

Risk-Aware Off-Road Navigation via a Learned Speed Distribution Map

Xiaoyi Cai[†], Michael Everett[†], Jonathan Fink[‡], and Jonathan P. How[†]

Abstract—Motion planning in off-road environments requires reasoning about both the geometry and semantics of the scene (e.g., a robot may be able to drive through soft bushes but not a fallen log). In many recent works, the world is classified into a finite number of semantic categories that often are not sufficient to capture the ability (i.e., the speed) with which a robot can traverse off-road terrain. Instead, this work proposes a new representation of traversability based exclusively on robot speed that can be learned from data, offers interpretability and intuitive tuning, and can be easily integrated with a variety of planning paradigms in the form of a costmap. Specifically, given a dataset of experienced trajectories, the proposed algorithm learns to predict a distribution of speeds the robot could achieve, conditioned on the environment semantics and commanded speed. The learned speed distribution map is converted into costmaps with a risk-aware cost term based on conditional value at risk (CVaR). Numerical simulations demonstrate that the proposed risk-aware planning algorithm leads to faster average time-to-goals compared to a method that only considers expected behavior, and the planner can be tuned for slightly slower, but less variable behavior. Furthermore, the approach is integrated into a full autonomy stack and demonstrated in a high-fidelity Unity environment and is shown to provide a 30% improvement in the success rate of navigation.

I. INTRODUCTION

Autonomous robotic navigation in off-road environments is important for many tasks, including planetary exploration [1], [2] and search-and-rescue missions [3]. In an off-road setting, a fundamental challenge in the local planning problem is that both the *geometry* (e.g., positive and negative obstacles) and the *semantics* (e.g., terrain type, time of year) impact the speed and safety with which a robot can traverse the environment. For example, Fig. 1 shows a mobile robot surrounded by grass and foliage. While a purely geometry-based system might consider much of the scene as obstacles (e.g., since a lidar pointcloud would have many returns from the leaves and grass), the semantics suggest much of the vegetation is sufficiently soft that the vehicle could drive through it, but trees and rocks would likely stop the robot.

To capture these semantics, many approaches train semantic segmentation modules for camera images [4], [5] or lidar pointclouds [6], which can reduce the dimensionality and enable learning in a low visual fidelity simulation environment [7], [8]. However, the ontology of existing labeled datasets for off-road navigation [4], [9]–[11] is a fundamental limitation for capturing traversability. For instance, the 20 and 24 classes in [9], [10] only contain the broad “bush”,



Fig. 1: Off-Road Navigation. Even though much of this scene can be categorized broadly as grass/foliage, the vehicle’s ability to traverse through different regions can vary dramatically. This paper proposes a planner that accounts for such variability within semantic terrain/object classes, by learning a speed distribution map.

“grass”, and “tree” classes for vegetation, where the varying degree of traversability within each class is not captured. For instance, some bushes can be driven through, some will slow the robot down, and others will stop the robot. Alternatively, [11] contains labels with finer granularity such as “traversable grass” and “non-traversable low vegetation”, but these are specific to the large vehicle in mind during the expensive manual labeling procedure, and thus would not generalize to other vehicles.

Furthermore, because each individual mission or operator can have different objectives, it is important that the risk tolerance of the planner (e.g., whether to take a shortcut that has a 1% chance of causing the robot to get stuck) can be quickly adjusted. Risk-aware planning has been extensively studied, e.g., [12]–[16], with different notions of risk such as collision probability and classification uncertainty. Recent works such as [17], [18] have adopted conditional value at risk (CVaR) as the risk metric, which has been analyzed in-depth in [19] about why CVaR allows robots to assess risks rationally. Alternatively, other work tries to directly learn policies, dynamics models, or cost functions [20]–[26] that satisfy the desired risk tolerance. However, these off-road techniques often require a non-intuitive cost function tuning procedure or a complete re-training of the model to adjust the risk tolerance.

To address these issues and bridge the gap between semantic perception and risk-aware planning, this work proposes a new representation of traversability as a general distribution of robot speed conditioned on environment semantics and the commanded speed. The proposed pipeline first automatically labels a dataset of collected trajectories with realized vehicle

[†]Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. {xyc, mfe, jhow}@mit.edu.

[‡]U.S. Army Combat Capabilities Development Command Army Research Laboratory, Adelphi, MD 20783, USA. jonathan.r.fink3.civ@army.mil.

speeds to capture the variability in speed outcomes associated with each semantic class. This dataset can then be used to learn a distribution of achievable speeds, conditioned on the commanded speed and semantics of the nearby terrain. The learned speed distribution is then converted into a *speed map* representation that can be leveraged with various planning paradigms. Notably, we incorporate risk-awareness into the planner via CVaR and show how to adjust the risk without collecting any extra data or re-training the learned model.

The contributions of this work include: i) a new representation of traversability as a probability distribution of speeds the robot could achieve, which can be learned from data and allows physically meaningful interpretation in m/s; ii) a new risk-aware minimum-time planner based on Model Predictive Path Integral (MPPI [27]) control that uses the learned speed distribution map, allowing risk level adjustment without re-training or collecting more trajectories; iii) a demonstration of a robot reaching its goal with up to 30% improvement in success rate than a risk-unaware algorithm in a high-fidelity Unity simulation environment in a full autonomy stack.

II. RELATED WORK

Traversability analysis can be achieved via both proprioceptive and exteroceptive [28] sensors, where the former category includes IMUs that measure vibration and orientation of the robot [29], [30], and the latter category that includes lidar and RGB cameras that provide geometric and semantic understanding of the environment. Purely geometry-based analysis has been widely adopted, e.g., [18], [31]–[33], which often involves a weighted sum of costs extracted based on geometric properties such as slope, roughness and step height. Notably, geometry-based methodology has been demonstrated successfully in the DARPA Subterranean Challenge [18], where the cost function is assumed to be Gaussian, allowing easy computation and adjustment of risk level via CVaR. In contrast, this work proposes a new representation of traversability based on experience and brings semantics into the problem.

Among methods that combine semantic-based and geometry-based techniques, [34] proposes a fusion strategy that uses geometry-based cost (slope added to step height) for the terrain, unless the associated semantic label is known to be undesirable. [35] uses both geometric and semantic layers in a multi-layer costmap, and fuses the costs by accounting for layer uncertainty. [6] classifies a dense 3D pointcloud to extract traversability labels (Free, Low Cost, Medium Cost, Lethal), where the ground truth labels are designed based on human expertise. These methods either require human expertise in associating semantics with traversability, or require combining costs with different units, which makes tuning non-intuitive. In contrast, this work uses vehicle speed as a common unit in the cost function to enable intuitive risk level adjustment.

Other recent work proposes methods to learn navigation policies or cost functions from experience via imitation learning [20]–[23], inverse reinforcement learning [24], semi-supervised learning [36], or model-free reinforcement learn-

ing [37]. While these methods leverage datasets or simulators to reduce some of the expert knowledge requirements, a key limitation is that adjusting the risk tolerance could require collection of a new set of expert trajectories and/or re-training the learned models. Alternatively, [25], [26] learn a predictive “events” (e.g., bumpy, collision, smooth) model from a diverse dataset of experiences. By predicting the probability of undesirable events, the riskiness of the planner in [25] can be adjusted by changing the penalty for these events without re-training the network. However, the cost terms in [25] have different units such as terrain bumpiness and goal proximity, leading to a difficult conversion from risk tolerance to reward function weights. This work also leverages learning from a dataset of experienced trajectories, but instead proposes a pipeline to produce speed maps that can be incorporated into many planning paradigms.

III. TRAVERSABILITY

For fast off-road navigation, time-to-goal is a typical performance measure that depends on the quality of planned trajectory and how well a robot executes the planned maneuvers. A good assessment of terrain traversability allows a planner to generate trajectories that are both fast and can be executed. To this end, the discrepancy between the vehicle’s *planned* speed and *realized* speed provides a natural quantity to describe the *traversability* of terrain. Importantly, traversability is probabilistic in nature, due to imperfect sensing, broad semantic class labels for terrain, and the dynamics of vehicle-terrain interactions. Therefore, we propose to capture traversability via a conditional distribution of realized speed given the commanded speed and sensor observation.

A. Traversability as a Conditional Speed Distribution

Denoting the set of realized speeds as S and the set of possible observations about a terrain patch as O , we define traversability of the terrain as the conditional distribution

$$p_\theta(s | s^{\text{cmd}}, o) : S \mid S \times O \rightarrow \mathbb{R}, \quad (1)$$

where $s, s^{\text{cmd}} \in S$ are the realized speed and the commanded speed, $o \in O$ is the observation about the terrain, and p_θ is a probability distribution parameterized by θ , which in practice can be learned via a neural network. This representation is general enough to capture the multi-modality of the distribution and allows the planner to extract desired statistics for trajectory planning such as mean, modes and variance.

For high-speed navigation in a cluttered environment (e.g., a forest), the planner often has to trade off opportunities to reduce navigation time with the risk of colliding with an obstacle or getting stuck. To quantify these risks, we adopt the Conditional Value at Risk (CVaR), which satisfies a group of axioms important for rational risk assessment [19]. The Conditional Value at Risk at level $\alpha \in [0, 1]$ is defined as:

$$\text{CVaR}_\alpha(S) := \frac{1}{\alpha} \int_0^\alpha \text{VaR}_\tau(S) d\tau, \quad (2)$$

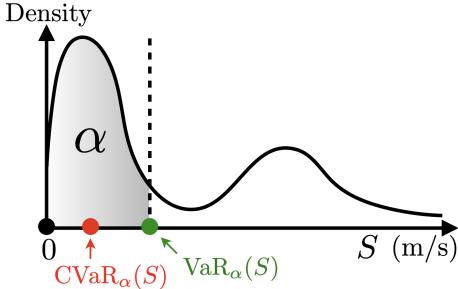


Fig. 2: This work defines conditional value at risk (CVaR) to capture worst case speed (i.e., near 0), rather than other common definitions based on the (right-side) tail.

where $\text{VaR}_\alpha(S)$ is the Value at Risk, or the α -quantile:

$$\text{VaR}_\alpha(S) := \max\{s \mid p(S < s) \leq \alpha\}. \quad (3)$$

Note that we define CVaR to capture the *worst-case speed* outcomes (i.e., lowest speed), as visualized in Fig. 2. Intuitively, $\text{CVaR}_\alpha(S)$ measures the average speed outcomes that are lower than the α -quantile of the speed distribution, capturing the worst-case expected speed. Notice that CVaR is the same as the mean when $\alpha = 1$, so often a low α is picked for sufficient distinction from the expectation.

B. Generating a Risk-Aware Traversability Map

Next, we show how a learned speed distribution from Eq. (1) can be used to convert a semantic map (built from the robot's sensor data) into a representation of traversability for the planner.

The architecture is illustrated in Fig. 3. The input to the pipeline is a semantic gridmap, $\mathbf{M}_s \in \mathbb{R}^{C \times H \times W}$, with C semantic classes, width W , and height H . Let $\mathbf{M} \in \mathbb{R}^{K \times H \times W}$ represent a K -layer speed map, where each layer has width W and height H . For map \mathbf{M} , we denote $m_{k,h,w}$ as the cell value in layer k , row h and column w . Given the speed limits of $[0, s^{\max}]$, we let the k -th layer correspond to the commanded speed range of $[\frac{(k-1)}{K}s^{\max}, \frac{k}{K}s^{\max}]$, where $k \in \{1, \dots, K\}$. Lastly, we associate the distribution $p_\theta(s \mid s^{\text{cmd}} = \frac{(k-0.5)}{K}s^{\max}, o_{h,w})$ to each cell indexed by (k, h, w) , where $o_{h,w}$ denotes the observation about the terrain patch that lies in the cell. Note that any cell with unknown traversability should be marked, e.g., with a negative number.

Although it is up to the user to populate the map cell values based on the associated speed distribution, it is important that the values provide interpretability and allow the user to easily adjust the riskiness of the planner. To this end, we propose to use the convex combination of the mean and $\text{CVaR}_\alpha(\cdot)$ at level α of the underlying speed distribution:

$$m_{k,h,w} = \beta \cdot s_{k,h,w}^{\text{risk}} + (1 - \beta) \cdot s_{k,h,w}^{\text{mean}}, \quad \beta \in [0, 1], \quad (4)$$

where we know $0 \leq s_{k,h,w}^{\text{risk}} \leq s_{k,h,w}^{\text{mean}} \leq s^{\max}$. Intuitively, $m_{k,h,w}$ can be interpreted as the *risk-adjusted speed* which lies between the CVaR speed estimate and the mean speed estimate due to convex combination.

For convenience, we define a look-up function $\mathcal{S}(\mathbf{M}, \mathbf{p}, s)$ that returns the risk-adjusted speed estimate in the multi-layer

speed map \mathbf{M} given a position $\mathbf{p} \in \mathbb{R}^2$ and speed $s \in \mathbb{R}$:

$$\mathcal{S}(\mathbf{M}, \mathbf{p}, s) = \begin{cases} 0 & \text{if out-of-map or unknown,} \\ m_{k,h,w} & \text{otherwise,} \end{cases} \quad (5)$$

where (k, h, w) are the map indices corresponding to position \mathbf{p} and speed s if \mathbf{p} lies in the map. Note that 0 speed is returned when the look-up is not valid, which is desirable when the unknown region can be dangerous, such as water. However, if domain knowledge suggests that the unknown is benign, a more optimistic value can be returned such as the query speed s .

IV. MINIMUM TIME NAVIGATION

We adopt the MPPI controller proposed in [27, Algorithm 2] for minimum time planning. MPPI is an information theoretic model predictive control (MPC) algorithm that tries to approximate the mean of the optimal control distribution via weighted samples of a Gaussian proposal distribution in order to minimize the KL-divergence between the two. This approach is attractive because it is derivative-free and works with general cost functions and dynamics. Next, we follow the notation in [27] and propose a new cost function for min-time planning using the proposed risk-aware traversability map. Fig. 4 illustrates a high-level overview of our strategy and the notation used.

Consider the discrete time stochastic system:

$$\mathbf{x}_{t+1} = F(\mathbf{x}_t, \mathbf{v}_t), \quad (6)$$

where $\mathbf{x}_t \in \mathbb{R}^n$ is the state vector and $\mathbf{v}_t \in \mathbb{R}^m \sim \mathcal{N}(\mathbf{u}_t, \boldsymbol{\Sigma})$ is the noisy realization of the nominal control input $\mathbf{u}_t \in \mathbb{R}^m$. Given the initial condition \mathbf{x}_0 , a sequence of input $\mathbf{v}_{0:T-1}$ leads to the state trajectory $\mathbf{x}_{0:T}$ according to the dynamics (6). For the purpose of min-time planning, we assume it is possible to extract the planar position $\mathbf{p}_t \in \mathbb{R}^2$ and speed $s_t \in \mathbb{R}$ from state \mathbf{x}_t . Furthermore, let $\mathbb{1}^{\text{done}}(\mathbf{x}_{0:t})$ be an indicator function that returns 1 when any state \mathbf{x}_τ has reached the goal at position \mathbf{p}^{goal} for $0 \leq \tau \leq T$, and returns 0 otherwise.

The min-time state-dependent objective is defined as

$$C(\mathbf{x}_{0:T}) = \phi(\mathbf{x}_T) + \sum_{t=0}^{T-1} q(\mathbf{x}_t), \quad (7)$$

where $\phi(\mathbf{x}_T)$ and $q(\mathbf{x}_t)$ are the time-to-go and the stage cost, respectively:

$$\phi(\mathbf{x}_T) = \frac{\|\mathbf{p}^{\text{goal}} - \mathbf{p}_T\|}{s_{\text{default}}} (1 - \mathbb{1}^{\text{done}}(\mathbf{x}_{0:T})) \quad (8)$$

$$q(\mathbf{x}_t) = \frac{s_t \cdot \Delta}{\mathcal{S}(\mathbf{M}, \mathbf{p}_t, s_t)} (1 - \mathbb{1}^{\text{done}}(\mathbf{x}_{0:t})), \quad (9)$$

with s_{default} being the default speed for estimating time-to-go at the end of the rollout, and Δ being the sampling duration. Intuitively, the lower the risk-adjusted speed $\mathcal{S}(\mathbf{M}, \mathbf{p}_t, s_t)$ is, the more nominal stage cost Δ is scaled up, indicating longer time to travel. If any state \mathbf{x}_τ reaches the goal for $0 \leq \tau \leq T$, all subsequent states do not incur stage cost or cost-to-go.

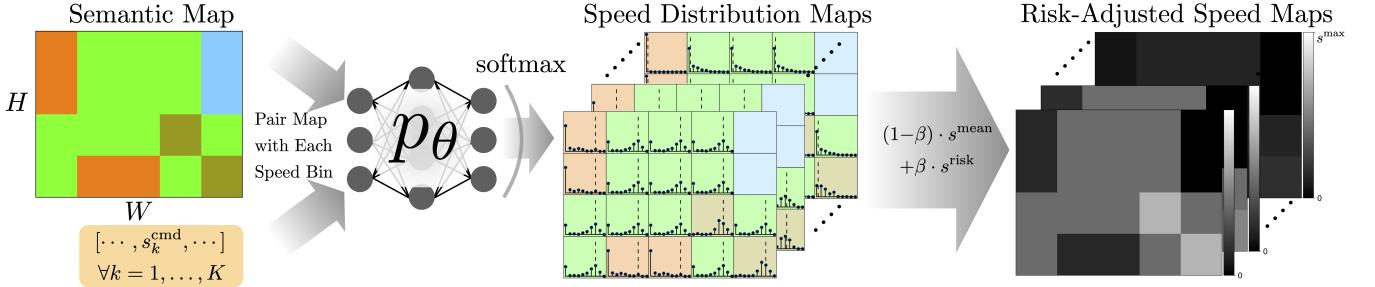


Fig. 3: Low-dimensional visualization of the pipeline to generate risk-aware speed map from semantic input and desired commanded speeds. A 10-layer 100×100 map with 10 output speed bins can be efficiently evaluated on an 8-core CPU under 2 ms with a simple neural network with two 64-node hidden layers. The speed distribution maps allow any planner to easily adjust the risk tolerance via relative weight β or via the α level of CVaR directly.

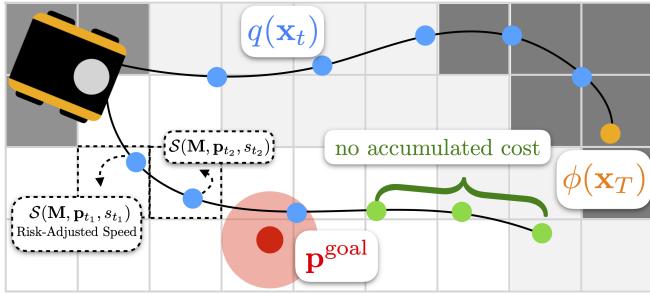


Fig. 4: Diagram showing how each MPPI rollout is assigned cost in order to minimize time-to-goal, where the underlying colored grid is the risk-adjusted speed map. The cost (7) consists of the stage cost $q(\mathbf{x}_t)$ (blue) for $0 \leq t \leq T - 1$ and the estimated time-to-goal $\phi(\mathbf{x}_T)$ (orange). The core idea is that the nominal traversal time at each step t will be adjusted based on the risk-adjusted speed estimate $S(\mathbf{M}, \mathbf{p}_t, s_t)$, where s is speed map, \mathbf{p}_t and s_t are the position and speed extracted from state \mathbf{x}_t . Note that if any state on a rollout satisfies the goal tolerance (red), the subsequent states do not accumulate stage cost or the terminal cost (shown in green).

At each time t and given the nominal control sequence $\mathbf{u}_{t:t+T}$, MPPI estimates the mean of the optimal control distribution as the weighted sum of control rollouts that are sampled from $\mathcal{N}(\mathbf{u}_\tau, \Sigma)$ for all $t \leq \tau \leq t+T$. The weight of each rollout is an exponentiated cost function (7) evaluated along the induced state trajectory. The algorithm runs in an receding horizon fashion, where the nominal control sequence $\mathbf{u}_{t+1:t+T+1}$ in the next round is set to be newest estimate of the mean of the optimal control distribution.

V. RESULTS

In an off-road environment, where the geometric and semantic properties are hard to assess, traversability can be highly non-Gaussian. For instance, if terrain is classified as vegetation but there is no distinction between dense bushes or soft grass, the speed outcome may exhibit bimodal distribution (i.e., being stuck or not). With this core issue in mind, we first validate our approach of capturing speed distributions in a grid world (Section V-A), and then integrate the proposed planner into a full autonomy stack in a high-fidelity Unity environment (Section V-B). Given this environment and autonomy stack, Section V-C describes the process of collecting a training dataset and training a neural network to predict PMFs of the speed distribution. Then, Section V-D demonstrates the improved navigation

performance as a result of the learned speed maps and risk-awareness.

A. Grid World Navigation

To validate that a speed distribution representation of traversability can be incorporated into a risk-aware planner and lead to improved performance, we first designed a grid world (see upper left of Fig. 5), where each $1 \text{ m} \times 1 \text{ m}$ cell is associated with a semantic type (either vegetation or dirt). The task is to navigate from the start position to the goal position in minimum time by planning a sequence of actions chosen from $\{\text{UP}, \text{DOWN}, \text{LEFT}, \text{RIGHT}\}$ that move the robot to a neighboring cell in the corresponding directions with nominal speed of 1 m/s. Although every action is deterministic, the actual traversal time is stochastic due to the underlying speed distribution of traversed cell, as shown in Fig. 5 (upper right). Note that we use a single-layer traversability map (i.e., $K = 1$) and denote $m_{h,w}$ for each value.

To find the min-time trajectory, we use a best-first search algorithm [38] with a prioritized search queue where new nodes with reachable states given actions are added. The stage cost of the map cell at row h and column w is the estimated traversal time $1/m_{h,w}$, where CVaR is fixed at level $\alpha = 0.1$. We compare the performance of the planner with $\beta \in \{0, 0.5, 1\}$, which correspond to using mean speed, risk-adjusted speed, and the (pure) risk speed. The optimal trajectories and their average time-to-goal values over 1000 trials are visualized in Fig. 5 (upper left and bottom). By only capturing the mean speeds ($\beta = 0$), the planner's performance has high variance because it does not consider the worst-case. By only accounting for risk ($\beta = 1$), the planner's performance has very low variance, but it is also highly conservative. By utilizing both risk and mean of the speed distribution ($\beta \in (0, 1)$), the planner's riskiness can be chosen to achieve shorter average time-to-goal with slight increase in variance. This experiment demonstrates that representing traversability as a speed distribution can be incorporated into a risk-aware planner and lead to improved performance.

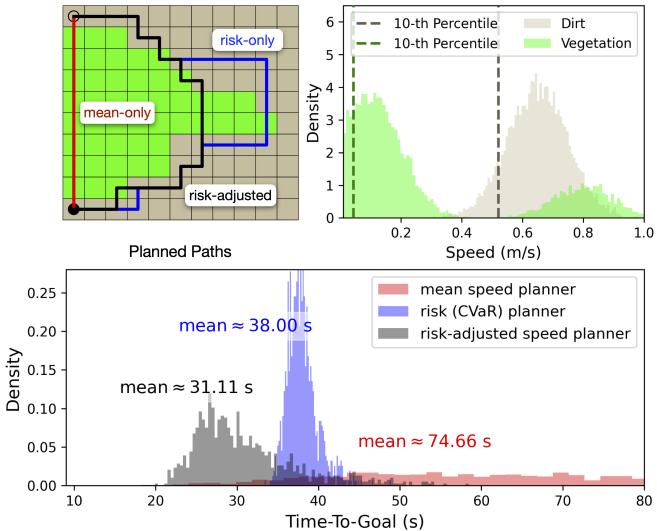


Fig. 5: A grid world example (top left) where the green *vegetation* and brown *dirt* terrain types have distinct speed distributions (top right) given the fixed commanded speed. A min-time planner that ignores risk (red) has the worst time-to-goal with high variance. On the other hand, the planner that only uses CVaR speed estimates acts conservatively by avoiding most of the vegetation cells, which leads to low-variance performance but still slow in time-to-goal. By combining both the risk and mean speeds, our proposed planner (gray) gains better average performance by taking moderate risks.

B. Integration with Autonomy Stack in High-Fidelity Environment

Next, we integrate the proposed methods into a full autonomy stack [39] in a high-fidelity Unity simulation environment with a Clearpath Warthog platform [40], as shown in Fig. 6. To focus on the challenge of coarse semantic labeling and its impact on traversability analysis, the Unity environment contains *dirt* and *vegetation* terrain types, where roughly $\frac{1}{4}$ of the bushes (classified as *vegetation*) on grass are non-traversable (i.e., the robot cannot drive through them), which slows down navigation and the resulting wheel slip has a side-effect of causing substantial drift in the vehicle’s pose estimate.

C. Data Collection and Network Training

In order to learn the traversability model as defined in (1), samples consisting of tuples of (*commanded speed*, *terrain type*, *realized speed*) were gathered with a joystick-controlled robot, where the joystick provided the commanded speeds, and the traversed terrain type and true speed were taken directly from the Unity simulation engine. The robot drove for 3 minutes in the training area (Fig. 6 left), resulting in about 7000 and 2000 samples associated with the vegetation and dirt terrain, respectively. The dataset was used to train a multi-layer feedforward NN with 2 hidden layers of 64 nodes and ReLU activation functions. The network input contains the one-hot encoding of the terrain type and the commanded speed, and the output consists of the probability mass function (PMF) for 10 output speed bins between 0 m/s and maximum speed 5 m/s via a softmax layer. The network was trained for 10 epochs with the Adam optimizer with the learning rate of 0.005, resulting in the speed distribution

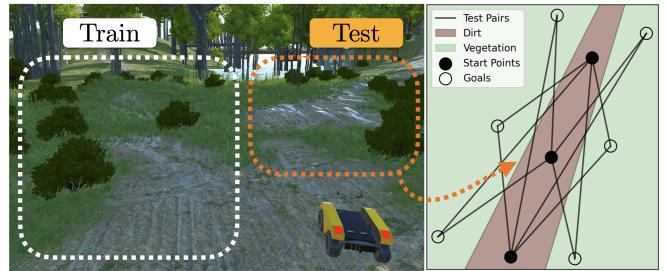


Fig. 6: Unity environment (left) where the robot learns and uses speed distribution maps for min-time planning. The key challenge is that bushes (1/4 of which lead to collision) are not distinguished semantically from the collision-free grass. After learning the distributions in the training region, the robot is tested to reach goals specified in regions with vegetation (right).

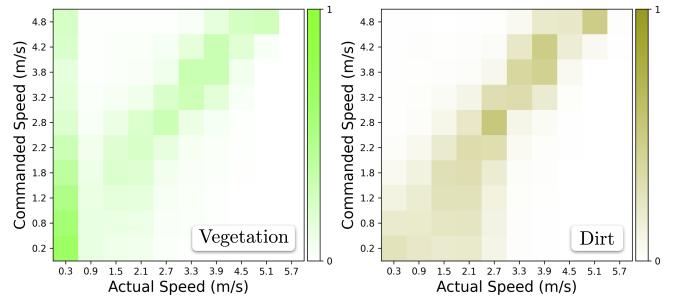


Fig. 7: Learned speed distributions for vegetation and dirt terrains, where the color gradient indicates magnitude of the probability mass. Note the bi-modal distribution associated with vegetation, whose outcome speeds are concentrated around 0 m/s (due to collisions with bushes) and around the commanded speeds. On the other hand, driving on dirt has a uni-modal, roughly linear relationship between commanded speed and realized speed.

maps visualized in Fig. 7, where the rows correspond to 10 binned commanded speeds and their resultant PMFs of speed outcomes.

During deployment, top-down 100×100 semantic images of the environment with 0.4 m cell resolution was processed by the network, as illustrated in Fig. 3. The training and testing areas contain mostly dirt and vegetation terrain types, and unknown semantic types were assumed to induce 0 m/s. Note that the semantic map and every commanded speed are paired and reshaped into a large batch input to the network. A 10 layer 100×100 speed distribution map with 10 output speed bins can be evaluated under 2 ms using the CPU (all runtimes reported on a desktop computer with an Intel i7-7700K CPU and 32GB RAM). The risk-aware speed maps can be extracted from the speed distribution maps as the convex combination of PMF mean and CVaR. When the entire autonomy stack was running and competing for CPU, mean and CVaR took 15 ms and 150 ms to compute, respectively. The computation for CVaR was done via rectangular approximation of density within each output speed bin (software optimization and GPU parallelization could likely reduce the CVaR calculation times substantially). Due to computational constraints, the traversability map was published at 2 Hz to the MPPI local planner.

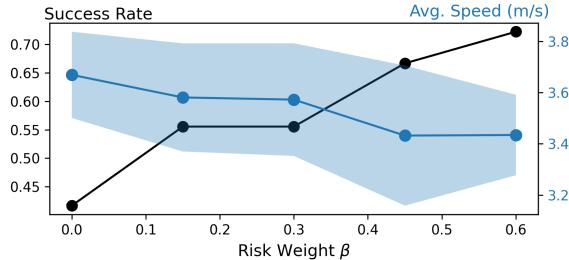


Fig. 8: Effect of risk weight $\beta \in [0, 1]$ on average speed of successful trials and the overall success rate, where CVaR is fixed at level $\alpha = 0.1$. By increasing β from 0 to 0.6, the robot achieves a much higher success rate ($42\% \rightarrow 72\%$), while the average speed reduces moderately as a performance trade-off. However, very high risk weight (e.g., $\beta > 0.8$) can cause excessive conservatism.

D. Min-Time Navigation Benchmark

In order to benchmark the effectiveness of learned risk-aware speed maps, the planner is tasked to navigate the robot from a set of pre-specified starting positions to goal positions in an unseen test environment, as illustrated in Fig. 6. Each test pair of starting point and goal were repeated for 3 trials over a range of risk weights $\beta \in [0.0, 0.15, 0.3, 0.45, 0.6]$. The goal tolerance was set to be a 3 m circle and the longest distance the robot had to travel was about 35 m. A timeout period of 40 s was imposed to terminate the trials where the robot got stuck or disoriented due to collisions with bushes. The planner models the Clearpath Warthog robot as a differential drive robot whose control input consists of two wheel speeds. During each MPPI optimization round, 500 control rollouts were sampled over 5 s horizon at 20 Hz according to noise standard deviation of 5 rad/s for each wheel. The estimated optimal control distribution was iteratively refined as MPPI ran in a receding horizon fashion. A small default speed of 0.5 m/s was used for estimating time-to-go to encourage the robot to approach the goal.

The benchmark results are shown in Fig. 8 which contain the success rate and the average speeds over the successful trials for a range of risk weights. As the risk weight increases, the robot has lower average speed but higher success rate ($42\% \rightarrow 72\%$) as beta increases from 0 to 0.6. Example rollouts produced by $\beta = 0$ and $\beta = 0.6$ are shown in Fig. 9, where the risk-aware trajectories ($\beta = 0.6$) overlap more with the dirt terrain (lower risk), whereas the planner that only accounts for the expectation ($\beta = 0$) prefers shorter paths that overlap more with vegetation (higher risk). As a result, one of the trajectories led to a collision with bushes, which caused localization errors and failure to reach goal.

VI. CONCLUSION & FUTURE WORK

This work proposed a new notion of traversability as the conditional speed distribution achievable by a robot, conditioned on the environment semantics and commanded speed. This representation can be learned directly from experienced trajectories and can be incorporated into various planning paradigms as a speed map. The proposed planning strategy was shown to lead to faster average time-to-goals compared

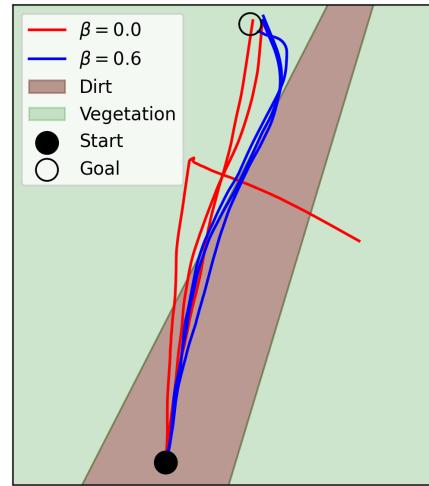


Fig. 9: Example MPPI min-time trajectories with risky ($\beta = 0$) and conservative ($\beta = 0.6$) behaviors. The risky planner's trajectories (red) follow a relatively straight path to the goal by traversing longer distance on vegetation, but there's a higher risk of collisions with bushes. Collision did happen to one of the trials and led to odometry failure due to wheel slips, which made the robot move in the wrong direction. On the other hand, the risk-aware planner's trajectories spend longer time on dirt, thus are more likely to reach the goal safely.

to other methods that did not consider the worst-case. Lastly, the proposed risk-aware strategy led to higher success rate in minimum-time navigation task in a high-fidelity simulator.

One area of future work is in automatically tuning the risk parameter β online, based on differences between realized speed and commanded speed. Additionally, the work could be extended to capture other forms of uncertainty, such as from out-of-distribution inputs or from probabilistic outputs from the semantic segmentation module. Finally, a learned cost-to-go estimator could be used to improve the cost assigned to the end of each MPPI rollout for improved performance.

ACKNOWLEDGMENT

Research was sponsored by the Army Research Office and was accomplished under Cooperative Agreement Number W911NF-21-2-0150. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

- [1] J. Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, and W. Whittaker, “Amblab: An autonomous rover for planetary exploration,” *Computer*, vol. 22, no. 6, pp. 18–26, 1989.
- [2] M. Massari, G. Giardini, and F. Bernelli-Zazzeri, “Autonomous navigation system for planetary exploration rover based on artificial potential fields,” in *Proceedings of Dynamics and Control of Systems and Structures in Space (DCSS) 6th Conference*, 2004, pp. 153–162.
- [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] A. Valada, G. Oliveira, T. Brox, and W. Burgard, "Deep multispectral semantic scene understanding of forested environments using multi-modal fusion," in *International Symposium on Experimental Robotics (ISER)*, 2016.
- [5] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "Adapnet: Adaptive semantic segmentation in adverse environmental conditions," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4644–4651.
- [6] A. Shaban, X. Meng, J. Lee, B. Boots, and D. Fox, "Semantic terrain classification for off-road autonomous driving," in *Conference on Robot Learning*. PMLR, 2022, pp. 619–629.
- [7] F. Sadeghi and S. Levine, "Cad2rl: Real single-image flight without a single real image," in *Robotics: Science and Systems XIII*, 2017.
- [8] M. Everett, J. Miller, and J. P. How, "Planning beyond the sensing horizon using a learned context," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1064–1071.
- [9] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A rugged dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [10] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1110–1116.
- [11] 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.
- [12] M. Ono, T. J. Fuchs, A. Steffy, M. Maimone, and J. Yen, "Risk-aware planetary rover operation: Autonomous terrain classification and path planning," in *2015 IEEE aerospace conference*. IEEE, 2015, pp. 1–10.
- [13] A. A. Pereira, J. Binney, G. A. Hollinger, and G. S. Sukhatme, "Risk-aware path planning for autonomous underwater vehicles using predictive ocean models," *Journal of Field Robotics*, vol. 30, no. 5, pp. 741–762, 2013.
- [14] L. De Filippis, G. Guglieri, and F. Quagliotti, "A minimum risk approach for path planning of uavs," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 203–219, 2011.
- [15] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011.
- [16] B. Luders, M. Kothari, and J. How, "Chance constrained rrt for probabilistic robustness to environmental uncertainty," in *AIAA guidance, navigation, and control conference*, 2010, p. 8160.
- [17] A. Hakobyan, G. C. Kim, and I. Yang, "Risk-aware motion planning and control using cvar-constrained optimization," *IEEE Robotics and Automation letters*, vol. 4, no. 4, pp. 3924–3931, 2019.
- [18] D. D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Aghamohammadi, "Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation," in *Robotics: Science and Systems*. RSS Foundation, 2021, pp. 1–21.
- [19] A. Majumdar and M. Pavone, "How should a robot assess risk? towards an axiomatic theory of risk in robotics," in *Robotics Research*. Springer, 2020, pp. 75–84.
- [20] D. Silver, J. A. Bagnell, and A. Stentz, "Applied imitation learning for autonomous navigation in complex natural terrain," in *Field and Service Robotics*. Springer, 2010, pp. 249–259.
- [21] D. Silver, J. Bagnell, and A. Stentz, "High performance outdoor navigation from overhead data using imitation learning," *Robotics: Science and Systems IV, Zurich, Switzerland*, vol. 1, 2008.
- [22] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al., "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.
- [23] 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," *arXiv preprint arXiv:1709.07174*, 2017.
- [24] Z. Wang, X. Xiao, G. Warnell, and P. Stone, "Apple: Adaptive planner parameter learning from evaluative feedback," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7744–7749, 2021.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] F. G. Oliveira, A. A. Neto, D. Howard, P. Borges, M. F. Campos, and D. G. Macharet, "Three-dimensional mapping with augmented navigation cost through deep learning," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 3, pp. 1–21, 2021.
- [30] S. Otte, C. Weiss, T. Scherer, and A. Zell, "Recurrent neural networks for fast and robust vibration-based ground classification on mobile robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5603–5608.
- [31] J. Larson, M. Trivedi, and M. Bruch, "Off-road terrain traversability analysis and hazard avoidance for ugv's," CALIFORNIA UNIV SAN DIEGO DEPT OF ELECTRICAL ENGINEERING, Tech. Rep., 2011.
- [32] 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.
- [33] ———, "G-vom: A gpu accelerated voxel off-road mapping system," *arXiv preprint arXiv:2109.13176*, 2021.
- [34] T. Guan, Z. He, D. Manocha, and L. Zhang, "Ttm: Terrain traversability mapping for autonomous excavator navigation in unstructured environments," *arXiv preprint arXiv:2109.06250*, 2021.
- [35] Y. Tan, N. Virani, B. Good, S. Gray, M. Yousefhussien, Z. Yang, K. Angeliu, N. Abate, and S. Sen, "Risk-aware autonomous navigation," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*, vol. 11746. International Society for Optics and Photonics, 2021, p. 117461D.
- [36] B. Suger, B. Steder, and W. Burgard, "Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3d-lidar data," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3941–3946.
- [37] V. Wiberg, E. Wallin, T. Nordfjell, and M. Servin, "Control of rough terrain vehicles using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 390–397, 2021.
- [38] S. Russell and P. Norvig, "Artificial intelligence: a modern approach," 2002.
- [39] "Arl autonomy stack," 2022. [Online]. Available: <https://www.arl.army.mil/business/collaborative-alliances/current-cras/sara-cra/sara-overview/>
- [40] "Clearpath warthog," 2022. [Online]. Available: <https://clearpathrobotics.com/warthog-unmanned-ground-vehicle-robot/>