

Traversability-Aware Adaptive Optimization for Path Planning and Control in Mountainous Terrain

Se-Wook Yoo , Graduate Student Member, IEEE, E-In Son , Graduate Student Member, IEEE,
and Seung-Woo Seo , Member, IEEE

Abstract—Autonomous navigation in extreme mountainous terrains poses challenges due to the presence of mobility-stressing elements and undulating surfaces, making it particularly difficult compared to conventional off-road driving scenarios. In such environments, estimating traversability solely based on exteroceptive sensors often leads to the inability to reach the goal due to a high prevalence of non-traversable areas. In this letter, we consider traversability as a relative value that integrates the robot's internal state, such as speed and torque to exhibit resilient behavior to reach its goal successfully. We separate traversability into apparent traversability and relative traversability, then incorporate these distinctions in the optimization process of sampling-based planning and motion predictive control. Our method enables the robots to execute the desired behaviors more accurately while avoiding hazardous regions and getting stuck. Experiments conducted on simulation with 27 diverse types of mountainous terrain and real-world demonstrate the robustness of the proposed framework, with increasingly better performance observed in more complex environments.

Index Terms—Field robots, integrated planning and learning, robust/adaptive control.

I. INTRODUCTION

AS RESEARCH on the autonomous navigation of ground robots in off-road environments is actively progressing, its utilization is gradually increasing in the fields of defense, agriculture, and industry [1], [2], [3]. Compared with urban environments, off-road environments have various obstacles that hinder movements, such as bushes and boulders, and the geometry is complicated due to the unstructured and uneven terrain [4]. In particular, mountainous terrain poses a greater challenge due to its extreme variance in geometrical features such as steep slopes, high elevations, and rugged terrain [5].

Early study [6] applies binary classification to terrain in order to classify it as either traversable or non-traversable for navigation on uneven terrain. Ref. [7] broadens the scope of the terrain

Manuscript received 2 February 2024; accepted 23 March 2024. Date of publication 10 April 2024; date of current version 19 April 2024. This letter was recommended for publication by Associate Editor J. Wang and Editor H. Kurniawati upon evaluation of the reviewers' comments. This work was supported by the BK21 FOUR Program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2024. (Se-Wook Yoo and E-In Son contributed equally to this work.) (Corresponding author: Seung-Woo Seo.)

The authors are with the Department of Electrical and Computer Engineering and ASRI, Seoul National University, Seoul 151-747, Republic of Korea (e-mail: tpdnr1360@snu.ac.kr; pingpang@snu.ac.kr; sseo@snu.ac.kr).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2024.3387042>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3387042

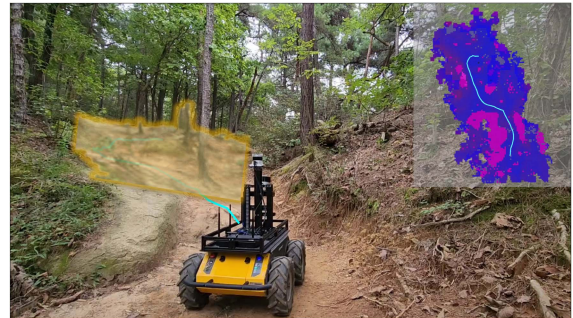


Fig. 1. Robot navigation in mountainous terrain. We propose a traversability-aware navigation framework in both virtual and real environments. Virtual environments are reconstructed from actual mountainous terrain. The top right of the figure displays the optimal path (cyan) calculated using traversability information.

classification from binary to multi-class and assigns a fixed speed profile for each class. However, discrete categorization is insufficient to naturally capture the terrain feature. To overcome this problem, [8] maps the terrain properties produced by perceptual data to a scalar cost value, so-called traversability. Such approaches consider designed traversability with hand-crafted metrics such as slope, slippage, and flatness in the optimization process. Ref. [9] uses exteroceptive sensor-based traversability as the coefficient of the control cost to slow down the robot in dangerous areas. However, in mountainous terrain where the overall traversability cost is high, it leads to frequent instances of the robot getting stuck due to the reduced velocity. To address this issue, it is necessary to consider the traversability cost in a relative perspective according to the robot's motion, rather than solely relying on geometric representation. For instance, a low velocity may prevent the robot from moving through actually traversable areas, while a high velocity may allow the robot to move through untraversable areas. Therefore, we incorporate proprioceptive sensor-based traversability, which takes into account the inertial state of the robot, as the tracking coefficient to adjust the robot's velocity dynamically. In addition, to prevent our method from overlooking the potential risk in hazardous terrain, we adopt constraints in optimization to limit the robot's behavior.

In this letter, we present **TAO**, Traversability-aware Adaptive Optimization in mountainous terrain (see Fig. 1) to assist in flexible decision-making processes for robot navigation. We design traversability into two components: apparent

traversability using exteroceptive features and relative traversability using proprioceptive features, which consider the ground-robot interaction through the robot's posture variability. Subsequently, we integrate the traversability into the optimization process of sampling-based planning and motion predictive control. Apparent traversability is used as a sampling bias for planning to generate a feasible path. This effectively reduces the search space while generating a feasible path that circumvents undulating regions, eliminating the high potential risk. Relative traversability is applied to regulate the robot's velocity to execute the desired behaviors more accurately while avoiding vigorous posture change, getting stuck, or being damaged.

The main contributions can be summarized as follows:

- We introduce a novel adaptive navigation method that takes into account the relative perspective of traversability based on the robot's inertial state.
- We achieve an improvement of up to 16% in both effectiveness and stability through hazard avoidance and dynamic velocity adjustments that take into account ground-robot interaction.
- We demonstrate the feasibility of our method to various types of off-road terrain through experiments conducted in real-world and simulations involving 27 diverse terrains.

The remainder of this letter is organized as follows. A comparative review of related works is presented in Section II. We state the problem definition in Section III. The proposed framework is explained in detail in Section IV. Experimental results are demonstrated in Section V. The conclusion follows in Section VI.

II. RELATED WORK

In this section, we discuss existing work on modeling terrain traversability and acquiring resilient skills for robot navigation in off-road environments.

A. Modelling Terrain Traversability

The utilization of exteroceptive and proprioceptive sensory data is widely studied for modeling terrain traversability. Using exteroceptive sensors consists of appearance-based and geometry-based approaches [1]. Appearance-based approaches devise terrain traversability by classifying terrain types according to a category (grass, road, tree, etc.) from visual measurements [10]. Geometry-based approaches compute the traversability by considering geometrical representations such as slope or roughness from depth measurements [11]. Unlike using exteroceptive sensors, which ignore the robot's inertial states, proprioceptive approaches model traversability using measurements that capture the robot's inertial state, such as ground reaction score [12], vibration [13], or energy consumption [14].

B. Traversability-Aware Navigation

Applying a classic path planner [15], [16], [17] to an uneven off-road terrain poses a challenge since the robot operates in

a dynamic environment while avoiding mobility-stressing elements. Initial work assigns a fixed speed profile according to the roughness [7] or semantic units [18] of the terrain surface. Ref. [19] estimates the cost map through terrain traversability assessment (TTA) using features on digital elevation map (DEM) or cumulative fractional area (CFA) and utilizes it as a guide of course direction. Specifically, TTA can be added to the distance cost when generating a search graph [20], or adjusting an existing route [21]. Recently, learning-based methods have been suggested to predict the cost map using data labeled with reachability [22], [23], [24]. In [25], the predicted cost map is utilized as adaptive sampling distribution to create a roadmap. Meanwhile, in [26], the cost map is learned using ground interaction data collected by the agent in the reinforcement learning process and used as a bias for course direction.

A large amount of training data is required to generalize a traversable map using a learning-based method. In particular, ground interaction data is scarce or difficult to obtain in off-road terrain, and the labeling cost is expensive. Furthermore, domain adaptation is required to apply the robot to the real-world environment even if the model is trained with simulation data. On the other hand, [9] applies rule-based TTA to the cost of the RRT node for path optimization and to find reactive parameters for MPC optimization. However, as inconsistency occurs between desired and actual behavior in tracking the planned path, the cumulative error gradually increases, resulting in a stuck, wheel slip, or tip-over.

Since the mountainous environment is difficult to define all types in advance, our method computes geometry-based TTA using depth measurement without classifying the terrain type. We also utilize the approach from [9] to seamlessly integrate the geometry-based TTA into the path planning and control process. However, we extend this approach by incorporating a traversability distribution with additional consideration of the geometry-based constraint map to apply an adaptive sampling scheme, which is inspired from [25]. Furthermore, we propose reactive parameters for optimizing control variables by using the ground reaction score, which considers the change in pitching angle that occurs when the robot interacts with the undulating terrain.

III. PROBLEM DEFINITION

Our goal is to develop a framework that enables a robot to navigate through mountainous terrain while ensuring safe and efficient traversal. We define a search space $X \subset \mathbb{R}^3$ to achieve this goal. Let us denote the location in search space as $\mathbf{x} = (x, y, z) \in X$, the space occupied by non-ground obstacles as $X_{obs} \subset X$, the area of the ground surface as $X_{surf} \subset X$, and the area of traversable space as $X_\tau = X_{surf} \cap X \setminus X_{obs}$. To distinguish traversable area in X_τ , we define apparent traversability as \mathcal{M}_τ and regional constraints as \mathcal{M}_γ , respectively, which are expressed by occupancy grid map \mathcal{M} . To calculate \mathcal{M}_τ , we utilize height map \mathcal{M}_z and normal vector map \mathcal{M}_n . We also define relative traversability Ψ_τ to consider ground-robot interaction. Ψ_τ represents the fluctuation of the pitching angle that occurs when a robot traverses uneven terrain.

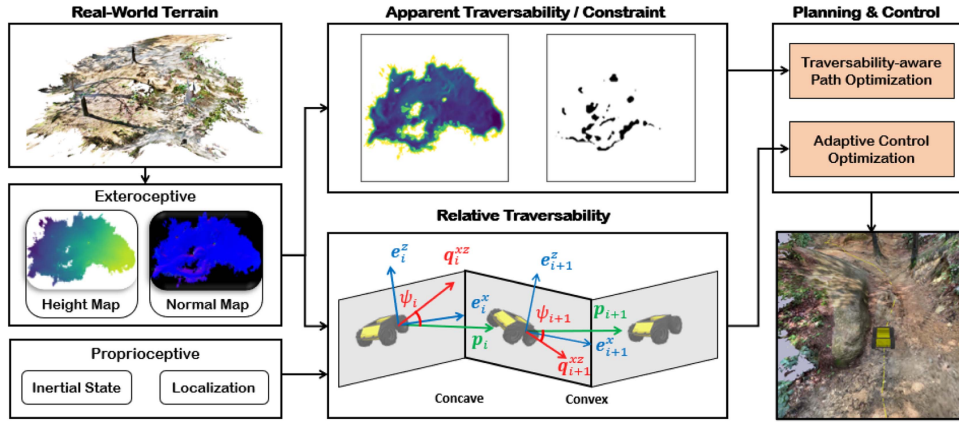


Fig. 2. Overall structure. Our framework takes inputs such as a height/normal map, localization, and the robot's inertial states (orientation \mathbf{q} , linear velocity ζ , and angular velocity ω). The computed traversability/constraint map of the terrain is used as guidance to generate a trajectory. The controller adjusts the robot's velocity with adaptive optimization based on apparent and relative traversability. This enables the robot to navigate extreme mountainous terrains robustly. The top left of the figure shows a sample of terrains collected on campus and mountains.

The formal definition of our problem is as follows: path planner searches path $\mathcal{P} \subset X_\tau$ from \mathbf{x}_{start} to \mathbf{x}_{goal} . The optimal path $\mathcal{P}^* \subset \mathcal{P}$ is obtained by minimizing \mathcal{M}_τ such that \mathcal{M}_τ while satisfying to avoid collision with obstacles. The path tracker searches for a feasible control strategy π^* that minimizes the difference between the desired state and predicted state for the planning time horizon. We also regularize optimization parameters with \mathcal{M}_τ and Ψ_τ to follow \mathcal{P}^* reactively considering the interaction of the terrain and the robot.

IV. METHODOLOGY

We present TAO, and Fig. 2 shows the overall structure. We compute apparent traversability to reflect the geometric features of the surrounding terrain using depth measurements from exteroceptive sensors. To consider ground-robot interaction, we calculate relative traversability using measurements from exteroceptive and proprioceptive sensors. Then, we use apparent traversability as sampling bias and the cost for generating an optimal trajectory. Relative traversability is utilized to dynamically adjust the robot's velocity, ensuring the tracking controller's ability to follow the given path.

A. Terrain Assessment for Non-Planar Ground

We consider the geometric features of the non-planar ground to traverse extreme terrain. The geometric features comprise a height map \mathcal{M}_z and a normal map \mathcal{M}_n . To get \mathcal{M}_z and \mathcal{M}_n , an observed plane is represented as N sets of points included in a square kernel with a side length K centered on x_i . A set of N kernels is represented as $\mathcal{K}(\mathbf{x}) = \{\mathcal{K}(\mathbf{x}_i) \mid \text{for } i = 1, \dots, N\}$. We utilize the SVD algorithm to fit a surface plane $\mathcal{K}(\bar{\mathbf{x}})$ with the set of points $\mathcal{K}(\mathbf{x})$. Here, we denote each point on the fitted plane as $\bar{\mathbf{x}}_i = (x_i, y_i, z_i)$ and the corresponding unit normal vector as $\mathbf{n}_i = (n_i^x, n_i^y, n_i^z)$. To obtain an occupancy grid representation, we map each point in the kernel to a corresponding point on \mathcal{M} with a size of $\mathbb{R}^{H \times W}$.

1) *Apparent Traversability and Regional Constraints From Exteroceptive Features:* We utilize the apparent traversability

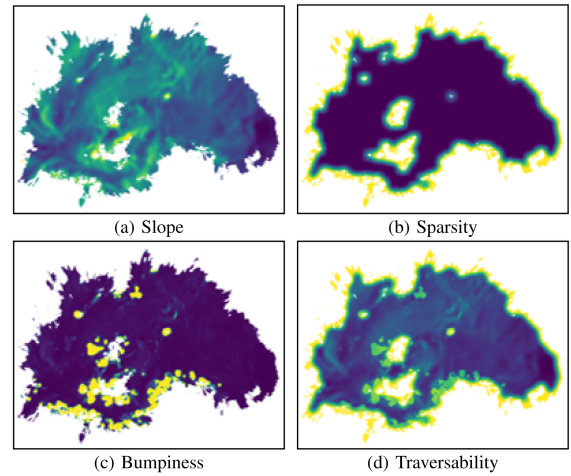


Fig. 3. Visualization of exteroceptive features. The color of each map is represented by a range from deep blue with a score of zero to vibrant yellow with a score of one. (d) is obtained as a linear combination of the components (a)–(c).

\mathcal{M}_τ as a criterion to select an optimal path with minimal slope and fluctuation. Inspired from [9], the traversability $\tau_i \in [0, 1]$ for i -th kernel is evaluated based on the linear combination of the following exteroceptive features: slope, sparsity, and bumpiness (see Fig. 3)

$$\tau_i = \alpha_1 \tilde{\eta}_{slop} + \alpha_2 \tilde{\eta}_{spar} + \alpha_3 \tilde{\eta}_{bump}, \quad (1)$$

where the weights α_1 , α_2 , and α_3 sum to 1. $\tau_i = 0$ indicates that the ground in i -th kernel is fully traversable, while $\tau_i = 1$ indicates that the ground is non-traversable. $\tilde{\eta}$ represents a normalized value for each feature. Each feature is calculated as follows:

$$\eta_{slop} = \arccos(\mathbf{n}_i^\top \cdot \mathbf{k}) \quad (2)$$

$$\eta_{spar} = \begin{cases} 1 & r_i > r_{max} \\ \frac{r_i - r_{min}}{r_{max} - r_{min}} & r_i \in [r_{min}, r_{max}] \\ 0 & r_i < r_{min} \end{cases} \quad (3)$$

$$\eta_{bump} = \inf_{\gamma \in \Pi(P_{\mathcal{K}_i}, Q_{\bar{\mathcal{K}}})} \mathbb{E}_{x,y \sim \gamma} [\|x - y\|], \quad (4)$$

where \mathbf{k} is a unit vector in 3-dimensional Euclidean space that is perpendicular to the xy plane. Vacancy ratio r_i represents the proportion of vacant parts of $\mathcal{K}(\mathbf{x}_i)$. r_i is determined by r_{min} and r_{max} , representing the minimum and maximum values, respectively. If there are clusters of vacant points, it suggests a high possibility of pits and scarps on the ground. Unlike [9], we also incorporate the bumpiness feature, as the slope alone may not fully account for the risk posed by significant terrain fluctuations. We use the earth mover's distance (EMD) [27] to compute the bumpiness of the ground. EMD is obtained by comparing the distribution of $\mathcal{K}(\mathbf{x}_i)$ with the distribution of $\mathcal{K}(\bar{\mathbf{x}})$. Each distribution is denoted as $P_{\mathcal{K}_i}$ and $Q_{\bar{\mathcal{K}}}$, respectively. The set $\Pi(P_{\mathcal{K}_i}, Q_{\bar{\mathcal{K}}})$ represents all joint distributions that have $P_{\mathcal{K}_i}$ and $Q_{\bar{\mathcal{K}}}$ as their respective marginal distributions. Furthermore, we utilize the regional constraints \mathcal{M}_γ to avoid hazardous terrain on non-planar ground. Based on designed features in (2)–(4), the constraints $\gamma_i \in \{0, 1\}$ for i -kernel is evaluated as follows:

$$\gamma_i = ((\tilde{\eta}_{stop} > \epsilon_{stop}) \vee (\tilde{\eta}_{bump} > \epsilon_{bump})) \wedge (\tilde{\eta}_{spar} < \epsilon_{spar}), \quad (5)$$

where ϵ is the threshold value for each feature. $\gamma_i = 1$ indicates that the perceived point on the ground is hazardous.

2) *Relative Traversability From Exteroceptive and Proprioceptive Features*: To robustly follow a given path in undulating terrain, it is essential to consider ground-robot interaction in areas around the robot. Addressing this, we propose relative traversability Ψ_τ that considers the relations among the robot's orientation \mathbf{q} , the ground normal vector \mathbf{n} , and the planned path direction \mathbf{p} . Here, we denote the relative traversability as $\psi_i \in \Psi_\tau$, where i is the index corresponding to the kernel that the robot is located in k -th step of the planned path. As shown in Fig. 2, ψ_i is calculated as follows:

$$\mathbf{e}_i^z = \mathbf{n}_i \quad (6)$$

$$\mathbf{e}_i^x = \frac{\mathbf{p}_i - k_i \mathbf{e}_i^z}{\|\mathbf{p}_i - k_i \mathbf{e}_i^z\|^2} \quad (7)$$

$$\mathbf{e}_i^y = \mathbf{e}_i^x \times \mathbf{e}_i^z \quad (8)$$

$$\mathbf{q}_i^{xz} = \text{Proj}_{\mathbf{e}_i^y}(\mathbf{q}_i) \quad (9)$$

$$\psi_i = \arcsin((\mathbf{q}_i^{xz} \times \mathbf{p}_i)^\top \cdot \mathbf{e}_i^y), \quad (10)$$

where $\mathbf{p}_i = p_{i+1} - p_i$ is directional vector towards the next planned point and $k_i = \mathbf{p}_i^\top \cdot \mathbf{e}_i^z$ is the component of the vector \mathbf{p}_i in the direction of \mathbf{e}_i^z . Note that planned points are computed using (13), which considers the robot's linear and angular velocity.

B. Planning and Control

1) *Traversability-Aware Path Optimization*: We propose an adaptive sampling scheme and path optimization method that can be integrated into a sampling-based search tree method for path planning. This is compatible with various RRT variants and can improve the solution quality. In general, each element of the search tree \mathcal{T} is represented as a node \mathcal{N}_i instead of a location \mathbf{x}_i . We expand this representation as a tuple $\mathcal{N}_i = (\mathbf{x}_i, \tau_i, \gamma_i)$.

Algorithm 1: RejectionSampling($B, \mathcal{M}_n, \mathcal{M}_z$).

- 1: $\mathcal{N}_i \leftarrow \text{UniformSampling}(B)$
 - 2: Calculate τ_i from Eq. (1) using \mathcal{M}_n and \mathcal{M}_z
 - 3: Calculate γ_i from Eq. (5) using \mathcal{M}_n and \mathcal{M}_z
 - 4: Calculate ρ_i from Eq. (11) using τ_i and γ_i
 - 5: Sample $\nu \sim U(0,1)$
 - 6: **if** $\nu < \rho_i$ **then**
 return \mathcal{N}_i
 - 7: **else**
 return RejectionSampling($B, \mathcal{M}_n, \mathcal{M}_z$)
-

The added τ_i and γ_i are utilized as criteria for tree expansion and path selection. Based on this, we generate a path that circumvents terrains difficult for traversal, instead of just seeking the shortest path. To achieve this, we define a sampling weight ρ_i that can provide guidance for sampling new nodes for tree expansion as follows:

$$\rho_i = (1 - \tau_i)(1 - \gamma_i) \quad (11)$$

The rejection sampling algorithm is presented in Algorithm 1, where ρ_i is the target distribution. The input $B = [x_{min}, x_{max}] \times [y_{min}, y_{max}]$ represents the boundary of the area perceived by the robot. In line 1, we use B to generate proposal samples using the UniformSampling function. In line 5, $U(0,1)$ represents uniform distribution over the range $[0,1]$. Even though new nodes are not directly placed in the constrained region, the tree expansion process may create new edges that cross this area. To avoid this, we connect N_i and N_j with a path segment $\mathbf{x}_{i:j} = \{\mathbf{x}_i, \dots, \mathbf{x}_k, \dots, \mathbf{x}_j\}$ of length l_{ij} composed of infinitesimal lengths ΔL . Then, ρ_k is evaluated at a point $\mathbf{x}_k \in \mathbf{x}_{i:j}$ using (11). Based on the notation above, we design the cost function g over an edge $\mathcal{E}_{ij} = (\mathcal{N}_i, \mathcal{N}_j)$ that connects two nodes as follows:

$$g(\mathcal{E}_{ij}) = \left(1 + \kappa \left(\frac{1}{\sum_{k=i}^j \rho_k} - |\mathbf{x}_{i:j}| \right) \right) l_{ij}, \quad (12)$$

where κ is a penalty weight for the proposed criterion, and $|\mathbf{x}_{i:j}|$ is denoted as the number of points in the path segment. In addition, we use a Reed-Shepp [28] curve to generate path segments to reduce the high degree of control freedom in mountainous terrain.

2) *Adaptive Control Optimization*: We formulate a nonlinear model predictive controller (NMPC) that prevents getting stuck or damaged when following the feasible path generated by the planner at every moment. We use a skid-steering model [29] f to predict the robot's state $\mathbf{s} = (x, y, \theta)$, which is as follows:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \cos \theta_t & 0 \\ \sin \theta_t & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_t \Delta t, \quad (13)$$

where $\mathbf{u}_t = (\zeta_t, \omega_t)$ is a control variable, denoting linear velocity and angular velocity, respectively, θ_t is heading angle and $\Delta t > 0$ is time interval. We utilize both apparent and relative

traversability to optimize control parameters, as outlined below:

$$\min_{\mathbf{s}, \mathbf{u}} \sum_{k=t}^{t+T-1} \{ \|\mathbf{s}_k - \mathbf{s}_k^d\|_Q^2 + \|\mathbf{u}_k\|_{R_k}^2 + \|\mathbf{v}_k - \mathbf{v}_k^d\|_{W_k}^2 \} + \lambda C_\tau(\mathbf{x}_{k:T}), \text{ s.t. } \mathbf{s}_{k+1} = f(\mathbf{s}_k, \mathbf{u}_k) \quad (14)$$

$$R_k = \left(1 - \frac{1}{|\mathcal{K}(\mathbf{x}_k)|} \sum_{\mathbf{x}_j \in \mathcal{K}(\mathbf{x}_k)} \tau_j \right)^{-2} R \quad (15)$$

$$W_k = (1 + k_q \max(0, \tan \psi_k)) W^v, \quad (16)$$

where \mathbf{s}^d is desired state, \mathbf{v}^d is velocity on \mathcal{P}^* , and T represents prediction horizon. $Q \in \mathbb{R}^{3 \times 3}$ and $R \in \mathbb{R}^{2 \times 2}$ are positive definite matrices, where $Q = \text{diag}(Q^x, Q^y, Q^\phi)$ and $R = \text{diag}(R^c, R^\omega)$. In (14), the first term minimizes the error between the predicted state and the desired state. The second term slows down the robot by adjusting the control variable magnitude based on traversability, as shown (15). The third term dynamically adjusts the robot's velocity by regulating W_k through (16), which enables the robot to maneuver bumps or ditches. The last term considers the open space based on apparent traversability around predicted states. We sample points uniformly around the predicted states, then compute C_τ by averaging traversability for these random points, where λ is a constant.

V. EXPERIMENTS

We designed our experiments to answer three questions:

- Q1. How does our planning method compare to prior methods for generating a feasible path in extreme mountainous terrains?
- Q2. Can our navigation method be generalized to rough and undulating terrains?
- Q3. What are the benefits achieved through coupling resilient behavior to existing control methods?

A. Simulation Settings

We set a virtual environment with a mesh model of the real-world terrain on the PyBullet physics engine [30]. The real-world terrain data is acquired as a point cloud via Clearpath Husky UGV mounted with LiDAR and transformed into the mesh. We collected 27 types of various terrains on campus and mountains, which have an average size of $22 \text{ m} \times 16 \text{ m}$. To demonstrate the robustness of TAO on rough and undulating terrains, we augment the terrain from 27 to 108 types by introducing four variations in the simulation. The variations are produced by adjusting the overall inclination in 5° increments between 0° and 15° . For the evaluation, we sort terrains based on the traversability score into levels of easy, plain, and hard. A higher level indicates rougher terrain with steeper inclinations. Table III describes the hyperparameters used for our experiments.

B. Baselines and Evaluation Metrics

We compare the effectiveness of our planning method (TAO-Plan) with three variations of the sampling-based planning method: DEM [19], PUTN [9], and \mathcal{M}_τ -Only. To ensure fairness in our comparison, all methods uniformly sample nodes for

tree expansion. For the DEM, the cost is calculated based on a digital elevation map. In the case of PUTN, the methodology for evaluating terrain remains intact. Unlike \mathcal{M}_τ -Only, PUTN utilizes the flatness feature instead of the bumpiness in (4). The \mathcal{M}_τ -Only does not consider \mathcal{M}_γ to calculate the cost, which serves as an ablation study for TAO-Plan. Moreover, we compare two control methods: Vanilla and TAO-Ctr. For Vanilla, we adopt MPC [31] and MPPI [32], which are derivative-based and sampling-based approaches, respectively. For TAO-Ctr, we extend Vanilla method with adaptive control optimization, which demonstrates the ease of coupling our method with other controllers. Vanilla does not consider ground interaction, whereas TAO-Ctr considers the interaction by using the relative traversability Ψ_τ . In the following contents, we denote TAO-Ctr as Adaptive for the ease of notion. In the case of PUTN, we employ the original control strategy from PUTN, referred to as PUTN-Ctr, which reduces velocity in regions with low traversability. Note that this approach solely relies on apparent traversability and does not incorporate relative traversability. We combine the aforementioned methods and evaluate the navigation performance. All evaluations are conducted with 30 trials for each terrain using the following metrics:

- Average Elevation Gradient (AEG) - The average of the elevation gradients experienced by the robot along the trajectory. The elevation gradient is calculated by the difference of the robot's elevation at any time instant.
- Traversability - The summation of traversability of the nodes located along a planned trajectory. Note that the higher traversability value indicates that the ground is non-traversable.
- Cross Tracking Error (CTE) - The average distance error between the robot's position and the trajectory.
- Inconsistency - The average distance error between the desired position and the actual position after executing a planned behavior.
- Success - The rate at which the robot has successfully reached the goal while avoiding collisions or getting stuck. We assume that the robot is stuck when the difference in position and heading is insignificant for 30 seconds.
- Step - The average number of steps the robot takes to reach the goal within a single episode. The maximum step is set to 3000.
- Progress - The ratio of the distance the robot actually traveled to the distance of the planned path.

C. Navigating Extreme Mountainous Terrains

We show quantitative results in Table I. TAO (TAO-Plan + TAO-Ctr) outperforms all the other methods by achieving successful navigation with the fewest number of steps. As shown in Progress, even in failure cases, TAO can travel the farthest distance before a collision or getting stuck. This is particularly evident at the hard level (Q2), where TAO outperforms the second-best baseline in success rate by a margin of +16% (66% to 82%). As expected, the DEM performs well at the easy level but its performance sharply declines as the difficulty level increases. This demonstrates that height information alone is

TABLE I
COMPARISON FOR NAVIGATION METHODS

LEVEL	METRICS	DEM				PUTN				\mathcal{M}_τ -ONLY				TAO-PLAN			
		VANILLA		TAO-CTR		PUTN-CTR		TAO-CTR		VANILLA		TAO-CTR		VANILLA		TAO-CTR	
		MPC	MPPI	MPC	MPPI	MPC	MPPI	MPC	MPPI	MPC	MPPI	MPC	MPPI	MPC	MPPI	MPC	MPPI
EASY	SUCCESS (%)	52.22	49.51	52.84	47.53	91.33	95.60	91.60	95.87	95.80	92.10	95.56	94.32	96.67	97.04	98.15	<u>97.78</u>
	STEP (k)	1.73	1.84	1.70	1.84	0.68	0.66	0.71	0.65	0.66	0.70	0.66	0.70	0.65	0.74	0.60	<u>0.63</u>
	PROGRESS (%)	71.16	71.34	71.25	69.69	93.59	97.03	94.21	97.21	96.96	97.56	96.75	97.56	98.06	98.34	98.76	<u>98.68</u>
PLAIN	SUCCESS (%)	22.96	22.96	19.51	23.83	83.47	89.33	76.80	<u>90.93</u>	85.19	85.43	85.31	90.12	85.68	89.14	86.67	93.83
	STEP (k)	2.47	2.47	2.54	2.42	1.02	1.06	0.99	0.90	1.00	1.08	0.98	<u>0.89</u>	1.01	1.02	0.98	0.82
	PROGRESS (%)	53.65	53.20	50.38	53.93	89.80	93.41	87.27	93.50	92.26	92.55	92.91	93.73	92.20	<u>94.00</u>	93.16	96.79
HARD	SUCCESS (%)	6.79	7.16	6.67	12.96	40.27	55.33	50.00	61.20	38.52	50.86	40.62	62.10	61.11	<u>65.80</u>	61.23	81.73
	STEP (k)	2.84	2.84	2.85	2.69	2.21	1.94	1.89	1.62	2.15	1.96	2.09	<u>1.60</u>	1.68	1.75	1.68	1.27
	PROGRESS (%)	42.04	42.46	41.04	45.68	70.05	79.26	76.54	82.88	69.57	76.31	70.95	82.94	80.38	<u>83.74</u>	81.25	89.50

TAO (TAO-Plan + TAO-Ctr) outperforms other methods in every metric, especially at the hard level. The best results in each category are in bold, with the second best results underlined.

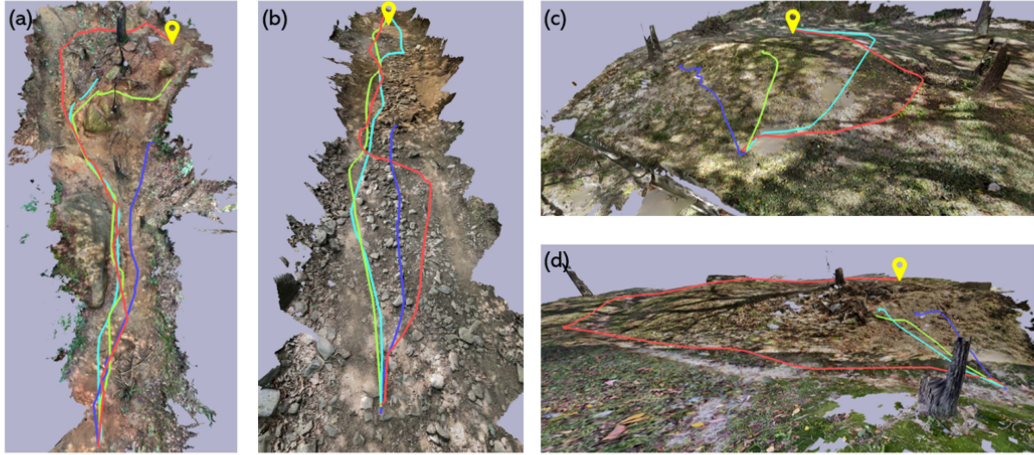


Fig. 4. Navigation in mountainous environments. We compare 4 different path planning methods - DEM (purple), \mathcal{M}_τ -ONLY (green), PUTN (cyan) and TAO-plan (red) - using adaptive MPPI controller. The goal point is indicated by a yellow marker. Our proposed method (TAO) successfully navigates in mountainous terrains by computing trajectories that circumvent bumpy and risky regions and dynamically adjust the robot's velocity.

insufficient for a robot to navigate complex terrain. Both PUTN and \mathcal{M}_τ -Only exhibit comparable performance to TAO in the easy and plain levels. However, in hard terrains, performance decreased as they failed to generate paths that circumvent bumpy and inclined regions. Fig. 4 visualize trajectories of four different path planning methods using an Adaptive MPPI controller. Fig. 4(a) illustrates the example of hard-level terrain with the highest traversability score. The right path includes a narrow and steep slope below the goal point. In order to reach the goal, the robot must follow the left path. DEM fails in reaching the goal due to the presence of a higher sill, which is beyond the robot's ground clearance. PUTN encounters difficulty in identifying the tree roots, which leads to getting stuck and being unable to reach the goal. \mathcal{M}_τ -Only computes a trajectory that circumvents the sill but still takes the right passage. TAO generates the left path and successfully navigates to the goal through resilient path planning and control.

D. Traversability-Aware Path Planning

Fig. 4(b) depicts a steeply slanting environment with scattered small and large rocks, which causes the robot to get stuck while moving across the region. DEM generates a trajectory that passes through the bumpy region, but since it lacks the ability to analyze

intricate terrain characteristics beyond the height map, the robot eventually gets stuck. PUTN tends to stick close to or step over gravel in narrow passages. Although \mathcal{M}_τ -Only avoids the bumpy middle region clearly by considering the bumpiness of terrain features in (4), it generates a trajectory toward a relatively narrow passageway on the left. In contrast, our proposed method chooses the wider right passageway and successfully navigates through the least undulating region to reach the goal (Q1).

In Fig. 4(c) and (d), there are steep hills between the starting and the goal point with rough ground. DEM and \mathcal{M}_τ -Only focus on the shortest distance, they fail to reach the goal due to slippage or roll-over. PUTN can reach the goal in Fig. 4(c), but shows an unstable trajectory passing over a high slope. In Fig. 4(d), similar to prior methods, it fails to generate a trajectory to circumvent the risky regions. On the other hand, our approach designates the rough area as a constraint and computes a trajectory that bypasses it, allowing the robot to successfully reach the goal. The ability of TAO-Plan to generate such a path is attributed to the proposed rejection sampling in (11) and cost function in (12). The path generation process in Fig. 4(d) is depicted in Fig. 5.

Table II shows performance comparison in terms of the path planning method. All methods used an Adaptive MPPI controller. In most cases, the TAO-Plan outperforms other methods,

TABLE II
COMPARISON FOR PATH PLANNING METHODS

LEVEL	AEG (mm)				TRAVERSABILITY (%)				CTE (cm)				INCONSISTENCY (cm)			
	DEM	PUTN	\mathcal{M}_τ -ONLY	TAO-PLAN	DEM	PUTN	\mathcal{M}_τ -ONLY	TAO-PLAN	DEM	PUTN	\mathcal{M}_τ -ONLY	TAO-PLAN	DEM	PUTN	\mathcal{M}_τ -ONLY	TAO-PLAN
EASY	6.53	10.77	4.92	4.79	10.99	4.42	1.21	1.08	16.88	16.24	12.56	12.75	3.46	3.91	2.74	2.74
PLAIN	8.67	12.38	5.32	5.26	13.67	4.82	1.20	1.15	18.14	19.09	12.57	12.45	3.82	3.79	2.76	2.78
HARD	14.72	13.77	8.85	7.94	13.98	5.71	1.78	1.47	23.79	22.64	21.06	21.56	4.35	3.32	3.08	3.06

TAO consistently outperforms other methods in terms of AEG and Traversability. Although the computed trajectories have a larger CTE, there is no degradation in inconsistency.

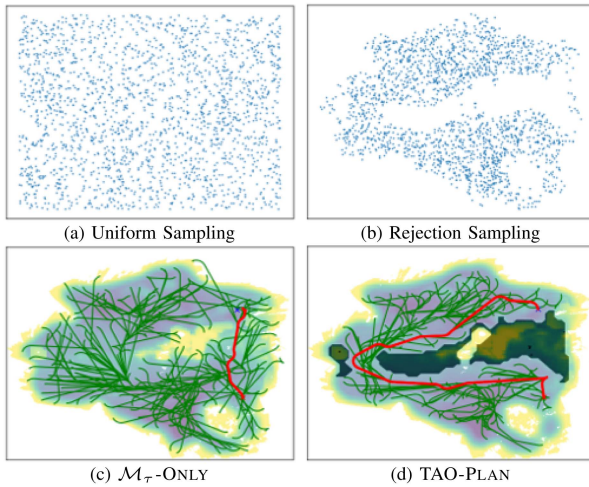


Fig. 5. Path generation process. The difference between (a) and (b) shows the sampling bias. (c) and (d) also visualizes the search tree and optimal path overlaid on the traversability map. The search tree is depicted as green solid lines, while the selected optimal path is shown as red solid lines.

TABLE III
HYPERPARAMETER

HYPERPARAMETER	VALUE
$T, N, H, W, \Delta L, \Delta t$	20, 2000, 500, 500, 0.1, 0.1
$\alpha_1, \alpha_2, \alpha_3, \kappa, k_q$	0.3, 0.3, 0.4, 5, 5
$r_{min}, r_{max}, \epsilon_{slop}, \epsilon_{spar}, \epsilon_{bump}$	0.2, 0.8, 0.7, 0.6, 0.5
$Q^x, Q^y, Q^\phi, R^\zeta, R^\omega, W^v$	10, 10, 1, 0.5, 0.5, 3

especially with traversability and AEG scores. This result implies that apparent traversability induces generating a stable path without large fluctuations. Meanwhile, even if CTE increases, there is no significant difference in inconsistency.

E. Resilient Behavior for Control

We analyze the effect of adaptive control optimization (Section IV-B2). We use the same planned path (TAO-Plan) to compare control strategies fairly. As shown in Table I, applying TAO-Ctr (Adaptive) leads to performance improvement in most methods. In particular, the performance is significantly improved at the hard level due to the strategy of regulating the velocity considering relative traversability, slowing down in dangerous areas, and accelerating in areas where the robot is likely to get stuck (Q3). Also, TAO-Ctr shows better performance compared to PUTN-Ctr in hard terrains within the PUTN planner, highlighting that the approach of regulating velocity based on the robot's inertial state is more effective than simply decreasing velocity based on solely geometric representation.

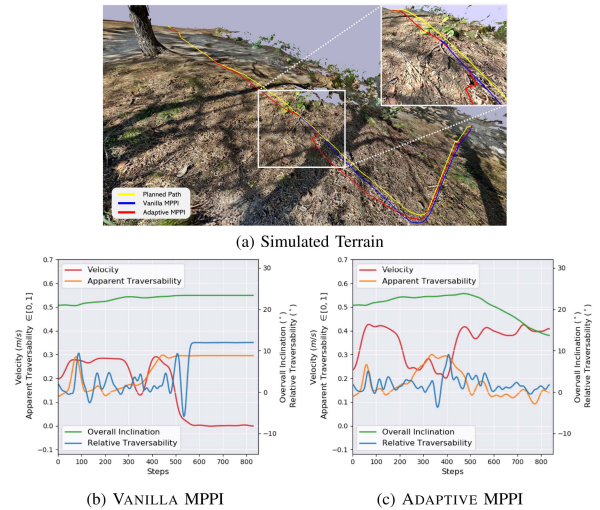


Fig. 6. Resilient behavior for adaptive control. (a) Shows the trajectories that the robot has followed using each control method. (b) Without resilient behavior, the robot's velocity becomes zero due to bumpy terrain with a steep inclination. (c) With resilient behavior, the robot dynamically adjusts the velocity according to the relative traversability. Note that ADAPTIVE refers TAO-Ctr.

Fig. 6 illustrates how our method dynamically adjusts the robot's velocity in an environment that has a steep inclination and uneven terrain. To pass through such an environment, the robot must momentarily increase its velocity. In Vanilla MPPI, the robot gets stuck in areas where the relative traversability suddenly increases, resulting in the velocity eventually dropping to zero. In contrast, in Adaptive MPPI, the robot momentarily accelerates when relative traversability suddenly increases.

Adaptive control optimization can be easily integrated into existing controllers. We compare the navigation performance of different controllers, MPC and MPPI in Table I. Using MPC shows better performance at the easy level. On the other hand, as the environment becomes more challenging, MPPI, which is less sensitive to the motion model, shows better performance.

F. Real-World Navigation

We set up a real Husky robot similar to the simulated robot. Intel NUC with i7 CPU and OS0-64 LiDAR is mounted on the robot. We utilize FAST-LIO2 [33] for localization and RRT³ [34] for generating a search tree.

We compare TAO with PUTN [9] in a real-world environment. The experiment is conducted in the same environment as Figs. 4(d) and 5 and the result is shown in Fig. 7. PUTN generates a path that cannot circumvent the steep slope, resembling the behavior of DEM. Moreover, the real-world environment exhibits more slippery than the simulated environment, PUTN

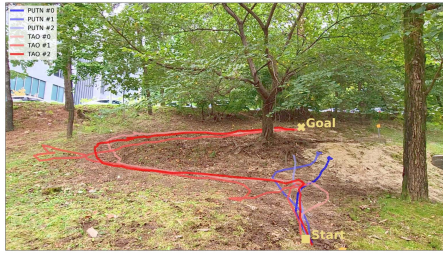


Fig. 7. Real-world navigation. TAO (TAO-plan + TAO-Ctr) successfully reaches the goal while PUTN fails.

fails to progress as expected due to a strategy of reducing velocity in high traversability areas. On the other hand, generates a circumvent path by marking a highly inclined area as a constraint and successfully reaches the goal by regulating the velocity. Note that TAO #1 exhibited a substantial deviation from other trajectories as a result of slippage.

VI. CONCLUSION

In this letter, we present a method for safely and efficiently navigating in extreme mountainous terrain. We assess the terrain with precisely designed features and generate a path that the robot can faithfully execute. Resilient behavior for robust navigation is achieved through adaptive control optimization by considering ground-robot interaction. Furthermore, our method can easily be combined with existing planning and control methods. While our method focuses solely on geometric features, incorporating visual features could be a valuable extension to enhance generalizability across seasonal changes in terrains.

REFERENCES

- [1] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1373–1385, 2013.
- [2] L. C. Santos, A. S. Aguiar, F. N. Santos, A. Valente, J. B. Ventura, and A. J. Sousa, "Navigation stack for robots working in steep slope vineyard," in *Proc. Intell. Syst. Appl.: Proc. Intell. Syst. Conf.*, 2021, pp. 264–285.
- [3] Z. Zhao and L. Bi, "A new challenge: Path planning for autonomous truck of open-pit mines in the last transport section," *Appl. Sci.*, vol. 10, no. 18, 2020, Art. no. 6622.
- [4] T. Hines et al., "Virtual surfaces and attitude aware planning and behaviours for negative obstacle navigation," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 4048–4055, Apr. 2021.
- [5] M. Paton et al., "Navigation on the line: Traversability analysis and path planning for extreme-terrain rappelling rovers," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 7034–7041.
- [6] S. L. Laubach, J. Burdick, and L. Matthies, "An autonomous path planner implemented on the Rocky 7 prototype microrover," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1998, pp. 292–297.
- [7] M. Castelнови, R. Arkin, and T. R. Collins, "Reactive speed control system based on terrain roughness detection," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2005, pp. 891–896.
- [8] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *J. Field Robot.*, vol. 26, no. 3, pp. 308–333, 2009.
- [9] Z. Jian et al., "PUTN: A plane-fitting based uneven terrain navigation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 7160–7166.
- [10] L. Gan, R. Zhang, J. W. Grizzle, R. M. Eustice, and M. Ghaffari, "Bayesian spatial kernel smoothing for scalable dense semantic mapping," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 790–797, Apr. 2020.
- [11] T. Shan, J. Wang, B. Englot, and K. Doherty, "Bayesian generalized kernel inference for terrain traversability mapping," in *Proc. Conf. Robot Learn.*, 2018, pp. 829–838.
- [12] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should I walk? Predicting terrain properties from images via self-supervised learning," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1509–1516, Apr. 2019.
- [13] M. A. Bekhti and Y. Kobayashi, "Regressed terrain traversability cost for autonomous navigation based on image textures," *Appl. Sci.*, vol. 10, no. 4, 2020, Art. no. 1195.
- [14] L. Gan, J. W. Grizzle, R. M. Eustice, and M. Ghaffari, "Energy-based legged robots terrain traversability modeling via deep inverse reinforcement learning," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 8807–8814, Oct. 2022.
- [15] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.
- [16] S. Koenig and M. Likhachev, "D* lite," in *Proc. 18th Nat. Conf. Artif. Intell.*, 2002, pp. 476–483.
- [17] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [18] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, "Terrain characterization and classification with a mobile robot," *J. Field Robot.*, vol. 23, no. 2, pp. 103–122, 2006.
- [19] Y. Tanaka, Y. Ji, A. Yamashita, and H. Asama, "Fuzzy based traversability analysis for a mobile robot on rough terrain," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 3965–3970.
- [20] M. Ono, T. J. Fuchs, A. Steffy, M. Maimone, and J. Yen, "Risk-aware planetary rover operation: Autonomous terrain classification and path planning," in *Proc. IEEE Aerosp. Conf.*, 2015, pp. 1–10.
- [21] M. Thoresen, N. H. Nielsen, K. Mathiassen, and K. Y. Pettersen, "Path planning for UGVs based on traversability hybrid A," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1216–1223, Apr. 2021.
- [22] S. Daftry et al., "MLNav: Learning to safely navigate on Martian terrains," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 5461–5468, Apr. 2022.
- [23] S. Siva, M. Wigness, J. G. Rogers, L. Quang, and H. Zhang, "Self-reflective terrain-aware robot adaptation for consistent off-road ground navigation," *Int. J. Robot. Res.*, vol. 43, 2024, Art. no. 02783649231225243.
- [24] S. Siva, M. Wigness, J. Rogers, and H. Zhang, "Enhancing consistent ground maneuverability by robot adaptation to complex off-road terrains," in *Proc. 5th Annu. Conf. Robot Learn.*, 2021, pp. 1200–1210.
- [25] L. Wellhausen and M. Hutter, "Rough terrain navigation for legged robots using reachability planning and template learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 6914–6921.
- [26] K. Weerakoon, A. J. Sathyamoorthy, U. Patel, and D. Manocha, "TERP: Reliable planning in uneven outdoor environments using deep reinforcement learning," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 9447–9453.
- [27] E. Levina and P. Bickel, "The Earth mover's distance is the mallows distance: Some insights from statistics," in *Proc. IEEE 8th Int. Conf. Comput. Vis.* 2001, pp. 251–256.
- [28] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific J. Math.*, vol. 145, no. 2, pp. 367–393, 1990.
- [29] K. Kozłowski and D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot," *Int. J. Appl. Math. Comput. Sci.*, vol. 14, no. 4, pp. 477–496, 2004.
- [30] E. Coumans and Y. Bai, "Pybullet, a Python module for physics simulation for games, robotics and machine learning," 2016.
- [31] L. Grüne, J. Pannek, L. Grüne, and J. Pannek, *Nonlinear Model Predictive Control*. Berlin, Germany: Springer, 2017.
- [32] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *J. Guid., Control, Dyn.*, vol. 40, no. 2, pp. 344–357, 2017.
- [33] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.
- [34] M. Otte and E. Frazzoli, "RRT^X: Asymptotically optimal single-query sampling-based motion planning with quick replanning," *Int. J. Robot. Res.*, vol. 35, no. 7, pp. 797–822, 2016.