2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)
October 19-25, 2025. Hangzhou, China

Vibration-Aware Trajectory Optimization for Mobile Robots in Wild Environments via Physics-Informed Neural Network

Aochun Xu[†,1], Andong Yang[†,2], Wei Li[*,1], *Member, IEEE*, Yu Hu[*,1], *Member, IEEE*

2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3315-4393-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IROS60139.2025.11246622

***Abstract*— The suspension system, through effective damping of vibrations and shocks, can enhance the stability of wheeled robots traversing challenging terrain. Because the suspension system decouples the rigid correspondence between terrain changes and robot vibrations, considering suspension modeling in trajectory planning offers the advantage of more accurate prediction of the robot's response to terrain. This improved predictive capability facilitates the planning of safer trajectories and may reduce tracking errors in the subsequent control process. In this work, inspired by the structure of Physics-Informed Neural Network (PINN), we propose a physics-informed planning method that considers the vibrational effects of complex nonlinear suspension systems. In addition, we design a two-stage process to accelerate training. By incorporating PINN, our method can better guarantee the physical feasibility of the planned trajectories. The proposed approach has been evaluated on a real robot platform. Compared to state-of-the-art baseline methods, our proposed approach achieves a 15.38% reduction in hazardous planning for mobile robots in wild environments.**

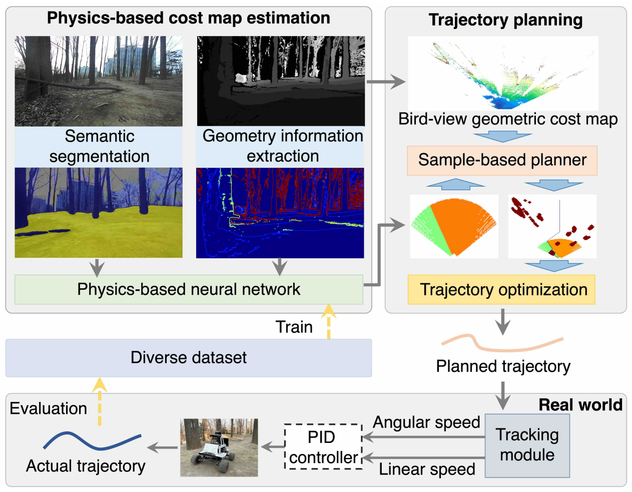


Fig. 1: This article proposes a trajectory planning framework for mobile robots in wild environments that considers the robot's suspension system.

I. Introduction

Trajectory planning is a critical part for mobile robots to accomplish high-automation tasks, such as planetary exploration, precision agriculture, search and rescue. These applications typically operate in wild environments characterized by rough terrain and varying surface types (e.g., grass, gravel, sand, mud), which pose challenges to trajectory planning methods in terms of safety and smoothness.

In such environments, the vibration of mobile robots is significant, which affects the accuracy of various sensors and control systems, and in severe cases, may damage the robot [1]. Therefore, when planning trajectories, it is necessary to consider the impact of different terrains on the robot's vibration. Existing methods typically extract this vibration from the semantic and geometric information of the terrain, and calculate the trajectory by assuming that terrain changes are consistent with the changes in the robot's body [2]–[4]. However, in extreme wild environments, where terrain undulations are large, the impact of the robot's suspension system cannot be ignored. The assumption that terrain changes are consistent with the robot's body changes is difficult to hold in such cases.

Incorporating suspension models into trajectory planning for wheeled robots aids in assessing traversability across uneven ground, allowing for the avoidance of areas that exceed the robot's capabilities. Moreover, because the suspension system absorbs shocks and distributes weight asymmetrically across the wheels, modeling suspension helps the planner predict the robot's response to terrain disturbances and can improve control performance, leading to reduced tracking errors. This is especially critical in challenging environments where the robot is susceptible to tipping or losing balance. Currently, there are mathematical analytical models for modeling suspension systems, such as the spring-mass-damper model [5], [6]. However, due to the complexity of the suspension system, these simplified models lead to inaccurate estimation of the environmental impact on the robot's state, resulting in degraded or failed planning outcomes, especially in wild environments. In addition, such a non-linear module has a high computational complexity.

Learning-based methods are emerging as a promising alternative for better suspension modeling, facilitated by the advancements in deep learning. Nevertheless, solely data-driven network learning struggles to meet stringent constraints, such as the inherent dynamics of robots. To address the aforementioned challenges, we have developed a physics-informed trajectory planning method that accounts for the

This work was supported by Beijing Natural Science Foundation (L243008), and in part by National Natural Science Foundation of China under Grant No. 62003323 and No. 62176250.

† Authors contributed equally to this work.

* Corresponding author. Email: liwei2019@ict.ac.cn, huyu@ict.ac.cn.

[1] Aochun Xu, Wei Li and Yu Hu are with the Research Center for Intelligent Computing Systems, Institute of Computing Technology, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China. {xuaochun22z, liwei2019, huyu}@ict.ac.cn.

[2] Andong Yang is with the Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China, and also with Xinchen Qihang Inc.

robot's suspension system. Motivated by the recently developed physics-informed neural network (PINN), we designed a network for accurate and efficient suspension modeling. A subsequent cost map module is designed to provide more precise environmental information, increasing the safety of the integrated optimization-based trajectory planner. By incorporating PINN, the underlying physics helps ensure that the generated trajectories adhere more closely to physical feasibility, reducing the need for extensive post-processing compared to traditional planning methods and improving robustness in noisy or complex environments.

Two significant challenges remain in implementing PINN-based suspension modeling. The first is to determine how to conveniently acquire force feedback data from wheeled robots for training. The second is that, despite the fact that PINN is more data-efficient than purely data-driven methods, it still requires a considerable amount of training data and may suffer from convergence issues. Thus, we develop a two-stage training scheme: the first stage uses the imprecise model to guide the gradient direction, and the second stage employs supervised learning to quickly improve accuracy. As for the force feedback data, since it is difficult and expensive to mount force sensors on compact wheeled robots, we introduce a force estimation method based on the weight and historical state of the mobile robot. In practice, the estimated force can reflect the force applied by the ground to the robot, which is sufficient to capture physical properties and effectively guide the direction of the gradient during the early stages of network training.

In this paper, we propose a physical-informed trajectory planning (PITP) method, which models the suspension of mobile robots and extends the ability of trajectory planning in wild environments. In summary, our contributions are as follows.

1) Propose a physical-informed neural network-based method to effectively model the suspension of a mobile robot in wild environments. By embedding physical principles into network loss, our PINN-based model achieves superior accuracy in predicting suspension behavior, enabling more reliable trajectory planning in challenging terrains.
2) Present a force estimation method that reflects the force applied by the ground to the robot. This provides the necessary training data for the PINN in a cost-effective manner, as it does not require additional force sensors.
3) Design a two-stage training scheme to accelerate the training of PINN. This scheme leverages a combination of pre-training and fine-tuning techniques to optimize the learning efficiency and reduce the overall training time.

In addition, the proposed method is deployed on a real robot constructed on a hierarchical stack to verify its performance.

II. Related Works

A. Trajectory Planning in wild Environments

Trajectory planning involves determining a feasible and optimal path for a vehicle or robot to travel from an initial position to a goal, while satisfying various constraints such as vehicle dynamics, obstacle avoidance, and safety requirements [7]. Trajectory planning in wild environments requires a greater emphasis on safety considerations, such as rollover prevention and limiting vibrations that could damage the chassis or sensors.

Early trajectory planning methods predominantly rely on optimization techniques, where the goal is typically to minimize a cost function. These approaches, including model predictive control (MPC) [8], Potential Field (PF) [9], and Timed-Elastic-Band(TEB) [10], are well suited for handling robot kinematics and dynamic constraints, but can be computationally expensive for real-time applications [2]. In more complex environments, sampling-based methods such as Rapidly-Exploring Random Trees (RRT) and Probabilistic Roadmaps (PRM) are commonly employed [11]. These techniques generate feasible paths by randomly sampling the state space and connecting feasible configurations. Although these methods are relatively efficient and flexible, they may fail to generate smooth trajectories or provide optimal solutions in extreme wild environments [12].

Learning-based planning methods can learn the distribution of optimal trajectories from data, enabling them to handle environments with complex dynamics and non-convex constraints, where analytical models are difficult to obtain. In current research, self-supervised learning for traversability [13], imitation learning [14] and reinforcement learning [15], [16] are three prominent trends. With the goal of executing covert missions, CoverNav [15] proposes a deep reinforcement learning method to compute a local costmap that indicates paths with maximal covertness in unstructured outdoor environments. SCOML [16] designs a meta-reinforcement learning architecture to handle trajectory planning in hybrid terrains. ViPlanner [17] generates local paths based on geometric and semantic information, and is trained end-to-end using imperative learning. However, these studies neglect suspension modeling for simplification, which is crucial in dealing with rugged terrain. Currently, research on suspension systems is focused more on vehicles [6], [18]. In this work, we incorporate suspension modeling into the planning process for outdoor wheeled robots.

B. Physics-informed Neural Network

Physics-informed neural network (PINN) [19] was first proposed to solve data-driven solutions and data-driven discovery of partial differential equations (PDEs), which are also known as forward and inverse problems. In this paper, we are mainly focused on the inverse problem, which infers the unknown parameters of a physical system from the observed data and the governing equations. By embedding physical information within the loss function, PINN is trained to simultaneously minimize data error and physics-informed error, ensuring that the predicted results adhere to the underlying physical principles. Recently, PINN has demonstrated their effectiveness in solving a variety of problems, including solid mechanics [20], [21], fluid mechanics [22], [23], and geophysics [24], etc.

As for robot navigation and planning tasks, EikoNet [25] uses PINN to solve the Eikonal equation for modeling the time field concerning seismology, which is a nonlinear first-order partial differential equation encountered in wave propagation problems. The equation characterizes the first-arrival time field in heterogeneous 3D velocity structures. Building upon this, Ni et al. [26] propose a more manageable form of the Eikonal equation and introduce a new progressive learning strategy to train neural networks without expert data in complex, chaotic, and high-dimensional robotic motion planning scenarios. Subsequently, the authors of this work improve a physics-informed constrained motion planning (CMP) framework [27] that can solve the Eikonal equation on constrained manifolds.

III. Method

We discuss our novel planner designed to enable ground robots to navigate in unstructured wild environments. An overview of our approach is shown in Fig. 1. The major stages are as follows: 1. Building dataset and training the PINN; 2. Computing a navigation cost-map based on the PINN and local observations; 3. Generating locally least-cost trajectories; 4. Sending trajectories to the motion control module and executing corresponding actions.

A. Trajectory Planning

Trajectory planning is a fundamental module for mobile robots in wild environments. At each time instance t, the objective of this module is to find a feasible path τ_t^* with minimum travel time for mobile robots that connects the start and target position, given the dynamic model ϕ of the robot and the observations. In this work, observations include the RGB images $\mathbf{I}_t$, depth images $\mathbf{D}_t$, and postures $\mathbf{s}_t$ calculated from the Inertial Measurement Unit (IMU).

Let the robot's environment space be $\mathcal{X} \in \mathbb{R}^n$, where $n \in \mathbb{N}$ represents dimensionality. The obstacles in the environment, denoted as $\mathcal{X}_{obs} \subset \mathcal{X}$, are obtained based on RGB and depth images. The process of generating $\mathcal{X}_{obs}$ is as follows: first, to consider the geometric information of the environment, all areas with height changes beyond the capabilities of the mobile robot are marked as impassable, including step-like terrain and excessively steep slopes. Second, a semantic segmentation network, GANav [28], is used to obtain the semantic types of terrain based on RGB images. We classify the terrain type into three categories, as shown in Table I, and regions classified as obstacle are marked as impassable. Meanwhile, areas marked as other types are considered accessible. An example of the obstacle detection process is shown in Fig. 2. The union of these two sets gives the set of obstacles $\mathcal{X}_{obs}$. Then a feasible space $\mathcal{X}_{free} \subset \mathcal{X} \backslash \mathcal{X}_{obs}$ is formed.

In this work, we also consider the effect of the suspension system. So, there is another space $\mathcal{X}_{PINN} \in \mathbb{R}^n$ representing the effect of the terrain on the suspension system. In this cost map, a higher cost indicates greater vibration of the robot's main body under the current terrain, which may lead to a decline in the quality of various sensor data and potentially affect the robot's safety. Training a PINN network requires vibration data from a real robot, which involves significant time and cost for data collection. To reduce training complexity, the cost map is structured by dividing the 110-degree field of view in front of the robot into 75 groups. For each direction, the impact of the terrain on the robot is predicted, guiding the trajectory planning, as shown in Fig. 3. The vibrations of each group are mapped to the obstacle map as a fine sector, and if the normalized vibrations are greater than a set threshold (e.g. 0.5), the area is set as an obstacle. Based on the $\mathcal{X}_{PINN}$ and dynamic model of the robot ϕ, the feasible space $\mathcal{X}_{Pfree}$ is formed. Finally, the feasible space for the mobile robot is $\mathcal{X}^* = \mathcal{X}_{free} \cap \mathcal{X}_{Pfree}$. So, the objective of the robot trajectory planning algorithm is to find a trajectory $\tau \subset \mathcal{X}^*$ that connects the given robot start position $\mathbf{s}_{start} \in \mathcal{X}_{free}$ and the goal position $\mathbf{s}_{goal} \in \mathcal{X}_{free}$.

An example of the final map $\mathcal{X}^*$ used for planning is shown in Fig. 3. We use A* to search for a feasible trajectory and use trajectory optimization to obtain a trajectory represented by the B-spline [29]. At time t, the point on the planned trajectory is $p(t) = \mathcal{B}_{\boldsymbol{q}}(t)$. In this work, we use a fixed time interval $\boldsymbol{T}$ to simplify the problem. An example of a planned trajectory is also shown in Fig. 3.

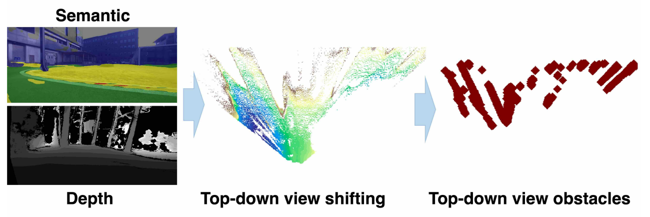


Fig. 2: Obstacle detection process, where all areas with height changes beyond the capabilities of the mobile robot and those classified as impassable based on semantics are marked as obstacles.

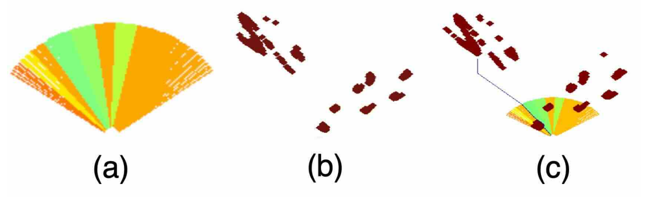


Fig. 3: Example of planning process. (a) The PINN-based cost map, where areas with colors closer to green indicate the terrain is easier to pass, while dark red indicates it is more difficult to pass. (b) The top-down view height map, where height variations exceeding a certain threshold are marked as obstacles. (c) The planned path and the final map, which includes environmental information and the physical constraints of the mobile robot's suspension system.

B. Physics-informed Framework

In this work, the suspension system of the mobile robot is considered in the planning process to achieve more accurate environmental assessments. This improves the accuracy of

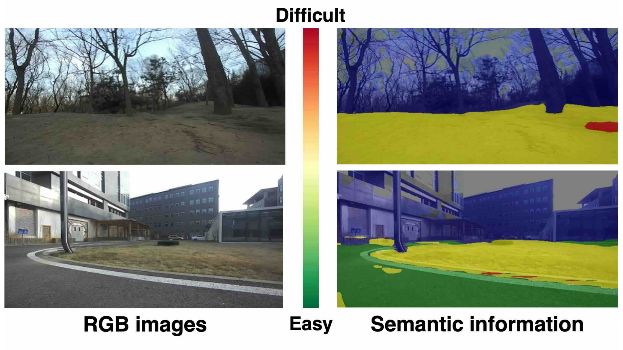


Fig. 4: Examples of the extracted terrain types, with a redder color indicating greater difficulty in traversing.

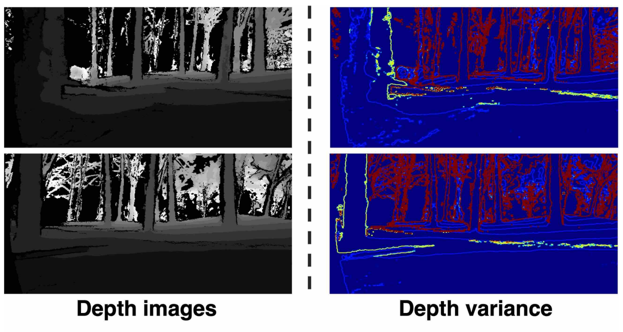


Fig. 5: Examples of the processed depth images.

estimating robot traversal costs $R = (\mathbf{I}_t, \mathbf{D}_t, \mathbf{s}_t)$ based on observations $\mathbf{I}_t, \mathbf{D}_t, \mathbf{s}_t$, leading to better trajectory planning results. We use $f_\theta(\mathbf{I}_t, \mathbf{D}_t, \mathbf{s}_t)$, a neural network parameterized by θ, to fit the mapping of the robot's traversal cost R. This cost map will be used for subsequent planning.

Since the observation is high-dimension data and requires a certain size network to process, we extract information and shrink the input size to reduce the difficulty of training the PINN f. For I_t, we extract semantic information with method [28], as semantic information affects the vibration the most among the information from RGB images. Examples are shown in Fig. 4, a redder color indicating greater difficulty traversing. This terrain type information $\varphi(I_t)$ is easier to process. For depth information, we select the degree of undulation as input for the PINN f. Compared to raw depth images, incorporating terrain undulation information, represented by the variance of depth values, can accelerate training. For each pixel in the depth image $\mathbf{D}_t$, the variance of depth between the current position and the 8 surrounding positions is calculated as $\psi(\mathbf{D}_t) = 1/8(conv(\mathbf{D}_t, \Theta) - \mathbf{D}_t)^2$, where Θ is a convolutional kernel represents the neighborhood of each pixel. It is a 3×3 matrix where all values are 1, except for the center, which is 0. Examples of depth variance are shown in Fig. 5.

The output of the physics-informed network $f_\theta(\mathbf{I}_t, \mathbf{D}_t, \mathbf{s}_t)$ is a vibration estimation $\mathcal{X}_{PINN}$ that indicates the smoothness of the robot body when crossing the observed terrain. The smoothness will affect the quality of sensor and control accuracy, so it is better to choose a smoother path for mobile robots, especially in wild environtments. We use the real vibration data from the IMU to train the network.

TABLE I: Terrain types and corresponding types.

Terrain types	Pixel-wise labels	value
Smooth terrain	Concrete, Asphalt	Accessible
Rough terrain	Soil, Grass	Accessible
Bumpy terrain	Mulch, Rubble, Puddle, Mud	Accessible
Restricted terrain	Trees, Logs, person, barrier	Obstacles
Background	Void, Sky	Ignored

C. Training Scheme

The architecture of the neural network employs a multi-head approach, where semantic information $\varphi(I_t)$ extracted from RGB images and undulation information $\psi(\mathbf{D}_t)$ from depth images are processed separately to extract modality-specific features. The RGB branch utilizes a lightweight backbone MobileNetV3 [30], to extract features from the RGB image. Similarly, the depth branch processes the input using a separate MobileNetV3 backbone. Both branches produce feature vectors of length 576 through global average pooling. These feature vectors are then concatenated to combine complementary information from both modalities. The fused features are passed through a fully connected layer of 256 neurons to predict vibration. This multi-head design allows the network to effectively leverage the strengths of both information from RGB and depth images while maintaining computational efficiency.

The loss function is designed to optimize the network for accurate vibration prediction, while incorporating physical constraints from a spring-damper model [5], [6]. The loss L consists of two parts, and the first is the Mean Squared Error (MSE) loss, which measures the difference between the predicted vibration level and the ground truth. This ensures that the network predictions are close to the real vibration values. $L_1 = ||y_t^* - y_t||_2$ where y_t^* is the real vibration, y_t is the predicted vibration.

Second, to incorporate physical constraints, a physics consistency loss term is introduced. This loss is based on the equation of a damped spring-mass system $mx^{''} + cx^{'} + kx = F(t)$, where m is the mass of the robot, $x^{''}$ is the acceleration of the robot, $x^{'}$ is the velocity, x is the position of the robot body, c is the damping coefficient, k is the spring constant, modeling the elastic properties of the suspension, and $F(t)$ is the force applied by the ground to the robot. The vibration in this work is represented as the acceleration of the z axis, so the predicted vibration has the same trend as the acceleration. Since it is expensive and difficult to install a force sensor on the mobile robot, we estimate $F(t)$ based on the state $\mathbf{s}_t$ and mass of the robot m. Specifically, when collecting training data, the robot is set to move at a constant speed in a straight line. Any changes in the robot's acceleration are mainly caused by the forces exerted by the ground on the robot. Therefore, based on the mass of the robot and the changes in the robot's acceleration, the force applied by the ground to the robot can be estimated using Newton's second law. The estimated force $F^*(t)$ is not accurate but is sufficient to reflect the change in real force $F(t)$. So the physics-based loss is

$$L_2 = (mf_{\boldsymbol{\theta}}(\mathbf{I}_t, \mathbf{D}_t, \mathbf{s}_t) + c\mathbf{s}_t^{'} + k\mathbf{s}_t - F^*(t))^2 \tag{1}$$

where m is the mass of the robot chassis, c is the damping coefficient, k is the spring constant. These parameters were measured using the existing method [31].

The total loss $L = \alpha L_1 + \beta L_2$ is a weighted sum of the MSE loss and the physics consistency loss, where α and β are normalized hyperparameters controlling the weight of the physics term. We design two loss functions to enhance the performance of the algorithm, with the data-driven loss focusing on accuracy and the physics-based loss ensuring physical consistency.

Since the estimated force and the simplified physics-based loss are not accurate, we design a two-stage training scheme to train the PINN. In the first stage, we start with a low value of α and let physics-based loss guide the training. Due to the diversity of wild environment scenes, during training, randomly sampled batches may include a small number of distinctly different scenarios, causing drastic changes in the network parameters and leading to forgetting of previous learning. Existing methods to alleviate such drastic changes [32] are computationally expensive. In this work, we employ a threshold-limiting approach. A threshold η is set between the epochs t and $t-1$ to restrict the loss L_{t-1} and L_t. When the loss change exceeds the limit $L_{t-1}/L_t > \eta$, the batch of data is resampled for training. In the second stage, we set a higher value of α and let supervised loss guide the training to improve accuracy. The trajectory planning process after training is shown in Algorithm 1.

Algorithm 1 Trajectory planning

```
1:  Initialize net $f$ with trained parameters $\boldsymbol{\theta}$, target position
    $\boldsymbol{s}_{goal}$, maximum iteration $K$.
2:  Initialize current target position $\boldsymbol{s}_{glocal}$.
3:  while not reach the final target $\boldsymbol{s}_{goal}$ and $K$ do
4:      Update current state $\boldsymbol{s}_t, \boldsymbol{I}_t, \boldsymbol{D}_t$
5:      Update $\varphi(\boldsymbol{I}_t)$ and $\psi(\boldsymbol{D}_t)$ to get semantic and geo-
        metric information of the environment.
6:      Determine next local target $\boldsymbol{s}_{glocal}$ based on $\boldsymbol{s}_t$ and
        final target position $\boldsymbol{s}_{goal}$
7:      Calculate cost map $\mathcal{X}_{Pfree}$, $\mathcal{X}_{free}$ and the feasible
        space $\mathcal{X}^*$
8:      Find initial path $\tau \subset \mathcal{X}^*$
9:      Generate trajectory $p(t) = \mathcal{B}_{\boldsymbol{q}^*}(t), t \in \boldsymbol{T}$
10:     Record current observation
11:     Send trajectory to the tracking module
12:     Obtain control actions $\boldsymbol{a}_t$ and execute
13: end while
```

D. Trajectory Tracking

At time instance t, given the planned trajectory τ_t^* represented by the B-spline and the current state $\mathbf{s}_t$ of the robot estimated based on IMU data. The tracking module calculates the specific control action $\boldsymbol{a}_t$ and executes. In this work, the action is linear and angular velocity. The Scout-mini chassis drive receives the velocities and converts them into control commands for execution. The module selects the next target point that is closest to the current position of the robot but still ahead on the trajectory at a fixed distance. If there is no planned trajectory at a fixed distance from the current position, the robot will select the point on the trajectory that is closest to its current position as the target point. If the distance between the robot and the trajectory exceeds two meters, the tracking will end.

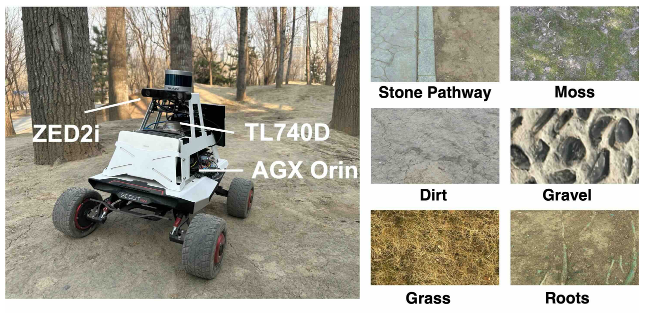


Fig. 6: The mobile robot used in experiments. In the experiments, the materials of the terrain include dirt, grass, gravel roads, moss, and roots. The geometric features of the terrain include tree roots, uphill and downhill slopes, etc.

IV. Experiments and Results

A. Hardware Setup

We use a modified Scout-mini robot from AgileX as our experiment platform, as shown in Fig. 6. The scout is equipped with a Stereolabs ZED 2i camera, an IMU RION TL740D as sensors, and a NVIDIA AGX Orin as the onboard computer. The ZED2i camera has a field of view of 110 degrees horizontally, with a depth range of 0.3 to 15 meters. The resolution is HD 720, and the image acquisition frame rate is set to 30Hz. Our planner works at 10Hz.

B. Dataset

The data used to train the PINN were collected by Scout-mini in an outdoor environment with varying terrain and obstacles. A total of 20k frames were collected across three segments. Each frame includes an RGB image and a depth image provided by the ZED 2i camera, as well as IMU data. Simulators cannot model real complex suspensions and vibrations, so we chose to use real-world data.

Terrain classification. The RGB images are resized to 300 × 375 and then processed through GANav [28] for terrain classification. The middle and lower parts of the results, with a size of 50 × 5, are extracted as input to the PINN, as this section of the terrain is most likely to influence the vehicle's vibration in the following seconds.

Variance of depth information. The variance of the depth information is calculated for each pixel in the 600 × 675 depth image by calculating the variance of the eight pixels surrounding that pixel. These variance data are also used as input to the PINN, as they describe the roughness of the terrain ahead.

Vibration indicator. The linear accelerations in the z-axis direction of the IMU data, after subtracting the gravitational acceleration, are extracted to describe the vibration of the Scout-mini at that moment. The average values of these data over the next 2 seconds are then calculated, normalized, and used as the output label for the PINN, representing the vibration indicator.

C. Comparing with the State-of-the-Art

We compare our proposed method with traditional and learning-based methods. First, we select a traditional method [33] that combines a rapidly exploring random tree algorithm with 3D object detection methods and MPC, which does not consider the undulations of the terrain.

RAORN [34] is a learning-based navigation algorithm that can predict the achievable speed distribution of the robot based on environmental semantic information and convert it into a costmap. This robot-speed-based traversability representation enables the algorithm to traverse rough terrain to some extent, but it does not account for the physical impact of terrain on the robot.

A fully trained deep reinforcement learning (DRL) network, Terp [35], takes the elevation map of the environment, robot pose, and goal as input and computes an environment attention mask. This helps the costmap encode information about elevation or flatness of the terrain, thereby identifying local minimum-cost navigation points.

BADGR [36] is an end-to-end reinforcement learning navigation system that can be trained using supervised, off-policy data collected from different environments and terrains. It adapts to new environments and continuously improves autonomously by collecting more data.

Finally, we compare with a trajectory planning network, SCOML [16], capable of handling mixed terrains. It uses a self-correction structure based on historical planning data to correct inappropriate trajectories. The above learning-based algorithms consider the impact of terrain on navigation to varying degrees, but lack an understanding of the physical models between the robot and the terrain, which may result in planning that is not physically feasible.

We conducted experiments on a 300-meter-long path in environments with different terrain, as shown in Fig. 6. We use the following metrics to quantitatively compare our method with others.

- **Success Rate.** The success rate is the frequency at which the mobile robot successfully reaches the target point while performing a navigation task. A successful task must avoid collisions during travel, and the roll angle must not exceed 30 degrees.
- **Hazard.** Hazard refers to the number of dangerous decisions made by the robot during its travel, including collisions, skidding, and instances where the robot's roll angle exceeds the safe rollover limit, which can lead to dangerous situations. This metric is calculated as the average value every 50 meters.
- **Smoothness.** Smoothness is defined as the cumulative difference in yaw angle between two consecutive moments as the robot moves along the planned route during navigation, normalized by the total length of the path to assess the performance of different algorithms. A larger value of this indicator indicates that the planned trajectory is more winding and difficult to traverse.
- **Consistency.** Consistency refers to the L2 distance between the planned trajectory and the actual traveled trajectory. A larger value of this metric indicates that the control module has more difficulty guiding the robot to follow the planned trajectory.
- **Normalized Trajectory Length.** This metric is obtained by dividing the length of the planned path by the straight-line distance from the starting point to the planned endpoint.

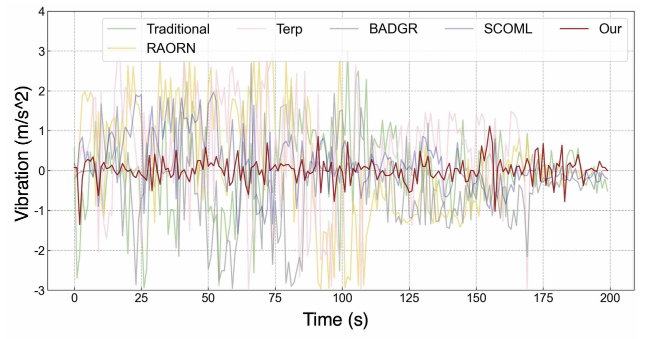


Fig. 7: Comparison of the vibrations of our method and the baselines during a navigation task.

Table II shows the results of the metrics for our method. The proposed method achieves the highest success rate, indicating that our method is effective in environments containing various terrains mentioned above. At the same time, our algorithm reduces hazardous planning occurrences by 15.38%, since the PINN module incorporates physical information about the terrain and the robot's suspension system, helping the trajectory planning process choose smoother and safer paths. The variation represented as acceleration in the z-axis direction for our method and the baselines during a navigation task is shown in Fig. 7. Since the PINN models the effect of terrain on the suspension system, our approach is able to choose the path with the least vibration.

The traditional method has the highest smoothness, but has a lower success rate and results in more occurrences of hazardous planning. This may be due to the fact that it does not consider terrain undulations and variations, and the planned path may cross dangerous areas. In contrast, our method may require steering to avoid steep terrain, which improves the success rate.

In addition, our method achieved the best consistency, which can be attributed to the physical model. The model incorporates the effects of the suspension system into the

TABLE II: Baseline comparisons and ablation study.

Method	Baseline					Ablation Study	
	Traditional	**RAORN**	**Terp**	**BADGR**	**SCOML**	**No PINN**	**Ours**
Success Rate (%)	46±8	56±4	64±8	70±4	78±4	60±6	**82±7**
Hazard (-)	32±12	23±10	26±13	21±8	13±7	28±10	**11±9**
Smoothness (rad)	**0.58±0.11**	0.84±0.13	0.76±0.09	0.8±0.15	0.7±0.05	0.65±0.13	0.73±0.13
Consistency (m)	10.2±5.3	16.5±6.3	11.8±4.7	9.7±3.8	6.9±3.2	8.9±3.7	**5.1±0.5**
Norm. Traj. Length (-)	1.9±0.1	2.1±0.6	2.3±0.8	2.2±0.4	**1.7±0.2**	1.9±0.3	2.0±0.2

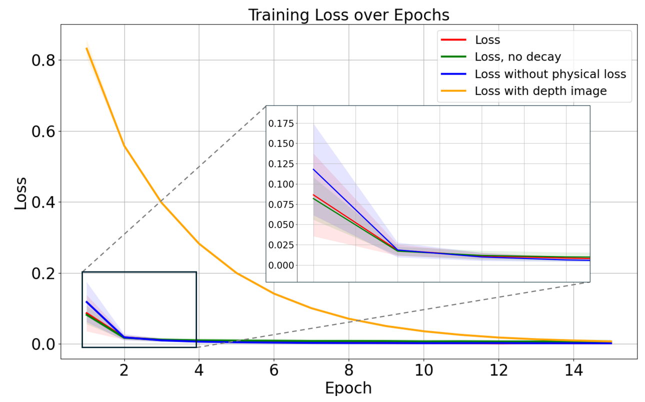
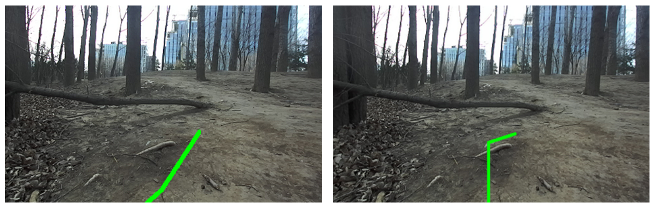


Fig. 8: Loss changes using the depth image directly as input versus using the depth variance as input with or without decay and physical loss.



(a). Trajectory with PINN (b). Trajectory without PINN

Fig. 9: Comparison of trajectories chosen by our method with and without vibration prediction.

planning process and helps to choose a smoother path, which in turn makes the path easier for the tracking module to follow and improves consistency.

The SCOML method achieves the best normalized trajectory length. Similarly to the smoothness metric, this is due to the compromise made by our method to avoid rugged terrain. The SCOML method treats the standardized trajectory length as one of the planning losses, which leads to the best performance. However, because of its focus on shorter paths, it may choose more rugged trajectories, resulting in a poor success rate and consistency.

D. Ablation Study

To evaluate the effectiveness of the PINN and the proposed planning method, we performed the following three ablation experiments: The first experiment replaces the depth variance with the original depth image as input to verify the effectiveness of the smoothness information in the depth variance. The second experiment removes the physical loss during the PINN training process to observe its effect on training performance. The third experiment directly removes the PINN from the planning process and tests the performance of the planning algorithm with and without the PINN.

The variance of the depth image contains terrain undulation information. Compared with raw depth images, we notice that this undulation information can accelerate training and improve the performance of the PINN. Fig. 8 shows the change in training loss when using the depth image directly as input and using the depth variance insdead. The former results in a high physical loss, since the depth image does not directly contain information about the smoothness of the terrain. However, as the network gradually learns, the loss slowly decreases and eventually approaches the performance of the network using depth variance.

The physical loss calculated from estimated force can reflect the force applied by the ground to the robot, which can effectively guide the gradient's direction during the first stages of network training. However, after a certain amount of training, it will prevent the loss from decreasing, so the weight of this physical loss needs to be gradually reduced during the training process. As shown in Fig. 8, the training process with the addition of physical losses converges faster and yields better results.

The trajectory planning module uses the vibration prediction provided by the PINN to account for the potential impact of nearby terrain on the robot and plans a relatively smooth trajectory for the robot. The quantitative results of the metrics after removing the PINN are shown in Table II. Without the vibration prediction provided by the PINN, the performance of our method significantly decreases, with a drop in the success rate and a substantial increase in the number of hazardous planning instances. Fig. 9 shows the planned paths with and without the PINN. It can be seen that without the PINN, the influence of the terrain on the suspension system is ignored and areas such as tree roots that could cause robot vibrations are not considered by the planner. This situation leads to a decline in metrics such as success rate and smoothness. In contrast, our method with vibration prediction chooses a smoother and safer path.

V. Conclusion

In this article, we present a PITP approach for mobile robots in wild environments. To consider the suspension system of the robots during planning, we propose a PINN-based method to estimate the effects of the suspension system in different terrains. To calculate the loss for training the network, we design a novel force estimation method that reflects the force applied by the ground to the robot. In addition, a two-stage training scheme is used to accelerate the training. Future work includes designing an online dataset to automatically train the network during testing.

References

[1] I. E. Gargano, K. D. von Ellenrieder, and M. Vivolo, "A survey of trajectory planning algorithms for off-road uncrewed ground vehicles," in *International Conference on Modelling and Simulation for Autonomous Systems*. Springer, 2025, pp. 120–148.

[2] "A comparative review on mobile robot path planning: Classical or meta-heuristic methods?" *Annual Reviews in Control*, vol. 50, pp. 233–252, 2020.

[3] A. Yang, W. Li, and Y. Hu, "F3dmp: Foresighted 3d motion planning of mobile robots in wild environments," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 643–12 649.

[4] J. Petereit, T. Emter, C. W. Frey, T. Kopfstedt, and A. Beutel, "Application of hybrid a* to an autonomous mobile robot for path planning in unstructured outdoor environments," in *ROBOTIK 2012; 7th German Conference on Robotics*. VDE, 2012, pp. 1–6.

[5] A. G. Mohite and A. C. Mitra, "Development of linear and non-linear vehicle suspension model," *Materials Today: Proceedings*, vol. 5, no. 2, Part 1, pp. 4317–4326, 2018, 7th International Conference of Materials Processing and Characterization, March 17-19, 2017.

[6] M. Hamed, B. Tesfa, F. Gu, and A. D. Ball, "Vehicle suspension performance analysis based on full vehicle model for condition monitoring development," in *Vibration Engineering and Technology of Machinery*, J. K. Sinha, Ed. Springer International Publishing, 2015, pp. 495–505.

[7] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," *Motion and operation planning of robotic systems: Background and practical approaches*, pp. 3–27, 2015.

[8] C. Liu, S. Lee, S. Varnhagen, and H. E. Tseng, "Path planning for autonomous vehicles using model predictive control," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 174–179.

[9] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[10] H. Xi, W. Li, F. Zhao, L. Chen, and Y. Hu, "A safe and efficient timed-elastic-band planner for unstructured environments," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 3092–3099.

[11] S. Dixit, S. Fallah, U. Montanaro, M. Dianati, A. Stevens, F. Mccullough, and A. Mouzakitis, "Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects," *Annual Reviews in Control*, vol. 45, pp. 76–86, 2018.

[12] I. Mir, F. Gul, S. Mir, M. A. Khan, N. Saeed, L. Abualigah, B. Abuhaija, and A. H. Gandomi, "A survey of trajectory planning techniques for autonomous systems," *Electronics*, vol. 11, no. 18, p. 2801, 2022.

[13] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter, "Fast Traversability Estimation for Wild Visual Navigation," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[14] Y. Pan, C. A. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou, and B. Boots, "Agile autonomous driving using end-to-end deep imitation learning," in *Robotics: Science and Systems 2018*, 2018.

[15] J. Hossain, A.-Z. Faridee, N. Roy, A. Basak, and D. E. Asher, "Covernav: Cover following navigation planning in unstructured outdoor environment with deep reinforcement learning," in *2023 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, 2023, pp. 127–132.

[16] A. Yang, W. Li, and Y. Hu, "Scoml: Trajectory planning based on self-correcting meta-reinforcement learning in hybrid terrain for mobile robot," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 14 125–14 132.

[17] P. Roth, J. Nubert, F. Yang, M. Mittal, and M. Hutter, "Viplanner: Visual semantic imperative learning for local navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5243–5249.

[18] G. Kim, S. Y. Lee, J.-S. Oh, and S. Lee, "Deep learning-based estimation of the unknown road profile and state variables for the vehicle suspension system," *IEEE Access*, vol. 9, pp. 13 878–13 890, 2021.

[19] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[20] C. Rao, H. Sun, and Y. Liu, "Physics-informed deep learning for computational elastodynamics without labeled data," *Journal of Engineering Mechanics*, vol. 147, no. 8, p. 04021043, 2021.

[21] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes, "A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 379, p. 113741, 2021.

[22] X. Jin, S. Cai, H. Li, and G. E. Karniadakis, "Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations," *Journal of Computational Physics*, vol. 426, p. 109951, 2021.

[23] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 360, p. 112789, 2020.

[24] U. bin Waheed, E. Haghighat, T. Alkhalifah, C. Song, and Q. Hao, "Pinneik: Eikonal solution using physics-informed neural networks," *Computers Geosciences*, vol. 155, p. 104833, 2021.

[25] J. D. Smith, K. Azizzadenesheli, and Z. E. Ross, "Eikonet: Solving the eikonal equation with deep neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 12, pp. 10 685–10 696, 2021.

[26] R. Ni and A. H. Qureshi, "Progressive Learning for Physics-informed Neural Motion Planning," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[27] ——, "Physics-informed neural motion planning on constraint manifolds," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 12 179–12 185.

[28] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, "Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8138–8145, 2022.

[29] L. Biagiotti and C. Melchiorri, "B-spline based filters for multi-point trajectories planning," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 3065–3070.

[30] B. Koonce and B. Koonce, "Mobilenetv3," *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, pp. 125–144, 2021.

[31] C.-P. Fritzen, "Identification of mass, damping, and stiffness matrices of mechanical systems," *Journal of Vibration, Acoustics, Stress, and Reliability in Design*, vol. 108, no. 1, pp. 9–16, 1986.

[32] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[33] H. Zhou, P. Feng, and W. Chou, "A hybrid obstacle avoidance method for mobile robot navigation in unstructured environment," *Industrial Robot: the international journal of robotics research and application*, vol. 50, no. 1, pp. 94–106, 2023.

[34] X. Cai, M. Everett, J. Fink, and J. P. How, "Risk-aware off-road navigation via a learned speed distribution map," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2931–2937.

[35] K. Weerakoon, A. J. Sathyamoorthy, U. Patel, and D. Manocha, "Terp: Reliable planning in uneven outdoor environments using deep reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9447–9453.

[36] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.