horizon. While this enhances the quality of the global path, the lack of an optimization mechanism for footholds may lead to unstable steps for the robot.

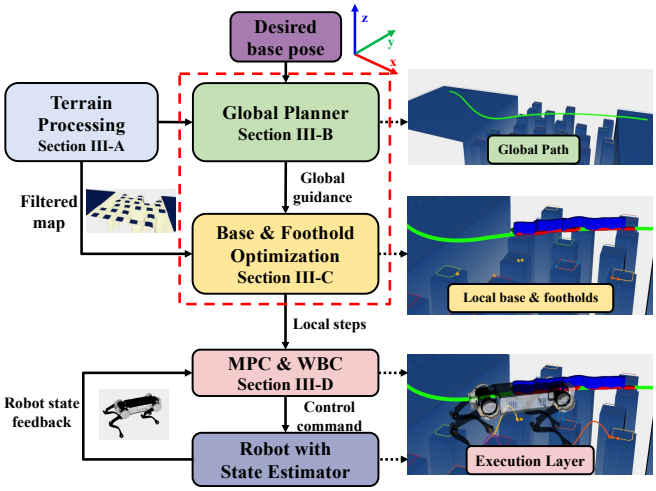


Fig. 2. Architectural overview of proposed motion planning pipeline. The red box highlights the key modules in this work.

B. Footholds planning

The selection of footholds is pivotal for enhancing robot mobility across diverse terrains. Heuristic methods are highly effective on flat surfaces [3]. For complex terrains like stairs and stepping stones, strategies from [4] use heuristic formulas to select the nearest terrain segment. However, these strategies often overlook the critical interplay between the robot's four legs and its Center of Mass (COM).

Recently, optimization-based approaches have surpassed heuristic and sampling methods in popularity. This trend is attributed to their ability to identify optimal solutions under specific criteria, rather than focusing merely on feasibility [7]. For instance, [13] outlines a method for generating footholds by conducting batch searches on the *grid_map* [14]. Furthermore, studies like [5], [9], [15] enhance foothold optimization by incorporating them as continuous variables in the trajectory optimization process and applying gradient-based methods. While in scenarios where the available stepping areas are fragmented, foothold planning should be reformulated to accommodate disjunctive constraints and typically solved through MIP. Despite the theoretical advantages of MIP for handling multi-contact challenges, its practical application is hindered by extensive computation times, often due to branch-and-bound techniques. The study in [6] employs contact planning on humanoid robots by simplifying a MIP into a linear programming format, thereby running faster. However, the approximation may lead to footholds that are not always in safe areas. Our approach

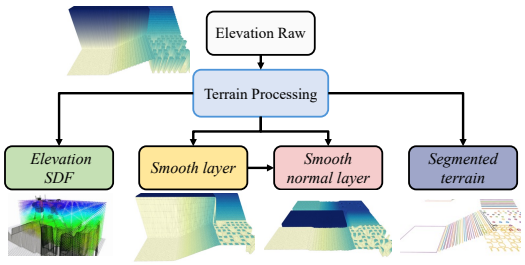


Fig. 3. Main output for terrain processing, *Elevation SDF* for collision validation, *Smooth layer* and *Smooth normal layer* for base pose, and *Segmented terrain* for feasibility validation of footholds.

integrates global path data with heuristic footholds to inform the MIP problem, ensuring accurate placement of footholds in safe zones and expediting the MIP solution.

III. ARCHITECTURAL OVERVIEW

An overview of the locomotion pipeline is presented in Fig. 2. The terrain processing module (Section IV-A) takes the raw elevation map as input and outputs the filtered map and segments step-able region of the map, then global planner (Section IV-B) tries to plan a path from robot current state to the goal state. Once the path is achieved, a MIP-based foothold planner (Section IV-C) is used to optimize the next 3 gait cycles (all footholds in about 3s, depending on the gait). The new footholds with the contact time are sent to MPC to optimize trajectories (1s) of swing legs and COM. Finally, WBC solves the joint torque according to optimized trajectories and sends it to the legged robot (Section IV-D).

IV. METHODOLOGY

A. Terrain Processing

The terrain processing module is based on *grid_map* [14], it takes the raw elevation map as input and four layers shown in Fig. 3. The *Smooth layer* is a Gaussian filtered [5] version of the original map, providing a highly smooth surface that serves as the virtual floor for the robot. The *Smooth normal layer* is normal information of the *Smooth layer*, these two layers decide the robot's height, roll, and pitch angle. The *Elevation SDF* is a signed distance field in 3D used in collision checking during path planning, it is computed by the open-source library *grid_map* [14]. Finally, the *Segmented terrain* is key to our global planner and local foothold planner, it checks for roughness, flatness, and edges of the map and gets the safe region on the map.

B. Global Planner

With the robot model and the terrain awareness, the online global planner is ready to plan a path using the Rapidly-exploring Random Tree (RRT) based method. The whole-body state of a robot is defined as the set $\mathcal{S}$:

$$\mathcal{S} : \{\boldsymbol{s} = [\boldsymbol{q}_b, \boldsymbol{q}_j]^{\mathrm{T}} \in SE(3) \times \mathbb{R}^{n_j}, \boldsymbol{q}_{j_{min}} \leq \boldsymbol{q}_j \leq \boldsymbol{q}_{j_{max}}\}$$

where the $\boldsymbol{q}_b$, $\boldsymbol{q}_j$ denotes the base pose in the world frame (i.e. its position and orientation) as well as the angles of n_j actuated joints. Note that the joint angles are constrained by the limitations of joint space.

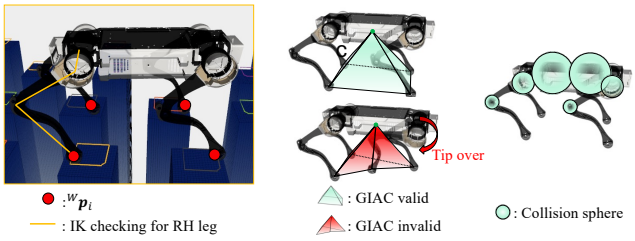


Fig. 4. Global planner state checker. Left: we validate the feasibility of foothold the terrain by IK. Middle: The Convex GIAC criterion, rejects unstable configuration. Right: The approximated collision sphere with different radii.

1) Sampling: The primary challenge associated with the SBP approach is the inefficiency of sampling within whole-body state spaces. To mitigate this, robot models and terrain data are integrated to create a mapping from the robot's $SE(2)$ state to its comprehensive whole-body state.

By sampling the robot's x-y position and orientation in yaw, represented as (x, y, γ), the robot's height is determined by adding the predefined height to the *Smooth layer*'s height at that point. The *Smooth normal layer* is utilized to determine the pitch and roll angles of the floating base, aligning it parallel to the terrain. Then the floating base's 6D pose in the world frame, $\boldsymbol{q}_b$ is determined.

For the remaining states, $\boldsymbol{q}_j$, represent the angles of the joints, despite the potential for multiple configurations at a specific base pose, only the configuration with the closest default foothold matching the terrain is considered. Following the floating base's 6D pose, $\boldsymbol{q}_b$, and predefined default footholds in the base frame, ${}^{B}\boldsymbol{p}_i$, for each leg (where $i \in 1, 2, 3, 4$ corresponds to Left Front (LF), Right Front (RF), Left Hind (LH), and Right Hind (RH) legs), the positions of these footholds in the map frame, ${}^{M}\boldsymbol{p}_i$, are calculated. The nearest point on the *Segmented terrain* to ${}^{M}\boldsymbol{p}_i$ is selected as the foothold for this 6D pose, indicated by ${}^{W}\boldsymbol{p}_i$. Consequently, a whole-body state tuple $\{\boldsymbol{q}_b, {}^{W}\boldsymbol{p}_1, {}^{W}\boldsymbol{p}_2, {}^{W}\boldsymbol{p}_3, {}^{W}\boldsymbol{p}_4\}$ in the world frame is obtained by merely sampling on $SE(2)$ pose of the robot. Note that this whole-body state is considered the most likely rational configuration, we don't check other footholds as this is a trade-off of efficiency.

2) Expansion and State Checker: When the RRT tree expands at $SE(2)$ space, the newly valid state will be connected to the rest of the tree in a certain way. We adopted a cubic spline between two states, as it's smooth on speed. The spline curve between two states is discretized by 0.5% of its length (maximum length is set to 5m), and these states are checked for validity by the state checker.

For the state checker in our planner, the key is to ensure that the output global path is executable by the legged robot. To realize this, not only the collision-free nature of the path needs to be considered, but also the path's feasibility of robots' footholds. The state checker focuses on four aspects.

- ***Feasibility of the footholds***: Each foothold in the state tuple $\{\boldsymbol{q}_b, {}^{W}\boldsymbol{p}_1, {}^{W}\boldsymbol{p}_2, {}^{W}\boldsymbol{p}_3, {}^{W}\boldsymbol{p}_3\}$ is checked whether it violates any joint limits by calculating Inverse Kinematics (IK) analytically, shown on the left of Fig. 4.

- ***Convex GIAC***: Gravito-Inertia Acceleration Cone (GIAC) [5] is used as a dynamic stability criterion in the TO planner. Here, with the quasi-static assumption, it can be constructed as the kinematics constraint of base position and footholds, shown in the middle of Fig. 4 and Eq. (1).
- ***Collision-Free Path***: We focus on the hip and thigh joints of legged robots, which are most prone to collisions with the environment. The *Elevation SDF* is employed to calculate the closest distance between these joints and any obstacles. The collision sphere is shown on the right of Fig. 4.
- ***Terrain Slope***: Considering that the base pitch and roll are determined by the normal of the terrain's *smooth* layer, these angles are limited to a specific range to prevent the robot from tipping over.

Specifically, during the feasibility checking of the footholds, the i-th foothold ${}^{W}\boldsymbol{p}_i$ is transformed into the hip frame and gets the joint's angles of i-th leg. Due to the analysis of the IK of a single 3DoF leg, the feasibility checking can run very fast. Any result that exceeds the joint limits $\{\boldsymbol{q}_{min}, \boldsymbol{q}_{max}\}$ is rejected. The convexity of GIAC is verified by checking the following inequality. Note the ^{W} is omitted:

$$\begin{aligned} &\boldsymbol{p}_{B1} \times \boldsymbol{p}_{B3} \cdot \boldsymbol{p}_{B2} \geq 0, \quad \boldsymbol{p}_{B1} \times \boldsymbol{p}_{B3} \cdot \boldsymbol{p}_{B4} \geq 0, \\ &\boldsymbol{p}_{B1} \times \boldsymbol{p}_{B2} \cdot \boldsymbol{p}_{B3} \leq 0, \quad \boldsymbol{p}_{B1} \times \boldsymbol{p}_{B2} \cdot \boldsymbol{p}_{B4} \leq 0, \\ &\forall i, \quad p_B^z > p_i^z \end{aligned} \tag{1}$$

where $\boldsymbol{p}_B$ denotes base position and $\boldsymbol{p}_{Bi} = \boldsymbol{p}_i - \boldsymbol{p}_B$, denotes the edge of the cone. p^z is coordinates in the z-axis. Take the first equation as an example, it means footholds of RF $\boldsymbol{p}_2$ should be the right side of the plane defined by $\boldsymbol{p}_B, \boldsymbol{p}_1, \boldsymbol{p}_3$ when looking from the x-axis of the base, other legs are the same. Then the footholds should not be higher than the base otherwise it will cause tip over. Collision check can be used for the IK result in the feasibility checking of footholds (if valid). The centers of the spheres $\boldsymbol{c}_i$ used to approximate the hip and thigh joints can be computed by the kinematics model of the robot, then c_i should be outside of obstacles, i.e. $sdf(\boldsymbol{c}_i) \geq r$, where r_i denotes the radius of the sphere.

Note that the global planner checks every single discrete state on the connecting path based on the quasi-static assumption. despite not requiring the legged robot to maintain a full stance on the ground at every state. This thorough examination ensures that states along the path are capable of supporting the robot in a full stand, making sure the lower-level local foothold planner can find solutions. Additionally, this approach does not constrain the robot's gait sequence, as the path planner has no idea when and which leg should touch down on the ground.

Once the path is successfully planned, the time interval between two states is determined by dividing the distance between them by a default velocity. The global trajectory of the base is denoted by $\boldsymbol{c}_{ref}(t)$.

C. Base position and Foothold Planner

Imagine that when humans move from one point to a destination, we often consider a path first, and then think locally about how to place our feet and adjust the position of our bodies. Inspired by this, the foothold planner is treated as a local planner running online.

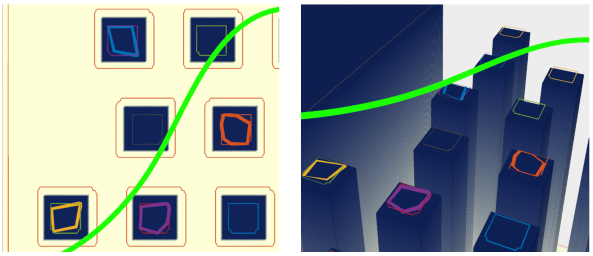


Fig. 5. The regions represented by $\{\boldsymbol{S}_{i,j}^k, \boldsymbol{s}_{i,j}^k\}$. Blue, orange, yellow, and purple are the safe contact constraints. The green line in the middle is the global path. Left: Top view. Right: 3D view.

1) Base trajectory: The planner takes the global planners' result as a reference and initial guess to accelerate the solving process. We parameterize the base position as a B-spline curve of degree d on the piece-wise polynomial function $B_{i,d}(t)$.

$$\boldsymbol{c}(t) = \sum_{i=0}^{d} B_{i,d}(\frac{t}{T})\boldsymbol{C}_i \tag{2}$$

where $\boldsymbol{C} = [\boldsymbol{c}_0 \quad \boldsymbol{c}_1 \quad ...]$ are the $d-1$ control points [16].

2) Safe contact constraints: The safe region on the terrain general is not continuous, such as in stair or stepping-stone scenes, where accessible regions are discretized into pieces as shown in Fig. 5. Here a linear inequality for the footholds $\{\boldsymbol{S}_{i,j}^k, \boldsymbol{s}_{i,j}^k\}$, coming from convex polygons inside *Segmented terrain*, is a half-space constraint represents the k-th safe footholds region of leg i segmented plane in step j. It means $\exists k, \boldsymbol{S}_{i,j}^k \boldsymbol{p}_{i,j} \leq \boldsymbol{s}_{i,j}^k$, which can be written as problems with disjunctive constraints and solved by the MIP solver. By using Big-M formulation [7], the safe contact constraint can be rewritten as follows:

$$\forall i,j,k, \quad \boldsymbol{S}_{i,j}^k \boldsymbol{p}_{i,j} \leq \boldsymbol{s}_{i,j}^k + M(1 - a_{i,j}^k); \sum_{k=0}^{n} a_{i,j}^k = 1 \tag{3}$$

where $M \gg 0$ represents a significantly large number, the binary decision variable $a_{i,j}^k \in \{0,1\}$ indicates the selected region. For instance, if $a_{i,j}^1 = 1$, then $\boldsymbol{S}_{i,j}^1 \boldsymbol{p}_{i,j} \leq \boldsymbol{s}_{i,j}^1$ and for other constraints $\{\boldsymbol{S}_{i,j}^k, \boldsymbol{s}_{i,j}^k\}(k \neq 1)$ are automatically satisfied since M is a very large number and those constraints are inactive. Finally, the sum of $a_{i,j}^k$ equals 1 suggests exactly one region is selected.

3) Footholds constraints: Different from [7], we consider the base position and can constraint the footholds in a reachable region, given by:

$$||\boldsymbol{c}(t_{i,j}) - \boldsymbol{p}_{i,j}||^2 \leq l_{max} \tag{4}$$

where $\boldsymbol{c}(t_{i,j})$ is the base position when leg i touch down at step j, l_{max} is the maximum distance between footholds and base centre. Using such an approximation avoids non-linear forward kinematics, but at the same time achieves good results. Another important constraint is leg collision avoidance, we limit the minimum distance between the footholds of different legs:

$$\forall i \neq m, \quad ||\boldsymbol{p}_{m,j} - \boldsymbol{p}_{i,j}||^2 \geq l_{min} \tag{5}$$

Additionally, we also reject the results that require robot takes a large step.

$$\forall j > 1, \quad ||\boldsymbol{p}_{i,j} - \boldsymbol{p}_{i,j-1}||^2 \leq h_{max} \tag{6}$$

4) Cost: For the base spline trajectory, a core task is to follow the result of a global planner. The whole trajectory is divided into n_k segments of equal time intervals and tasks those nodes to calculate the tracking cost with weights w_1.

$$\min_{\mathbf{c}} \sum_{k=0}^{n_k} ||\boldsymbol{c}(\frac{t_k}{T}) - \boldsymbol{c}_{ref}(\frac{t_k}{T})||_{w_1}^2 \tag{7}$$

Second, for the footholds, there are three quadratic cost terms in the problem. The first term attempts to regularise footholds around Raibert's heuristic [17]. The heuristic footholds are given by:

$$\boldsymbol{p}_i^* = \boldsymbol{p}_{i,nom} + \sqrt{\frac{h}{g}}(\boldsymbol{v}_{B,cur} - \boldsymbol{v}_{B,com}) \tag{8}$$

where $\boldsymbol{p}_i^*$ denotes the heuristic foothold and $\boldsymbol{p}_{i,nom}$ represents the nominal foothold located directly below the hip, which can be readily determined with the known global path. And h is the nominal height of the robot, g is the gravitational constant. $\boldsymbol{v}_{B,meas}$ and $\boldsymbol{v}_{B,com}$ are measured and commanded base velocity respectively. The second term is approximate Zero Moment Point (ZMP) cost which drives the base position close to the center of the four landing points to enhance stability.

$$\min_{\boldsymbol{c},\boldsymbol{p}} ||\boldsymbol{p}_{i,j} - \boldsymbol{p}_{i,j}^*||_{w_2}^2 + ||\boldsymbol{c}^{x,y}(t_{i,j}) - \sum_{i=1}^{4} \boldsymbol{p}_{i,j}^{x,y}/4||_{w_3}^2 \tag{9}$$

5) Solve the MIP with global path: When traversing flat and simple terrain, it is unnecessary to consider all contact surfaces. Leveraging the global planner's output and Eq. (8), nominal footholds are initially identified, followed by the addition of safe contact constraints $\{\boldsymbol{S}_{i,j}^k, \boldsymbol{s}_{i,j}^k, a_{i,j}^k\}$ within a specific distance range. This strategy significantly enhances the solving process by minimizing the potential contact surfaces, effectively reducing the number of binary decision variables required. In the most ideal situation, where only one safe contact region is identified in proximity, the MIP problem simplifies into a Quadratically Constrained Quadratic Programming (QCQP) problem.

Combining all the constraints and costs, the base position and foothold planner solve the following problem:

$$\begin{aligned}
\min_{\boldsymbol{C},\boldsymbol{P},\boldsymbol{a_{i,j}}} \quad & \textstyle\sum_{k=0}^{n_k} ||\boldsymbol{c}(\frac{t_k}{T}) - \boldsymbol{c}_{ref}(\frac{t_k}{T})||_{w_1}^2 + ||\boldsymbol{p}_{i,j} - \boldsymbol{p}_{i,j}^*||_{w_2}^2 + \\
& + ||\boldsymbol{c}^{x,y}(t_{i,j}) - \textstyle\sum_{i=1}^{4} \boldsymbol{p}_{i,j}^{x,y}/4||_{w_3}^2 \\
\text{s.t.} \quad & \textstyle\sum_{k=0}^{n} a_{i,j}^k = 1, \quad \boldsymbol{a}_{i,j} \in \{0,1\}^n \\
& \forall i,j,k, \quad \boldsymbol{S}_{i,j}^k \boldsymbol{p}_{i,j} \leq \boldsymbol{s}_{i,j}^k + M(1 - a_{i,j}^k) \\
& \qquad ||\boldsymbol{c}(t_{i,j}) - \boldsymbol{p}_{i,j}||^2 \leq l_{max} \\
& \forall i \neq m, \quad ||\boldsymbol{p}_{m,j} - \boldsymbol{p}_{i,j}||^2 \geq l_{min} \\
& \forall j > 1, \quad ||\boldsymbol{p}_{i,j} - \boldsymbol{p}_{i,j-1}||^2 \leq h_{max}
\end{aligned}$$

In this planner, the base orientation is deliberately omitted to prevent the introduction of nonlinearity into the problem, thereby simplifying the solving process.

D. MPC & WBC

A nonlinear MPC approach is employed here to optimize the trajectories by incorporating the robot's dynamic model and foothold constraints. The robot is modeled using Single Rigid Body Dynamics (SRBD) [18]. In this context, soft constraints introduced in [4] ensure that the robot's footholds adhere to its reference position derived from the MIP above. For the swing legs, the spline is generated between the reference footholds with a desired liftoff and touch down velocity. The WBC uses a full dynamics model of the legged robot to track the optimized trajectory. Essentially, this involves solving a Quadratic Programming (QP) problem to determine optimal torques, taking into account various safety constraints and tracking costs. For more details on MPC and WBC, please refer to our previous work [8].

V. Experiments

A. Setup and Solvers

All the modules in the proposed framework are implemented in C++ programming language with ROS [19] as a communication middleware. The global planner is implemented and tested based on OMPL [20] which offers a convenient interface to benchmark our algorithm. The MIP problem is solved by Gurobi [21]. The MPC is executed via the OCS2 library [22], and robot dynamics is computed by Pinocchio library [23] in MPC and WBC. Finally, the QP problem in WBC is solved using qpOASES [24]. All the simulation experiments are completed on the AMD R7-5800H laptop.

B. Global Planner Success Rate Evaluation

1) On different terrains: Our global planner is evaluated on four different terrains with three difficulty levels.

- ***Maze***: $15\text{m} \times 7\text{m}$ in size, start and goal are fixed in $SE(2)$ states $(0, 0, 0)$ and $(11, 0, 0)$, respectively. Obstacles are randomly distributed on the map.
- ***Three stairs***: $9\text{m} \times 9\text{m}$ in size, start and goal are $(0, 0, 0)$ and $(6, 6, \frac{\pi}{2})$, respectively. There are three staircases with different step heights.
- ***Stepping stone***: $9\text{m} \times 3\text{m}$ in size, start and goal are $(0, 0, 0)$ and $(5, y, 0)$ and $y \in [-1.3\text{m}, 1.3\text{m}]$ is randomly sampled. As the difficulty increases, the stepping stones will be randomly removed.
- ***Combination terrain***: $9\text{m} \times 9\text{m}$ in size, start and goal are $(0, 0, 0)$ and $(6, 6, \pi)$. Combining the three terrains above, the robot should cross all of them.

We first run the planner for a sufficient duration to ensure the existence of a solution. This study utilizes the OMPL [20] benchmark tests and evaluates three distinct plan times across various terrains. The statistics result is shown in Tab. I, for all difficulty levels and planning times, we run the test for 100 times. For single kind terrain the planner takes at

TABLE I

GLOBAL PLANNER SUCCESS RATE ON DIFFERENT TERRAIN

Terrain	Plan time[s]	Success rate↑			100% success plan time[s]↓
		Easy	Medium	Hard	
Maze	0.4	.58	.22	.11	1.0
	0.6	.94	.75	.54	
	0.8	**.98**	**.93**	**.88**	
Three stairs	0.1	.71	.62	.24	0.5
	0.2	1.0	.98	.81	
	0.5	**1.0**	**1.0**	**1.0**	
St. stones	0.1	.87	.35	.38	1.0
	0.2	.99	.60	.57	
	0.5	**1.0**	**.96**	**.92**	
Combo. terrain	1.0	.89	.94	.30	5.0
	2.0	.99	1.0	.74	
	4.0	**1.0**	**1.0**	**.98**	

TABLE II

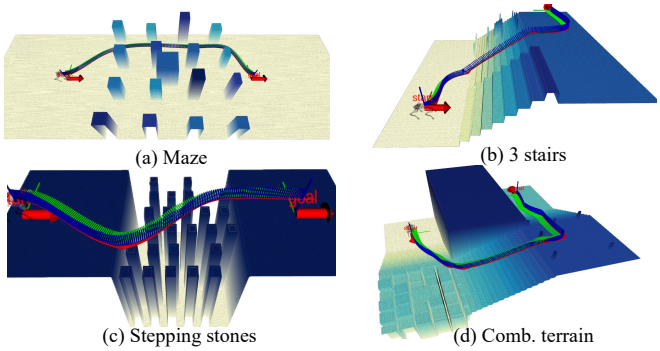
COMPARISON WITH QUAD-SDK

Terrain	Level	Plan time[s]	Success rate↑	
			Quad-SDK	Ours
Maze	Easy	0.4	**1.0**	.58
	Medium	0.6	**1.0**	.75
	Hard	0.8	**1.0**	.88
Three stairs	Easy	0.1	**1.0**	.71
	Medium	1.0	.20	**1.0**
	Hard	1.0	-	**1.0**
St. stones	Easy	0.1	-	**.87**
	Medium	0.2	-	**.60**
	Hard	0.5	-	**.92**
Combo. terrain	Easy	1.0	-	**.89**
	Medium	2.0	-	**1.0**
	Hard	4.0	-	**.98**

'-' means it failed to reach the goal in 5s.

most 1s to reach the goal, and for combination terrain, it spends more time since the path is longer than others.

The terrains and planning results are shown in Fig. 6. In the case of stairs, the path chooses the suitable stair height and avoids high steps. In the case of the stepping stone, the path bypasses the missing stone.



(a) Maze (b) 3 stairs (c) Stepping stones (d) Comb. terrain

Fig. 6. Global path on Different Terrain, start and goal states are marked with red arrows. It shows global planner is adept at circumventing unfavorable terrains, particularly with stairs and stepping stones in (b) and (c). It efficiently plans paths that avoid excessively high stairs and gaps created by missing stones.

2) Comparative Study: We also compare our planner with open-source planner Quad-SDK [3], the result shows in Tab. II, we run all tests for 100 times as well. At *Three stairs* easy level we reduce the time to 0.1s to show Quad-SDK is ultra-fast in flat terrain since it simply uses traversability score as criterion. However, it lost the opportunity to find solutions on steeper stairs and more challenging terrain.

C. MIP Time Performance

This study evaluates the computation time of the MIP planner. The plan times in milliseconds of MIP are reported in Fig. 8 when the plan steps are $n = 3$ and 4. It reads that without the global path guidance, the MIP solver must consider all possible combinations while sometimes unnecessary and cost more time. To demonstrate the planner's capability for real-time operation, the maximum planning time (denoted by '$\times$' in Fig. 8) and the average of the highest 10% of planning times (presented in Tab. III) are calculated. The results demonstrate that our planner consistently achieves planning times not exceeding 500ms, even in the worst-case scenarios. Typically, the slowest 10% of planning times do not surpass 270ms. Given that the gait cycle for activities like trotting lasts about 700ms, these findings suggest the planner's viability for real-time applications.

TABLE III

MIP PLANNING TIME EVALUATION

Terrain	Avg. of the highest 10% of plan time [ms]↓			
	steps $n = 3$		steps $n = 4$	
	w/o path	**w/ path**	w/o path	**w/ path**
Maze	166.8	**32.5**	238	**41.6**
Three stairs	157.2	**59.3**	369.3	**82.1**
St. stones	454.5	**144.0**	1078.8	**268.8**
Comb. terrain	168.7	**63.5**	306.6	**97.4**

D. Close Loop Success Rate in Physical Simulation

In our physical simulation experiments, Raisim [25] serves as the simulator. We perform an ablation study to evaluate our proposed framework under three distinct stepping stones scenarios A, B, and C to measure the closed-loop success rate. Scenario A features the largest stones and the smallest gap is the easiest one. Scenario B, with the smallest stones, aims to rigorously test the foothold planner's precision. Scenario C randomly removes 30% of the stones, thus challenging the robot's global planning capabilities. The study compared the performance of three algorithms: "Global+MIP", which represents our comprehensive proposed framework; "MIP", which simplifies the problem by connecting the start and goal states directly without utilizing a global path; and "Global", which excludes the foothold planner, opting for a heuristic foothold approach as described in [4]. The trot and walk speed is set to about 0.25m/s.

As Tab. IV implies, the success rate is higher for our framework. For Scenario B, in the absence of the MIP planner, the robot encounters failure when its left foothold conflicts with the right, leading to either self-collision or diminished stability. This explains the notably low success rate of the "Global+MPC" approach. Conversely, a MIP planner operating without path guidance occasionally fails to secure a stable solution, leading to failure. For Scenario C, the absence of a global path proves detrimental to the robot, implying that the MIP planner may either fail to find a feasible solution or require significantly large steps to navigate areas with missing stones. Fig. 9 outlines the reasons for their failures.

Additionally, we examine the impact of walk gait and increased base velocities on performance. Both walking and

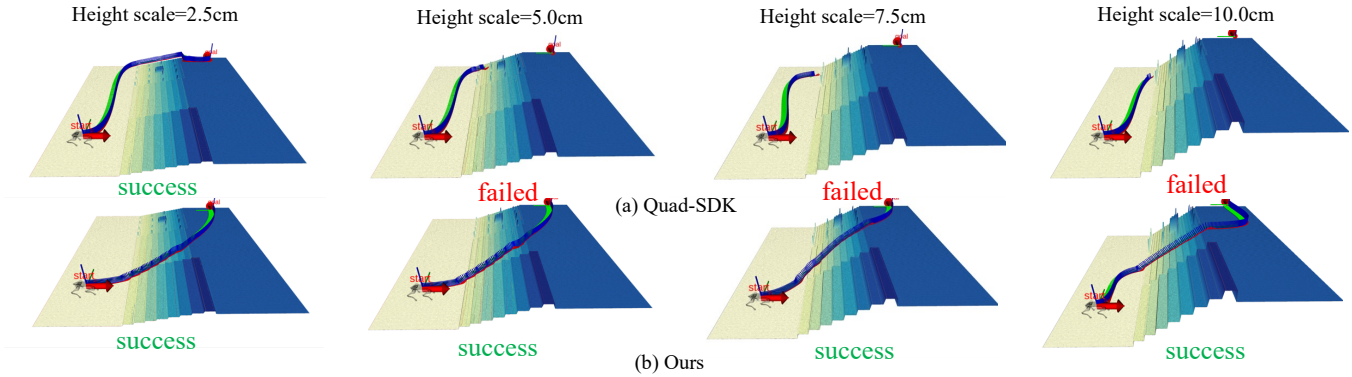


Fig. 7. Comparative Study. With the height scale increase, the Quad-SDK failed to fail the path due to their simple traversability criterion. Our planner can find suitable paths that avoid steps that are too high (Height scale=10cm).

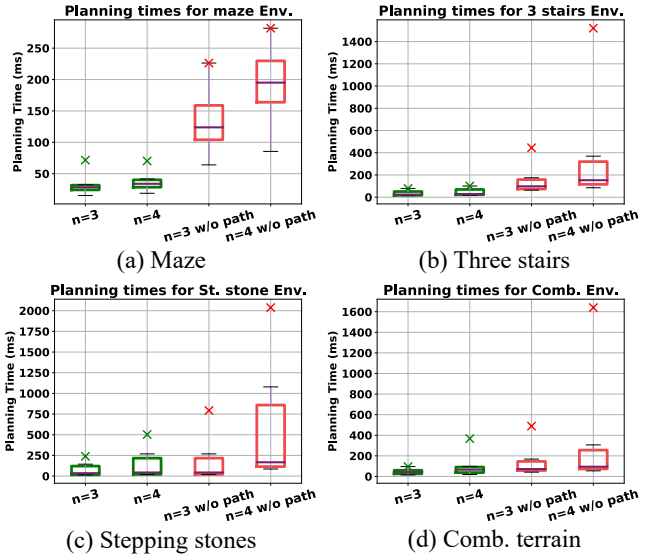


Fig. 8. Box Plot of Planning Time for MIP on Different Terrain. the green boxes show MIP with initial guidance provided by the global path, whereas the red boxes indicate MIP without such guidance. The '×' markers the maximum planning time.

TABLE IV

ABLATION STUDY: CLOSE LOOP PHYSICAL SIMULATION SUCCESS RATE ON STEPPING STONE WITH DIFFERENT ALGORITHM.

Scenario	Algorithm	Success rate↑		
		Trot	Walk	1.5x faster trot
Scenario A (Easy)	**Global+MIP**	**15/15**	**14/15**	**12/15**
	MIP	15/15	11/15	9/15
	Global	15/15	13/15	**12/15**
Scenario B (Challenge the foothold planner)	**Global+MIP**	**14/15**	**12/15**	**10/15**
	MIP	10/15	5/15	2/15
	Global	8/15	8/15	5/15
Scenario C (Challenge the global planner)	**Global+MIP**	**13/15**	**10/15**	**9/15**
	MIP	7/15	3/15	1/15
	Global	12/15	8/15	3/15

moving at faster speeds (1.5x faster trot) are harder at the control level, due to the increasing of swing velocity for each leg. This control complexity lowers the overall success rate.

E. Real-world Experiment

The proposed planning and control framework was tested in the X20 hardware together with the state estimator in our previous work [26]. Since running a real stepping stone is difficult to arrange, we use virtual terrain and check whether the footholds are inside the stone. we tested the full planning stack on each stepping stones terrain three times without failures. Fig. 10 (c) shows the screenshots of our real-world experiment, the X20 robot crossing the virtual stepping stones without missing a step.

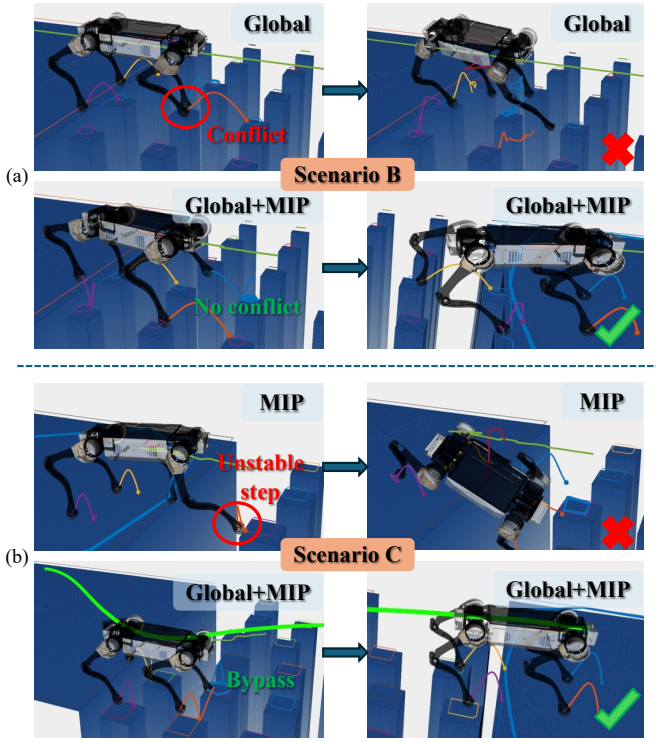


Fig. 9. Case Analysis. (a) In Scenario B, the absence of a MIP foothold planner leads to a collision between the left and right legs, failing. (b) Attempting to have the robot directly cross a large missing stone area without a global path, rather than bypassing it, also fails.

VI. CONCLUSIONS

In this paper, we introduce a fast motion planning and control framework designed for legged robots. Our proposed global planner enhances navigation capabilities across discrete and challenging terrains. With guidance from the global path, the MIP-based planner can optimize footholds across various gaits. Subsequently, the MPC and WBC systems refine and execute whole-body trajectories. Experimental validation of our planning and control system was achieved through both physical simulation and hardware implementation. Throughout these experiments, we observed that the control performance of robots significantly influences their

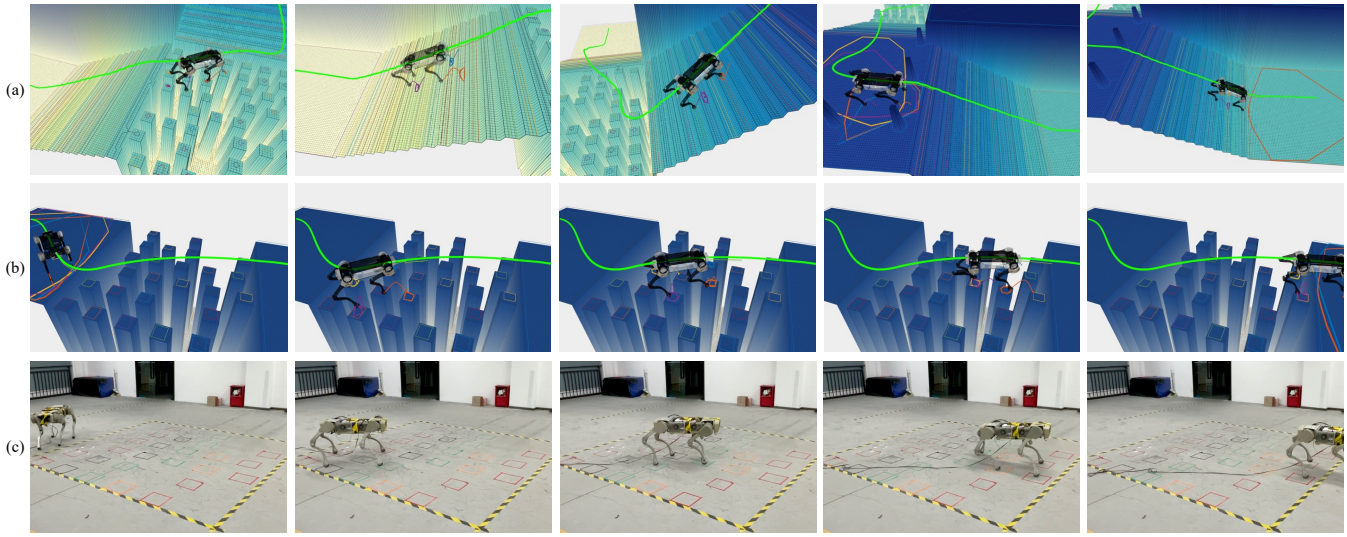


Fig. 10. Screenshots of Different Terrain Experiments. (a) Legged robot X20 navigating through the staircase, stepping stones, and obstacle terrain in physical simulation. (b)(c) are real-world robot crossing stepping stones in real-world experiments, (b) are planner visualization in Rviz, and (c) are real-world screenshots. The X20 robot crossed the stepping stones with 30% stones removed (Scenario C in physical simulation) without missing a step. All step-able areas are marked with colored rectangles.

ability to successfully navigate terrain, highlighting the need for thorough investigation in future works.

REFERENCES

[1] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 586–601, 2018.

[2] E. Jelavic, K. Qu, F. Farshidian, and M. Hutter, "Lstp: Long short-term motion planning for legged and legged-wheeled systems," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4190–4210, 2023.

[3] J. Norby, Y. Yang, A. Tajbakhsh, J. Ren, J. K. Yim, A. Stutt, Q. Yu, N. Flowers, and A. M. Johnson, "Quad-sdk: Full stack software framework for agile quadrupedal locomotion," in *ICRA Workshop on Legged Robots*, 2022.

[4] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model predictive control," *arXiv preprint arXiv:2208.08373*, 2022.

[5] F. Jenelten, R. Grandia, F. Farshidian, and M. Hutter, "Tamols: Terrain-aware motion optimization for legged systems," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3395–3413, 2022.

[6] S. Tonneau, D. Song, P. Fernbach, N. Mansard, M. Taïx, and A. Del Prete, "Sl1m: Sparse l1-norm minimization for contact planning on uneven terrain," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 6604–6610.

[7] T. Corbères, C. Mastalli, W. Merkt, I. Havoutis, M. Fallon, N. Mansard, T. Flayols, S. Vijayakumar, and S. Tonneau, "Perceptive locomotion through whole-body mpc and optimal region selection," *arXiv preprint arXiv:2305.08926*, 2023.

[8] J. Yu, Z. Li, Y. Chen, D. Wang, R. Zhang, Z. Zhang, R. Xiong, and Y. Wang, "Nonlinear mpc-based control framework for quadruped robots: Touch-down in complex terrain," in *2023 42nd Chinese Control Conference (CCC)*, 2023, pp. 4375–4381.

[9] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[10] A. Hereid and A. D. Ames, "Frost: Fast robot optimization and simulation toolkit," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 719–726.

[11] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1184–1189.

[12] J. Norby and A. M. Johnson, "Fast global motion planning for dynamic legged robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3829–3836.

[13] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter, "Perceptive locomotion in rough terrain–online foothold optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5370–5376, 2020.

[14] P. Fankhauser and M. Hutter, "A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation," in *Robot Operating System (ROS) – The Complete Reference (Volume 1)*, A. Koubaa, Ed. Springer, 2016, ch. 5. [Online]. Available: http://www.springer.com/de/book/9783319260525

[15] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Motion planning for quadrupedal locomotion: Coupled planning, terrain mapping, and whole-body control," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1635–1648, 2020.

[16] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019. [Online]. Available: https://drake.mit.edu

[17] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.

[18] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "mpc for torque-controlled legged robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4730–4737.

[19] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: https://www.ros.org

[20] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, Dec. 2012.

[21] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: https://www.gurobi.com

[22] F. Farshidian *et al.*, "OCS2: An open source library for optimal control of switched systems," [Online]. Available: https://github.com/leggedrobotics/ocs2.

[23] J. Carpentier, N. Mansard, F. Valenza, J. Mirabel, G. Saurel, and R. Budhiraja, "Pinocchio - Efficient and versatile Rigid Body Dynamics algorithms," Jul. 2021. [Online]. Available: https://github.com/stack-of-tasks/pinocchio

[24] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[25] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018. [Online]. Available: www.raisim.com

[26] Z. Zhang, Z. Li, Y. Chen, D. Wang, J. Yu, R. Xiong, and Y. Wang, "Invariant ekf based state estimator for quadruped robots," in *2023 42nd Chinese Control Conference (CCC)*, 2023, pp. 4311–4317.