Learning-based Uncertainty-aware Navigation in 3D Off-Road Terrains

Hojin Lee, Junsung Kwon, and Cheolhyeon Kwon

Abstract— This paper presents a safe, efficient, and agile ground vehicle navigation algorithm for 3D off-road terrain environments. Off-road navigation is subject to uncertain vehicle-terrain interactions caused by different terrain conditions on top of 3D terrain topology. The existing works are limited to adopt overly simplified vehicle-terrain models. The proposed algorithm learns the terrain-induced uncertainties from driving data and encodes the learned uncertainty distribution into the traversability cost for path evaluation. The navigation path is then designed to optimize the uncertainty-aware traversability cost, resulting in a safe and agile vehicle maneuver. Assuring real-time execution, the algorithm is further implemented within parallel computation architecture running on Graphics Processing Units (GPU).

I. INTRODUCTION

Autonomous navigation of ground vehicles in an off-road environment has been of vital importance in many applications, such as military operations, agriculture, and planet exploration [1]–[3]. Compared to paved road cases [4], [5], off-road terrains impose an additional challenge to autonomous navigation. This is mainly attributed to the complicated vehicle-terrain interactions, characterized by different factors such as terrain topology, terrain type, etc [1]. These factors are further subject to various uncertainties, making it difficult to assess the vehicle maneuver in reality. To safely and agilely guide the vehicle under such uncertain vehicle-terrain interactions, this paper mainly investigates the following two problems.

- Planning the feasible path of the off-road vehicle while addressing nonlinear vehicle dynamics over 3D terrain topology.
- Accounting for the uncertainties of vehicle maneuver in different terrain conditions, thereby generating a safe yet less-conservative path.

It is worth noting that the existing work focused on either one problem while treating another in an overly simplified sense. For example, some approaches generate a safe path considering 3D terrain topology and 6 DOF vehicle dynamics while ignoring the uncertainty caused by the types of terrains [6]–[9]. On the other hand, other approaches assess the path by different terrain conditions(e.g., roughness, slip coefficients) and apply a simple kinematic model of the vehicle with the assumption of driving on flat terrain [10]–[15]. Such approaches may be acceptable to generate safe
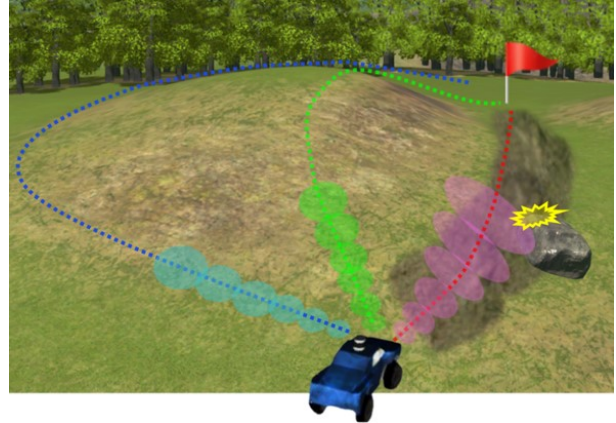
Fig. 1: Example paths along varying terrains. The shortest path (red line) encounters mud areas with higher motion uncertainty (red-ellipsoidal area), which has a higher chance of slipping and colliding with an obstacle. The path detouring the hill (blue line) is completely getting around the mud and slop at the cost of a long path. The proposed algorithm can appropriately compromise both paths and generate an agile yet safe path (green line).

paths in a simple environment. In practice, however, a lack of simultaneous consideration of the above two problems may result in a physically infeasible or too conservative path (See Fig. 1).

This paper proposes a navigation algorithm that can agilely guide a vehicle to the goal while accommodating the uncertainty in the 3D terrain with different terrain conditions (e.g., slop, friction coefficient, roughness). First, raw sensor measurements are processed to construct a geometric traversability cost map, and the terrain type map encodes semantic features (e.g., grass, mud, asphalt, etc.). Then, the vehicle-terrain interactions for individual terrain types are respectively learned through Gaussian Process (GP) regression models. By virtue of the GP models, the terrain-induced uncertainties can be expressed by probability distributions. These distributions are used to predict the actual path distributions of the vehicle when the vehicle follows the candidate path, resulting in *predictive path distributions* [16]. Next, the best path is found by evaluating the cost metrics associated with traversability, rollover risk, and distance to the goal. Note that the cost evaluation comprehends the terrain-induced uncertainty distribution via the Gaussian kernel smoothing technique [17].

The proposed algorithm involves high-dimensional non-linear dynamics in multiple GP models, making it computationally expensive. To run the algorithm through an onboard computing unit of the vehicle, we resort to sampling-based methods. For instance, a finite number of candidate paths

are generated based on the control input sampler. And the predictive path distribution for each path candidate is computed based on sample propagation. Lastly, the sampling-based nonlinear model predictive control (MPC) is applied to follow the resulting path. Furthermore, we leverage the recent advances in GPU parallel computing, which can effectively implement sampling-based algorithms for real-time operation.

II. RELATED WORKS

A. Vehicle Navigation over 3D Terrains

Early works have addressed the navigation on 3D terrains based on the 2D navigation methods while simplifying the original 3D problem. In [18], a multi-layer 2D map extracted from the 3D OctoMap has been utilized. In [19], a path sampling algorithm based on the locally fitted plane has been proposed to generate a global path. They applied GP model to interpolate the local paths computed from each locally fitted plane to make a denser path. In [8], [20], safe navigation areas have been classified based on the labeled 3D point cloud map given by the Lidar sensor. However, such a binary division cannot preclude dynamically infeasible paths or collisions due to the complex vehicle interaction with 3D terrain topology. A more general global path planning algorithm has been proposed in [9]. This method utilizes a potential field function and leverages the Voronoi diagram to improve the computational efficiency while considering the 3D kinematic vehicle model and the terrain topological constraints.

From the perspective of optimal control and local planning, MPC-based approaches have been developed for off-road environments but are limited to driving on flat terrains [21]–[23]. In [6], the author formulated a nonlinear MPC problem for optimal maneuvers on 3D terrains. In this approach, the terrain topology is assumed to be twice continuously differentiable, which might be invalid in practice. Very recently, a learning-based method has indirectly considered the effects of 3D terrains by utilizing the elevation map, and 3D vehicle poses as training features [24]. To sum up, all the aforementioned methods lack the comprehension of the non-linearity of the 3D vehicle dynamics and/or the effect of different terrain conditions.

B. Vehicle Navigation in View of Traversability

A plethora of research has been carried out to address the effect of vehicle-terrain interactions by the name of terrain traversability. The main idea is to evaluate the traversability of the terrain areas and safely navigate only through the traversable areas. The methods for determining the traversable area can be divided into three main categories; i) *geometry-based*; ii) *semantic-based*; and iii) *learning-based*.

First, the geometry-based methods examine the traversability according to the 3D geometry of the surface, i.e., slope, step height, and roughness. The elevation map, created by Lidar point clouds [25], [26], has been used to compute the geometrical features of the 3D surface [27], [28]. Using geometry-based traversability cost, local and global navigation algorithms have been proposed in [19], [29], [30]. However, such methods disregard the vehicle-terrain interactions derived from the terrain conditions, such as friction coefficients.

Meanwhile, the semantic-based methods typically analyze the types of terrain segmented from the camera image [27], [31]–[33]. For example, in [27], [33], [34], the traversability is assigned manually based on the terrain types, i.e., low traversability in the mud area and high traversability in the grass area. The limitation of such methods comes from the fact that heuristic user-interventions are inevitable when designing the traversability cost.

Recent learning-based methods can avoid such heuristics as they exploit the collected data to learn the traversability cost instead of manual cost assignment [11]–[13], [32], [35]. In [14], [32], [35], the roughness of the terrains is learned based on the sensor data, and they plan a smoother path for the vehicle. In [12], a neural network represents traversability as the achievable maximum speed distribution of vehicles on different terrains. Moreover, end-to-end learning approaches have been proposed to avoid geometric obstacles as well as untraversable terrains by analyzing the executed commands and the realized trajectories [11], [13]. Despite rich literature, the developed learning-based methods do not explicate 3D terrain topology, whose impact is too costly to learn from the collected data only.

III. PROBLEM FORMULATION

This section presents the notations, and dynamical vehicle model on 3D terrain topology, followed by the GP model for analyzing the vehicle-terrain interactions, which is the key ingredient of our proposed algorithm.

A. Notations and Definitions

Let us denote a world reference frame as $\{O^{\mathcal{W}}, X^{\mathcal{W}}, Y^{\mathcal{W}}, Z^{\mathcal{W}}\}$ with superscript $\mathcal{W}$, and a vehicle reference frame as $\{O^{\mathcal{B}}, X^{\mathcal{B}}, Y^{\mathcal{B}}, Z^{\mathcal{B}}\}$ with superscript $\mathcal{B}$. And the vehicle orientation with respect to the world reference frame is defined by a set of Euler angles, roll(ϕ), pitch(θ), and yaw(ψ). It is assumed that the roll and pitch of the vehicle can be determined by the surface normal vector and the yaw of the vehicle. The rotation matrix $R_{\mathcal{B}}^{\mathcal{W}}$ defines the transformation from vehicle to world reference frame as follows.

$$R_{\mathcal{B}}^{\mathcal{W}} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

where $c*$ and $s*$ are shorthand notations for $cos(*)$ and $sin(*)$, respectively. Then the gravitational force acting on the vehicle to its center of mass can represented as $F^{\mathcal{B}} = (R_{\mathcal{B}}^{\mathcal{W}})^{-1} F^{\mathcal{W}}$ where $F^{\mathcal{W}} = [0, 0, -mg]^{\mathrm{T}}$ is the weight force in world reference frame. The velocities of vehicle in the world frame are denoted by $v_x^{\mathcal{W}}$, $v_y^{\mathcal{W}}$, $v_z^{\mathcal{W}}$. And the longitudinal, lateral, and vertical velocity components in the body-fixed frame are denoted by $v_x^{\mathcal{B}}$, $v_y^{\mathcal{B}}$, and $v_z^{\mathcal{B}}$. Angular

velocities in the world reference frame can be computed as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \frac{s\phi*s\theta}{c\theta} & \frac{c\phi*s\theta}{c\theta} \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \begin{bmatrix} \omega_x^{\mathcal{B}} \\ \omega_y^{\mathcal{B}} \\ \omega_z^{\mathcal{B}} \end{bmatrix}$$

where $\omega_x^{\mathcal{B}}, \omega_y^{\mathcal{B}}, \omega_z^{\mathcal{B}}$ are the angular velocity in the body-fixed frame. Without loss of generality, the angular velocities $\omega_x^{\mathcal{B}}$, and $\omega_y^{\mathcal{B}}$ are approximated to be zero [6]. L_f, and L_r are the distance to the front and rear axles from the center of gravity (c.g.) of vehicle. L_w is the track width and h is the height of the c.g. of the vehicle measured from the ground. And h_R is the height of the c.g. from the roll center. m is the vehicle mass and I_{zz} is the yaw moment of inertia. $0_a \in \mathbb{R}^{a \times a}$ and $I_a \in \mathbb{R}^{a \times a}$ are zero and identity matrix.

B. Vehicle Dynamics over 3D Terrains

In this paper, the extended dynamic bicycle model is considered [6]. Correspondingly, the vehicle state is defined as $\xi := \begin{bmatrix} x^{\mathcal{W}} & y^{\mathcal{W}} & \psi & v_x^{\mathcal{B}} & v_y^{\mathcal{B}} & \omega_z^{\mathcal{B}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^6$, where $x^{\mathcal{W}}, y^{\mathcal{W}}$ represents the global position of the vehicle along $X^{\mathcal{W}}$, and $Y^{\mathcal{W}}$ axis, respectively. And the control inputs are denoted as $\zeta := \begin{bmatrix} \delta & a_x^{\mathcal{B}} \end{bmatrix}^{\mathrm{T}}$ where δ is the steering angle and $a_x^{\mathcal{B}}$ is the longitudinal acceleration. Then, the vehicle dynamics is described by

$$\dot{\xi} = f(\xi, \zeta) = \begin{bmatrix} v_x^{\mathcal{W}} \\ v_y^{\mathcal{W}} \\ \frac{c\phi}{c\theta}\omega_z^{\mathcal{B}} \\ a_x^{\mathcal{B}} \\ \frac{F_{yf}+F_{yr}-F_G}{m} - v_x^{\mathcal{B}}\omega_z^{\mathcal{B}} \\ \frac{F_{yf}L_f cos(\delta)-L_r F_{yr}}{I_{zz}} \end{bmatrix} \quad (1)$$

where $F_G = -mg \times c\theta s\phi$ is the component of vehicle gravity force in the lateral direction. F_{yf}, and F_{yr} are the lateral tire forces on the front and rear tires calculated using the linear tire model as follows.

$$F_{yf} = C_{\alpha f} F_{zf} \alpha_f, \quad F_{yr} = C_{\alpha r} F_{zr} \alpha_r$$

where $C_{\alpha f}$, and $C_{\alpha r}$ are the front and rear cornering stiffness and α_f, α_r are the front and rear tire slip angles. And the vertical loads can be drawn in the body-fixed frame as follows [36].

$$F_{zf} = \frac{L_r mg \times c\theta c\phi + hma_x^{\mathcal{B}} + hmg \times s\theta}{L_r + L_f}$$

$$F_{zr} = \frac{L_f mg \times c\theta c\phi - hma_x^{\mathcal{B}} - hmg \times s\theta}{L_r + L_f}$$

C. Gaussian Process for Vehicle-Terrain Interaction

In this section, GP model is presented to infer the vehicle-terrain interactions. We depict the vehicle-terrain interaction as the modeling errors of vehicle velocities, i.e., $(v_x^{\mathcal{B}}, v_y^{\mathcal{B}}, \omega_z^{\mathcal{B}})$. A similar approach has been delivered in [13], which estimates the traversable cost in terms of modeling errors. However, they only utilized the kinematic model

assuming the flat surface. Let us denote the velocity error between true vehicle and the nominal system model in (1) at the discrete time step k as follows.

$$e_{vx}(k) = v_x^{\mathcal{B}}(k) - \hat{v}_x^{\mathcal{B}}(k)$$
$$e_{vy}(k) = v_y^{\mathcal{B}}(k) - \hat{v}_y^{\mathcal{B}}(k)$$
$$e_{\omega}(k) = \omega_z^{\mathcal{B}}(k) - \hat{\omega}_z^{\mathcal{B}}(k)$$

where $\hat{v}_x^{\mathcal{B}}(k)$, $\hat{v}_y^{\mathcal{B}}(k)$, and $\hat{\omega}_z^{\mathcal{B}}(k)$ are the predicted states propagated by (1) with $\xi(k-1)$, and $\zeta(k-1)$. Then the GP model is trained to represent the following vector-valued function.

$$g(x_{gp}(k)) = y(k) = [e_{vx}(k+1) \ \ e_{vy}(k+1) \ \ e_{\omega}(k+1)]^{\mathrm{T}}$$

where $x_{gp} = [v_x^{\mathcal{B}} \ v_Y^{\mathcal{B}} \ \omega_z^{\mathcal{B}} \ \phi \ \theta \ \delta \ a_x^{\mathcal{B}}]^{\mathrm{T}}$ and $y(k)$ are the model input and output measurement vectors, respectively. A GP for a function g is set to have zero prior mean and a squared exponential kernel function $\kappa : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ as follows [37].

$$\kappa(x_{gp}, x'_{gp}) = \sigma^2 exp\left(-\frac{(x_{gp}-x'_{gp})^{\mathrm{T}} L^{-2}(x_{gp}-x'_{gp})}{2}\right)$$

where σ, and L are the hyperparameters trained by maximizing the log marginal likelihood. Based on the data of measurements y corresponding to the inputs X, the predictions of GP at x_{gp}^* are represented by the predictive mean μ_* and variance σ_*^2 defined as:

$$\mu_* = K(x_{gp}^*, X)K_*^{-1}y$$
$$\sigma_*^2 = K(x_{gp}^*, x_{gp}^*) - K(x_{gp}^*, X)K_*^{-1}K(X, x_{gp}^*) \quad (2)$$

where $K_* = K(X, X)$, and $K(\cdot, \cdot)$ are matrices constructed using the kernel $\kappa(\cdot, \cdot)$ evaluated at the test and training data, x_{gp}^*, and X. Our proposed algorithm utilizes multiple number of GP models such that $g_j, j \in \{1, \cdots, J\}$ represents the vehicle-terrain interaction on J different types of terrains, e.g., grass, rock, and mud etc.

IV. ALGORITHM DEVELOPMENT

This section describes sub-modules of the proposed learning-based uncertainty-aware navigation algorithm in detail.

A. Algorithm Overview

The algorithm is decomposed into three main modules; 1) preprocessing 3D environment; 2) uncertainty-aware path planning; and 3) safe path following. The preprocessing step outputs the geometric traversability cost and terrain type map based on the RGB camera images and 3D Lidar sensor point clouds. The path planning module generates the predictive path distributions from the sampled control inputs using the learned vehicle-terrain interaction model. Then the generated predictive path distributions are evaluated to select the best path based on the uncertainty-aware traversability cost, the risk of rollover, and distance to the goal. The last module solves the optimal control problem to follow the resulting path using a 6 DOF vehicle model in (1). The overview of the algorithm is illustrated in Fig. 2.
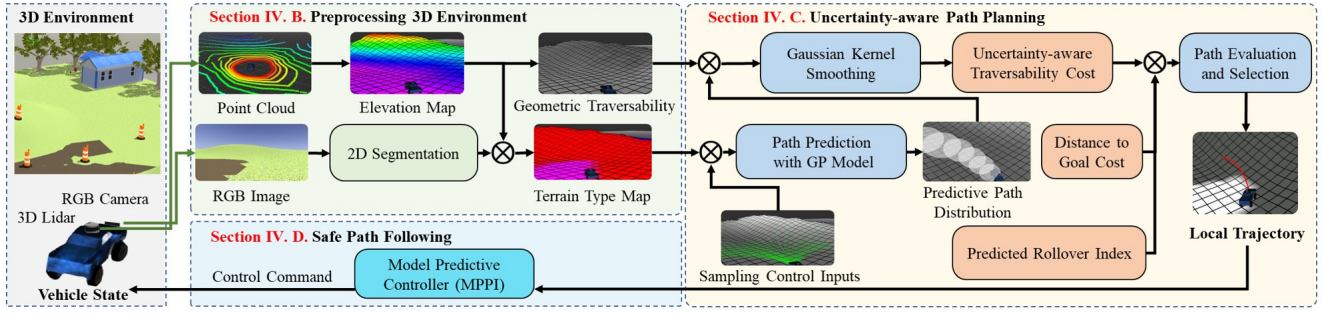
Fig. 2: Overview of the learning-based uncertainty-aware navigation in 3D terrain

B. Preprocessing 3D Environment

This module processes raw sensor measurements and generates global terrain grid maps. First, the elevation map developed by [38] is created using 3D point clouds from the LIDAR sensor. The created elevation map saves the height information in each grid. Then, the geometric traversability cost is computed for each grid as follows.

$$T_{geo} = w_1 T_s + w_2 T_r + w_3 T_h$$

where w_1, w_2, and w_3 are scale factors summing up to 1. And T_s, T_r, and T_h are the slop, roughness, and step height costs computed similar to [25]. We also set a robot-specific maximum threshold for traversability cost, so the total cost is bounded below the threshold.

Furthermore, terrain types are extracted from the RGB camera image with the help of the semantic segmentation module, i.e., [39], [40]. Each pixel is classified into different types of terrains from a predefined set of terrain categories. Then, the 2D pixels are projected to a 3D world coordinate frame [27]. Consequently, each grid in a elevation map can be labeled with types of terrain to form a terrain type map.

C. Uncertainty-aware Path Planning

In this module, the predictive path distributions corresponding to the individual control input samples are computed. Given the initial vehicle state $\xi(0)$, we sample I number of control inputs as $\zeta_i, i \in \{1, \cdots, I\}$. And we sample a collection of nominal state predictions (i.e., nominal path) with N time step prediction horizons, each matched to each control input.

$$\bar{\xi}_i(k), \ i \in \{1, \cdots, I\}, \ k \in \{0, \cdots, N\} \quad (3)$$

where each state is propagated by the nonlinear vehicle dynamics in (1).

$$\bar{\xi}_i(k+1) = f(\bar{\xi}_i(k), \zeta_i)$$

It is worth noting that one can analyze the traversability of the sampled nominal state predictions, $\bar{\xi}_i(k)$, by evaluating the costs from the created traversability maps that underlie the nominal path of the state predictions. However, this idea is prone to fail due to the uncertain vehicle-terrain interactions in the real world. Let us consider the vehicle that travels along the mud area near the obstacle, as shown in Fig. 3. There will be a higher chance that the vehicle collides

with the obstacle as the nominal path does not align with the actual path due to the slip or the deficient performance of the following control.
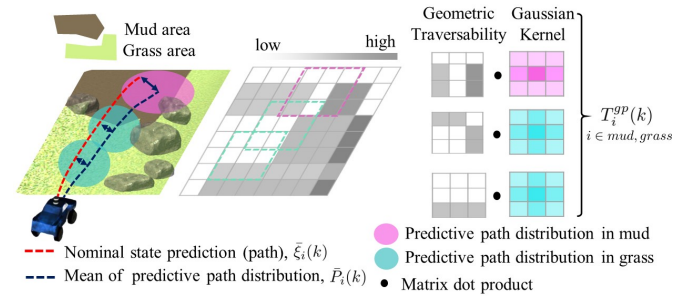

Fig. 3: Illustration of traversability cost computation for the given predictive distribution of vehicle poses at each step.

To overcome such limitations, our proposed algorithm learns the discrepancy between the nominal and actual path, and accommodates it in the planning phase. We first draw the predictive path distributions learned from the driving data. The GP model (2) is employed to generate M samples of actual state prediction (i.e., predictive path) for each control input.

$$\xi_i^{(m)}(k), \ i \in \{1, \cdots, I\}, \ k \in \{0, \cdots, N\}, \ m \in \{1, \cdots, M\}$$

where superscript m represents the predictive path sample index. This time, the state update can be derived by applying (1) and adding a sample output from the posterior GP distribution, represented by the following stochastic dynamical model.

$$\xi_i^{(m)}(k+1) = f(\xi_i^{(m)}(k), \zeta_i) + B \ g_j(F\xi_i^{(m)}(k), \psi, \theta, \zeta_i)$$

where $B = [0_3; I_3]$ and $F = [0_3 \ I_3]$. GP model, g_j, is selected based on the terrain type retrieved from the terrain type map at the vehicle position. Furthermore, we denote the position of each sample as $P_i^{(m)}(k) = [I_2 \ 0_4] \times \xi_i^{(m)}(k) \in \mathbb{R}^2$. Correspondingly, the mean and variance of predictive path distribution of vehicle can be computed as follows [16], [41].

$$\bar{P}_i(k) = \frac{\sum_{m=1}^{M} P_i^{(m)}(k)}{M}, \ k \in \{1, \cdots, N\}$$

$$\Sigma_i(k) = \frac{\sum_{m=1}^{M} \left(P_i^{(m)}(k) - \bar{P}_i(k)\right)^{\mathrm{T}} \left(P_i^{(m)}(k) - \bar{P}_i(k)\right)}{M - 1}$$

$$(4)$$

To compute the traversability cost over the predictive path distribution, we borrow the idea from 2D Gaussian kernel smoothing [17]. We construct a 2D Gaussian kernel matrix $\mathcal{K}_i(k) \in \mathbb{R}^{s \times s}$ for each position using $\bar{P}_i(k)$ and $\Sigma_i(k)$, $k \in \{1, \cdots, N\}$ where s is the predefined kernel size. And we denote $T_{geo}(\bar{P}_i(k)) \in \mathbb{R}^{s \times s}$ as the local submap of geometrical traversability cost map centered at $\bar{P}_i(k)$ (See Fig. 3). Then, one can define the predictive geometric traversability cost for the predictive path distribution with the mean $\bar{P}_i(k)$, and variance $\Sigma_i(k)$ as follows.

$$T_i^{gp}(k) := 1_s^{\mathrm{T}} \times \left(\mathcal{K}_i(k) \cdot T_{geo}(\bar{P}_i(k)) \right) \times 1_s$$

where 1_s is the column vector whose entries are all 1's. To prevent the actual path from excessively deviating from the nominal path, we further penalize the nominal path (3) for the errors of the corresponding predictive path distribution (4). Specifically, the following predictive uncertainty cost is defined by Mahalanobis distance [42].

$$e_i(k) := \sqrt{\left(B_e \bar{\xi}_i(k) - \bar{P}_i(k) \right)^{\mathrm{T}} \Sigma_i^{-1}(k) \left(B_e \bar{\xi}_i(k) - \bar{P}_i(k) \right)} \tag{5}$$

where $B_e = [I_2 \ 0_4] \in \mathbb{R}^{2 \times 6}$. Again, the cost in (5) is influenced by various factors, including vehicle-terrain interaction, the topology of 3D terrain, and even the capability of the following controller. To sum up, the uncertainty-aware traversability cost for each control input sample can be computed as follows.

$$T_i = \sum_{k=1}^{N} \left(w_{gp} T_i^{gp}(k) + w_e e_i(k) \right), \ i \in \{1, \cdots, I\}$$

where w_{gp} and w_e are the weighting parameters.

Apart from the uncertainty-aware traversability cost, the rollover risk is accounted for planning phase to assure safety during the agile motion of the vehicle. The Rollover index is used to express the vehicle status whether the vehicle reaches the critical rollover point [43], [44]. Given the vehicle states, the corresponding rollover index is computed by

$$R_{roll}(\xi) = \frac{2[(ma_y^{\mathcal{B}} + F_G)h_R - mgh_R \times s\psi)]}{L_w(F_{zf} + F_{zr})} \tag{6}$$

where $a_y^{\mathcal{B}} = \frac{C_{\alpha f}}{m}(\delta - \frac{v_y^{\mathcal{B}} + L_f \omega_z^{\mathcal{B}}}{v_x^{\mathcal{B}}}) + \frac{C_{\alpha r}}{m} \frac{L_r \omega_z^{\mathcal{B}} - v_y^{\mathcal{B}}}{v_x^{\mathcal{B}}}$ is the lateral acceleration in vehicle reference frame [45]. If any state predictions given the control inputs induce a rollover index above the safety threshold, i.e., $R_{roll}(\bar{\xi}_i^{(m)}(k)) > R_{thres}$, we exclude the corresponding path to ensure safety.

The final step of the planning module is selecting the best local path to guide a vehicle to the goal with the least traversability cost. For each nominal path generated by the ith sampled control input, one can compute the closest Euclidean distance from the sampled nominal path to the goal as T_i^{dist}. Then, the best local path is selected as follows.

$$\bar{\xi}_{i^*}(k), \ k \in \{0, \cdots, N\}, \ i^* = \operatorname*{arg\,min}_{\forall i \in \Omega_{Safe}} \left(T_i + w_{dist} \times T_i^{dist} \right)$$

where w_{dist} is the weighting parameter. And Ω_{Safe} is the index set of control inputs which do not violate rollover

condition (6) along the path.

$$\Omega_{Safe} := \{i \in \{1, \cdots, I\} | R_{roll}(\bar{\xi}_i^{(m)}(k)) \leq R_{thres},$$
$$k \in \{1, \cdots, N\}, m \in \{1, \cdots, M\}\}$$

D. Safe Path Following

Once the best path is selected, the path following algorithm is implemented based on a nonlinear MPC scheme. Given the current vehicle state and the local path $\bar{\xi}_{i^*}(k)$, $k \in \{0, \cdots, N\}$, a finite horizon optimization problem is formulated to obtain a sequence of control input that minimizes the cost function as follows.

$$\min_{\xi(k), \zeta(k)} \sum_{k=0}^{\bar{N}-1} \left\{ ||\xi(k) - \bar{\xi}_{i^*}(k)||_{\mathcal{Q}}^2 + ||\zeta(k)||_{\mathcal{R}}^2 \right\} \tag{7}$$
$$+ ||\xi(\bar{N}) - \bar{\xi}_{i^*}(\bar{N})||_{\mathcal{Q}}^2$$

where $\mathcal{Q}$ and $\mathcal{R}$ are the cost weight matrices. And $\bar{N}$ is the prediction horizon for the MPC. The solution for the optimization problem in (7) is denoted as $\zeta^*(k)$, $k \in \{0, \cdots, \bar{N}-1\}$ and the first sequence $\zeta^*(0)$ is applied to the vehicle. Since the dynamics are not twice continuously differentiable in the sharp elevation map, we utilize a sampling-based MPC method called model predictive path integral control (MPPI) [46], [47]. In short, this controller generates sample trajectories which are evaluated to find the minimum cost trajectory and the corresponding inputs. As sampling can be easily parallelized on the GPU, we can sample 5000 trajectories at about 10Hz in real-time.

V. EXPERIMENTS

A. Simulation Setup

A modified version of the Gazebo environment in the AutoRally research platform is used [48]. The vehicle is equipped with a front RGB camera and a 3D Lidar mounted on the vehicle. A 3D environment has hills and obstacles between the start and goal position, as illustrated in Fig. 4. There are two different types of terrains; i) the grass
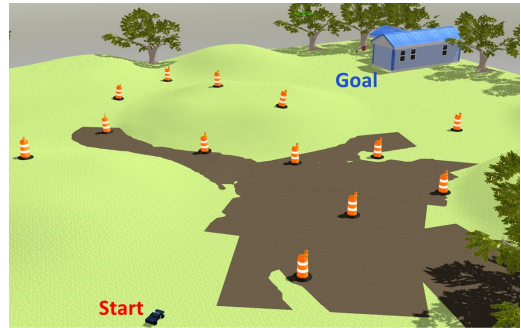

Fig. 4: Gazebo simulation environment with start and goal position.

area, which has higher friction coefficient; and ii) the mud area, which has lower friction coefficient than the grass area. Correspondingly, we trained two different GP models for each terrain using GPyTorch [49]. The training data was collected with a joystick-controlled vehicle in the same environment while driving on the both grass and mud area. Due to the slip that occurs in the mud area, the deviation

of the predictive path distribution from the nominal path is much higher in the mud area, as shown in Fig. 5. Therefore, we expect that much higher uncertainty costs, e_i, were assigned to the paths predicted over the mud area.
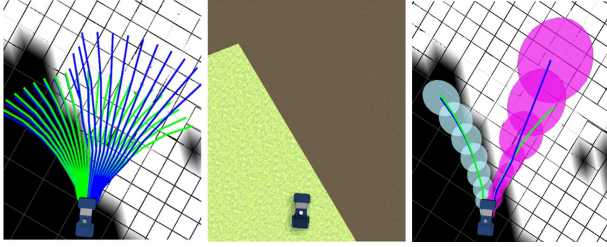

Fig. 5: Sampled nominal paths (green) and the mean of predictive path distributions (blue) are drawn on the left side. The predictive path distributions over the mud area (purple ellipsoid) and the grass area (cyan ellipsoid) are represented on the right side.

Instead of employing the 2D segmentation module like in [27], we assumed each pixel were classified into two labels (i.e., grass and mud) depending on the color where the RGB color boundary was manually set. The elevation, terrain type, and geometric traversability maps have a grid size of $0.5m$ and an update rate of $10Hz$. The algorithm and the simulation is processed by a desktop computer with Intel i9-9900k CPU, 64GB RAM, and Nvidia RTX 2070 GPU. For uncertainty-aware path planning step, we set $I = 63$, $M = 20$, $R_{thres} = 0.3$, and the Gaussian kernel size as $s = 5$. Also, the discrete system has a sample time of $0.1sec$, and the prediction Horizon is $3sec$ resulting in $N = 30$. Furthermore, during each MPPI path following process, we use sample time as $0.1sec$, and the prediction horizon as $2sec$. Local path was published at $3Hz$ to the MPPI controller.

B. Comparative Analysis

To validate the performance of the proposed algorithm, we conduct a comparative analysis with two baseline navigation methods: i) a method that only considers the geometric traversability cost [25]; and ii) a method that considers the hybrid traversability cost, which accounts for both semantic and geometric traversability costs by simply adding two different costs [27]. For real-time path planning of the baseline navigation methods, we choose A^* as the global path planner and Hybrid A^* as the local path planner since it has been proven to be successfully applied for many off-road vehicle navigation [29], [34], [50], [51]. Pure pursuit controller is applied for path following of baseline navigation methods [52].

Each algorithm was tested 10 times with different locations of obstacles in the same environment setting of 3D terrain topology and terrain type. The simulation result of the example scenario is shown in Fig. 6. The first baseline navigation method generated the red path driving over the large mud area as it did not account for semantic features at all. The vehicle started to spin due to a high slip in the mud area and could not reach the goal in this run. The second baseline navigation method exhibited the blue
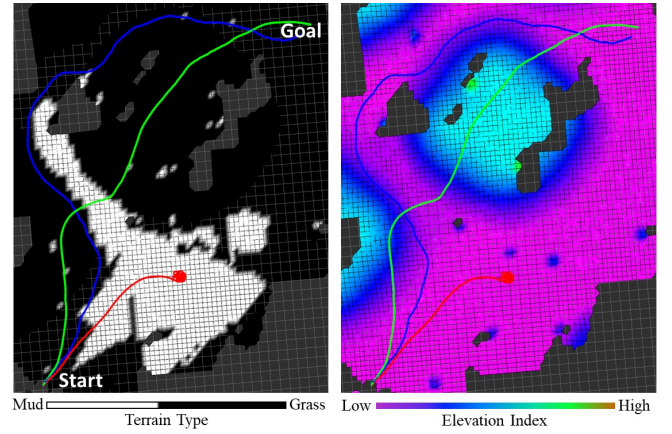

Fig. 6: Comparison of paths driven by the proposed algorithm (green), baseline1 method with geometric traversability cost only (red), baseline2 method with hybrid traversability cost (blue). Paths are illustrated on top of the processed terrain type map (left) and elevation map (right). Gray areas in both maps are undiscovered areas during the mission.

path, which detoured the mud area and resulted in a too conservative longer path. Lastly, our method sometimes avoided or drove over the mud area to make safe but agile navigation. Remarkably, the proposed algorithm guided the vehicle to cross the mud area as it guarantees vehicle safety based on the predictive path distribution. This, in turn, results in a much shorter path than the others.

The overall simulation statistics are shown in Table I. The qualities of paths are characterized by the average path length of successful runs and the number of successful runs, i.e., no collision and rollover. The path generated by the proposed method is shorter than the baselines while it successfully navigated to the goal point.

TABLE I: The overall simulation results

METHODS	AVG.PATH LENGTH	SUCCESSFUL RUNS (OUT OF 10 TRIALS)
BASELINE1 (RED)	61.2	4
BASELINE2 (BLUE)	65.7	8
PROPOSED (GREEN)	52.6	10

VI. CONCLUSIONS

This work proposed a novel uncertainty-aware navigation algorithm to cope with 3D terrain topology and varying terrain conditions. Our approach leverages a learning-based framework that incorporates the terrain-induced uncertainty into the traversability cost and further preserves the real-timeness based on sampling-based methods. We validated the proposed algorithm in a 3D simulation environment with different types of terrains and showcased safe and agile navigation performance. In the future work, we will test our method on a physical vehicle in real-world off-road environments. For further improvement in the algorithmic aspect, we also plan to extend our work by incorporating an uncertainty-aware global path planner to avoid the local minima problem.

REFERENCES

[1] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 4, pp. 1373–1385, 2013.

[2] H. Mousazadeh, "A technical review on navigation systems of agricultural autonomous off-road vehicles," *Journal of Terramechanics*, vol. 50, no. 3, pp. 211–232, 2013.

[3] G. Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, G. Pereira, and J. Spletzer, "Distributed search and rescue with robot and sensor teams," in *Field and Service Robotics*. Springer, 2003, pp. 529–538.

[4] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open Journal of Intelligent Transportation Systems*, 2022.

[5] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[6] S. Yu, C. Shen, and T. Ersal, "Nonlinear model predictive planning and control for high-speed autonomous vehicles on 3d terrains," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 412–417, 2021.

[7] C. Shen, S. Yu, and T. Ersal, "A three-phase framework for global path planning for nonholonomic autonomous vehicles on 3d terrains," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 160–165, 2021.

[8] P. Krüsi, P. Furgale, M. Bosse, and R. Siegwart, "Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments," *Journal of Field Robotics*, vol. 34, no. 5, pp. 940–984, 2017.

[9] J. Hu, Y. Hu, C. Lu, J. Gong, and H. Chen, "Integrated path planning for unmanned differential steering vehicles in off-road environment with 3d terrains and obstacles," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[10] M. Mizuno and T. Kubota, "A new path planning architecture to consider motion uncertainty in natural environment," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2182–2188.

[11] 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.

[12] X. Cai, M. Everett, J. Fink, and J. P. How, "Risk-aware off-road navigation via a learned speed distribution map," *arXiv preprint arXiv:2203.13429*, 2022.

[13] M. V. Gasparino, A. N. Sivakumar, Y. Liu, A. E. Velasquez, V. A. Higuti, J. Rogers, H. Tran, and G. Chowdhary, "Wayfast: Navigation with predictive traversability in the field," *IEEE Robotics and Automation Letters*, 2022.

[14] T. Manderson, S. Wapnick, D. Meger, and G. Dudek, "Learning to drive off road on smooth terrain in unstructured environments using an on-board camera and sparse aerial images," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1263–1269.

[15] J. M. Gregory, G. Warnell, J. Fink, and S. K. Gupta, "Improving trajectory tracking accuracy for faster and safer autonomous navigation of ground vehicles in off-road settings," in *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2021, pp. 204–209.

[16] A. Capone and S. Hirche, "Anticipating the long-term effect of online learning in control," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 3865–3872.

[17] M. Nixon and A. Aguado, *Feature extraction and image processing for computer vision*. Academic press, 2019.

[18] C. Wang, J. Wang, C. Li, D. Ho, J. Cheng, T. Yan, L. Meng, and M. Q.-H. Meng, "Safe and robust mobile robot navigation in uneven indoor environments," *Sensors*, vol. 19, no. 13, p. 2993, 2019.

[19] Z. Jian, Z. Lu, X. Zhou, B. Lan, A. Xiao, X. Wang, and B. Liang, "Putn: A plane-fitting based uneven terrain navigation framework," *arXiv preprint arXiv:2203.04541*, 2022.

[20] H. Rastgoftar, B. Zhang, and E. M. Atkins, "A data-driven approach for autonomous motion planning and control in off-road driving scenarios," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 5876–5883.

[21] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "A nonlinear model predictive control formulation for obstacle avoidance in high-speed autonomous ground vehicles in unstructured environments," *Vehicle system dynamics*, vol. 56, no. 6, pp. 853–882, 2018.

[22] H. Febbo, J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "Moving obstacle avoidance for large, high-speed autonomous ground vehicles," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 5568–5573.

[23] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "Combined speed and steering control in high-speed autonomous ground vehicles for obstacle avoidance using model predictive control," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8746–8763, 2017.

[24] 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.

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

[26] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3019–3026, 2018.

[27] T. H. Y. Leung, D. Ignatyev, and A. Zolotas, "Hybrid terrain traversability analysis in off-road environments," in *2022 8th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE, 2022, pp. 50–56.

[28] Y. Zhao, P. Liu, W. Xue, R. Miao, Z. Gong, and R. Ying, "Semantic probabilistic traversable map generation for robot path planning," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 2576–2582.

[29] M. Thoresen, N. H. Nielsen, K. Mathiassen, and K. Y. Pettersen, "Path planning for ugvs based on traversability hybrid astar," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1216–1223, 2021.

[30] T. Overbye and S. Saripalli, "Fast local planning and mapping in unknown off-road terrain," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5912–5918.

[31] L. Dabbiru, S. Sharma, C. Goodin, S. Ozier, C. Hudson, D. Carruth, M. Doude, G. Mason, and J. Ball, "Traversability mapping in off-road environment using semantic segmentation," in *Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2021*, vol. 11748. SPIE, 2021, pp. 78–83.

[32] L. Nardi and C. Stachniss, "Actively improving robot navigation on different terrains using gaussian process mixture models," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4104–4110.

[33] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Field and Service Robotics*. Springer, 2018, pp. 335–350.

[34] T. Guan, Z. He, R. Song, D. Manocha, and L. Zhang, "Tns: Terrain traversability mapping and navigation system for autonomous excavators," *Robotics: Science and Systems*, 2022.

[35] G. G. Waibel, T. Löw, M. Nass, D. Howard, T. Bandyopadhyay, and P. V. K. Borges, "How rough is the path? terrain traversability estimation for local and global path planning," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[36] M. Doumiati, A. Victorino, A. Charara, and D. Lechner, "Lateral load transfer and normal forces estimation for vehicle safety: experimental test," *Vehicle System Dynamics*, vol. 47, no. 12, pp. 1511–1533, 2009.

[37] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2.

[38] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *Mobile Service Robotics*. World Scientific, 2014, pp. 433–440.

[39] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2017.

[40] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-scnn: Gated shape cnns for semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5229–5238.

[41] T. Brüdigam, A. Capone, S. Hirche, D. Wollherr, and M. Leibold, "Gaussian process-based stochastic model predictive control for overtaking in autonomous racing," *arXiv preprint arXiv:2105.12236*, 2021.

[42] P. C. Mahalanobis, *On the generalized distance in statistics*. National Institute of Science of India, 1936.

[43] X. Qian, C. Wang, and W. Zhao, "Rollover prevention and path following control of integrated steering and braking systems," *Proceedings of*

the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, vol. 234, no. 6, pp. 1644–1659, 2020.

[44] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, pp. 1–14, 2014.

[45] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[46] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017.

[47] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1714–1721.

[48] B. Goldfain, P. Drews, C. You, M. Barulic, O. Velev, P. Tsiotras, and J. M. Rehg, "Autorally: An open platform for aggressive autonomous driving," *IEEE Control Systems Magazine*, vol. 39, no. 1, pp. 26–55, 2019.

[49] J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson, "Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration," *Advances in neural information processing systems*, vol. 31, 2018.

[50] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.

[51] J. Petereit, T. Emter, C. W. Frey, T. Kopfstedt, and A. Beutel, "Application of hybrid astar 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.

[52] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, Tech. Rep., 1992.