



# Risk-Aware Navigation for Mobile Robots in Unknown 3D Environments

Elie Randriamiarintsoa, Johann Laconte, Benoit Thuilot, Romuald Aufrère

## ► To cite this version:

Elie Randriamiarintsoa, Johann Laconte, Benoit Thuilot, Romuald Aufrère. Risk-Aware Navigation for Mobile Robots in Unknown 3D Environments. 2023. hal-04197811

**HAL Id: hal-04197811**

**<https://hal.science/hal-04197811>**

Preprint submitted on 6 Sep 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Risk-Aware Navigation for Mobile Robots in Unknown 3D Environments

Elie Randriamiarintsoa<sup>1</sup>, Johann Laconte<sup>2</sup>, Benoit Thuilot<sup>1</sup>, Romuald Aufrère<sup>1</sup>

**Abstract**—Autonomous navigation in unknown 3D environments is a key issue for intelligent transportation, while still being an open problem. Conventionally, navigation risk has been focused on mitigating collisions with obstacles, neglecting the varying degrees of harm that collisions can cause. In this context, we propose a new risk-aware navigation framework, whose purpose is to directly handle interactions with the environment, including those involving minor collisions. We introduce a physically interpretable risk function that quantifies the maximum potential energy that the robot wheels absorb as a result of a collision. By considering this physical risk in navigation, our approach significantly broadens the spectrum of situations that the robot can undertake, such as speed bumps or small road curbs. Using this framework, we are able to plan safe trajectories that not only ensure safety but also actively address the risks arising from interactions with the environment.

## I. INTRODUCTION

Navigating in unknown 3D environments is a crucial task for mobile robots. Before being able to move, a robot must represent the environment in which it evolves. A well-known and efficient way to map the environment is the occupancy grids introduced by Elfes [1]. An occupancy grid provides the robot with information about the potential presence of an obstacle at a given position. However, this information alone does not encompass all the capabilities of the robot to maneuver within its environment. For example, a wheeled robot can safely cross grasses, a curb, or a speed bump at low speed, as shown in Figure 1. Therefore, a way to assess their hazardous nature is necessary to evolve in these environments. Motivated by this fact, numerous recent works [2] have contributed to risk-aware navigation. However, Laconte *et al.* [3] demonstrated that the Bayesian occupancy grid, which stores the probability of collision of a given position, is ill-suited to compute the risk over paths. Indeed, the probability of collision, computed as the joint probability that every cell is free of obstacles, is highly dependent on the grid tessellation size. To overcome this difficulty, they developed a novel framework called Lambda-Field to assess physics-based risks on occupancy grids. Moreover, in occupancy grids framework, most navigation methods [4] use geometric and semantic reasoning to handle the obstacles present in the environment, such as curbs, traffic cones, speed bump, sidewalk, buildings, traffic circle and traffic lights shown in Figure 1. Laconte *et al.* [3] showed that path planning becomes more intuitive and meaningful when a physical



Fig. 1. Example of a situation that a vehicle might encounter while navigating. Speed bumps are frequent obstacles an intelligent vehicle has to safely overcome. With our framework, the vehicle is able to make more effective decisions based on the level of risk it is allowed to take.

risk metric is used. However, their work has been developed solely for 2D environments. Here, we adapt their work to 3D environments and use physical, risk-based reasoning to deal with the obstacles represented in Figure 1.

The main contributions of this paper are i) an extension of the Lambda-Field [3] to 3D environments; ii) a generalization of the risk that take into account traversable obstacles; iii) a mathematical formulation of a local risk-aware path planning algorithm in the Lambda-Field; iv) a demonstration of the applicability of the method, using data acquired in real urban environment.

## II. RELATED WORK

An important challenge in local motion planning is to find a feasible path that matches the robot's capabilities while being safe to traverse. Prior to identifying such a path, it is essential to accurately map the immediate surroundings of the robot, thereby representing its local environment. A popular representation of the robot surroundings is the occupancy grids, introduced by Elfes [1]. The main idea of the occupancy grids is to tessellate the environment into cells and to store in each cell the information of occupancy. Coué *et al.* [5] enhanced the previous idea by adding a Bayesian layer to the approach. Their work led to the Bayesian Occupancy Filter (BOF) techniques.

Using these Bayesian occupancy grids, standard motion planning frameworks plan collision-free paths by assessing the probability of collision of each potential robot path. However, assessing only the risk of colliding with an obstacle will always lead the robot to go around them. Therefore, the

<sup>1</sup>Université Clermont Auvergne, Clermont Auvergne INP, CNRS, Institut Pascal, F-63000 Clermont-Ferrand, France. elie.randriamiarintsoa@uca.fr

<sup>2</sup>University of Toronto Institute for Aerospace Studies (UTIAS), 4925 Dufferin St, Ontario, Canada. johann.laconte@utoronto.ca

collision risk does not fully exploit the robot ability. Indeed, a mobile robot can cross tall grass or a speed bump even if it results in collisions. To allow robots to make more insightful choices, several approaches propose to integrate the notion of risk in motion planning by using risk maps. Unlike conventional Bayesian occupancy grids, where each cell stores the occupancy probability of a given position, a risk map stores the risk at this position. Schroder *et al.* [6] designed a risk map for cognitive vehicles. The defined risk map is used to prevent the robot from approaching the hazardous obstacles, such as pedestrians or cars. Pereira *et al.* [7] used historical shipping traffic and bathymetry data of coastal regions to create a risk map for underwater vehicles. Assigning a probabilistic risk value to each position that is likely to be occupied allows to avoid potential hazardous collisions. Primates *et al.* [8] used a risk map to quantify the risk of an unmanned aerial vehicle flying over a given position to cause lethal incidents in an inhabited area. Even though these techniques produce good results, they all make the assumption that the risk is only dependent on the robot position, while intuitively the risk depends also on the robot state and capabilities. As such, our work proposes a risk model that exploits both the state of the environment and the robot ability to assess the risk of a given path.

In risk-aware navigation, Majumdar *et al.* [9] provide insight about which metrics should be used. Among them, the expected value and the Conditional value-at-risk (CVaR) are identified as valid candidates. The CVaR is used to capture the worst-case expected hazardous event that could happen over a given horizon of time, and has been used by several recent works [10], [11]. Hakobyan *et al.* [12] measure the risk of colliding with randomly moving obstacles. They define the problem of path planning as a constrained optimization where the distance to the obstacles must be superior or equal to a fixed threshold. They demonstrate that this formulation allows to plan an effective path for a quadrotor in a 3D environment, while adjusting the safety and prudence of the motions. For mobile robots, Fan *et al.* [13] enhance the previous work by adding multiple sources of risk, such as the slippage risk. A new risk map is then created by aggregating the risks that come from these different sources. Cai *et al.* [14] quantify the risk by converting a learned speed distribution map to a risk cost value. In contrast to our work, traversability is captured via experienced trajectories. Koval *et al.* [15] propose another risk-aware path planning which is also based on a priori data, precisely on a known map. In [3], the authors introduce the Lambda-Field, a generic method to assess a physical risk over a continuous path. The risk is defined as the expected force of collision along a path. They introduced a new mapping framework where each cell stores the density of collisions that could be hazardous to the robot. In contrast to the Bayesian occupancy grids framework, the Lambda-Field framework computes the integral of the risk over a given path. This approach has the ability to retain the physical units of the risk. They have demonstrated the relevance of the approach in 2D static environments. They extended their work to unstructured [16]

and dynamic environments [17], yet still only considering 2D obstacles. In this article, we adapt their framework to 3D static environments and propose a new physically coherent risk measure.

### III. PRELIMINARIES

In this section, we briefly present the framework developed by Laconte *et al.* [3]. In a similar way as occupancy grid frameworks, the environment is tessellated into cells, where all cells have a fixed size and area  $\Delta a \in \mathbb{R}_{>0}$ . The core difference of their approach, compared to a standard Bayesian occupancy grid, is the information that is stored in each cell. Instead of estimating the probability of collision, they estimate the density of collision  $\lambda \in \mathbb{R}_{\geq 0}$  for each cell, also called the intensity. The probability of collision within the cell is then  $\lambda \Delta a$ . The higher the intensity of a cell, the more likely it is that a hazardous event will happen in this cell. On the contrary, an intensity of zero means that the cell will never lead to a harmful collision and can be crossed safely. As shown in [3], the probability of facing at least one hazardous event (i.e., collision) over a path  $\mathcal{P}$  in the discretized field is

$$\mathbb{P}(\text{coll}|\mathcal{P}) = 1 - \exp\left(-\Delta a \sum_{c_i \in \mathcal{C}} \lambda_i\right), \quad (1)$$

where  $\mathcal{C}$  is the set of cells crossed by the path  $\mathcal{P}$ , and  $\lambda_i$  is the intensity of a cell  $c_i \in \mathcal{C}$ .

In order to build this map, Laconte *et al.* [16] use a 2D lidar. If  $e$  is the error region area attached to every lidar measurement, then a cell  $c_i$  is measured as hazardous if it is located within the error region area of a lidar beam impact. Otherwise, if the lidar beam traversed the cell without returning a collision, the cell is measured as safe. Under these considerations, the intensity  $\lambda_i$  of each cell is computed using an expectation maximization approach. Let  $s_i$  be the number of times a cell  $c_i$  is measured as safe and  $h_i$  the number of times a cell  $c_i$  is measured as hazardous, it can be shown, see [16], that the intensity of the cell  $c_i$  can be expressed as

$$\lambda_i = \frac{1}{e} \ln \left(1 + \frac{h_i}{s_i}\right). \quad (2)$$

Furthermore, the main advantage of the Lambda-Field is that it enables the computation of the probability of collision, but also of a generic risk over a path. Following the demonstration in [16], the expected value of a generic deterministic risk over a path  $\mathcal{P}$  crossing the cells  $\{c_i\}_{0:N-1}$ , is given by

$$\mathbb{E}[r(X)] = \sum_{i=0}^{N-1} K_i r(x_i), \quad (3)$$

where  $x_i$  corresponds to the position on the path associated to the cell  $i$  and  $r(x_i)$  is a generic risk function that gives the value of the deterministic risk if a collision would happen at  $x_i$ .  $K_i$  gives the probability of encountering an harmful

event at  $x_i$  and is given by

$$K_i = \exp \left( -\Delta a \sum_{j=0}^{i-1} \lambda_j \right) (1 - \exp(-\Delta a \lambda_i)). \quad (4)$$

$\mathbb{E}[r(X)]$  encompasses both the state of the environment through the intensity field, and the state of the robot when it comes at a given position along the path, though the risk function  $r(x_i)$ . As such, this framework provides a way to compute a meaningful risk over a path in occupancy grids.

#### IV. MAPPING

We present here our extension of the Lambda-Field to take into account traversable 3D obstacles in the environment. For that, we use a 3D lidar to compute a Digital Elevation Map (DEM) [18]. By aggregating multiple point clouds over time, we compute the difference in elevation of the environment. This is achieved by taking into account the elevations of the eight neighboring cells for each grid cell. Then, this DEM is used to compute the Lambda-Field.

To model traversable obstacles in the context of urban environments containing road curbs and speed bumps. We define a cell  $c_i$  as safe if the value of the difference in elevation is below a threshold value  $H_{\text{safe}}$ ; otherwise, this cell is defined as hazardous. The intensity equation Equation 2 is however modified to take into account the severity of the event and will stop the robot in its course. As such, the intensity of a cell  $c_i$  is computed with

$$\lambda_i = \frac{1}{e} \ln \left( 1 + \frac{h_i}{s_i} \right) p_i \quad (5)$$

where quantity  $p_i$  describing the severity of this collision is defined by

$$p_i = \min \left( \frac{|H_i|}{R}, 1 \right), \quad (6)$$

with  $H_i$  the difference in elevation of the cell  $c_i$  with reference to its neighbors and  $R$  the radius of the wheel. As such, the likelihood it stop  $p_i$  is one when the obstacle is higher than the wheel radius and the robot has no chance to go over the obstacle, and tends to zero for small obstacles which are harmful. In the conservative case where we assume the robot is not able to go over any obstacle, then  $p_i = 1$  and the measurement equation returns to Equation 2. Equation 5 is then a generalization of the approach where we consider that the robot is also able to traverse over small obstacles.

As an example, Figure 2 depicts a Lambda-Field computed using Equation 5, for the environment shown in Figure 1. The global Lambda-Field map is shown at the top in Figure 2 and the construction of the map around the speed bump (in yellow in Figure 1) at different times is illustrated below. At time  $t_1$ , the speed bump is in the lidar range and the intensities  $\lambda_i$  representing the speed bump began to converge. The speed bump is better represented at time  $t_2$  through new and additional measurements. As the speed bump has been completely scanned at time  $t_3$ , the intensities associated with it have completely converged. One can note that the curbs, traffic cones, speed bump, buildings and traffic lights

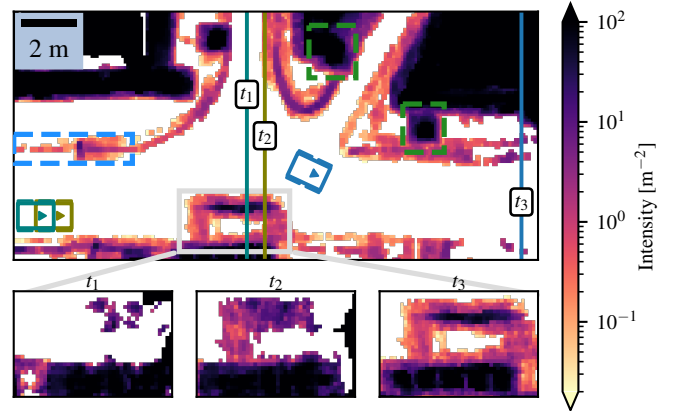


Fig. 2. Example of Lambda-Field of the environment depicted in Figure 1. The area containing the speed bump and the cones is enlarged and shown at different times. The lidar range (vertical line) and the robot (box) pose for each of these times are illustrated in teal ( $t_1$ ), dull green ( $t_2$ ) and blue ( $t_3$ ). The left curb (light blue) and the traffic lights (green) seen in Figure 1 are outlined in dashed lines.

poles are well detected. The borders of the speed bump have high intensity values, when its ascending and descending portions have more nuanced intensities, meaning that harmful collisions are less prone to happen. As a result, the robot will be able to cross the speed bump by taking a reasonable risk. Additionally, buildings, traffic light poles, and traffic cones have high intensities. These obstacles cannot be crossed by the robot without taking an unreasonable risk.

#### V. RISK ASSESSMENT

In order to navigate in the Lambda-Field, we define a risk function that reflects the potential obstacles that the robot can face. We choose to describe the risk by the maximum potential energy absorbed by the wheels.

We illustrate in Figure 3 the modeling of the wheel. The wheel is modeled as a deformable disk, see Figure 3. When the wheel crosses an elevated obstacle, it deforms at the point where the two objects collide. This deformation is approximated with the deformation of a spring of stiffness  $k_r$ . In order to find the maximum amplitude  $l_m$  of the tire

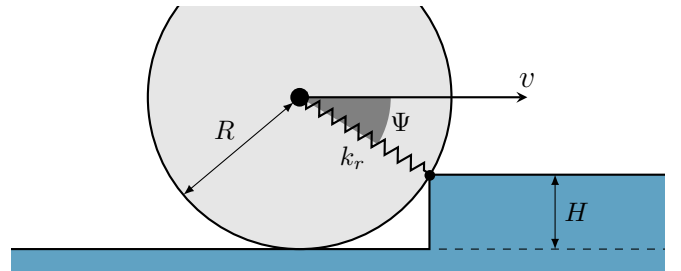


Fig. 3. Modeling of the wheel of radius  $R$  in collision with the curb (in blue) of height  $H$ , with a speed  $v$  and angle  $\Psi$ . The deformation of the wheel due to the collision is approximated with the deformation of a spring of stiffness  $k_r$ .

compression due to the collision, we solve the following differential equation:

$$\ddot{l} + \omega^2 l = 0 \quad \text{with } \omega = \sqrt{\frac{k_r}{m}} \quad (7)$$

where  $l$  is the compression at the contact point and  $m$  is the mass of the vehicle. As such, the maximum amplitude  $l_m$  of the tire compression is

$$l_m = \frac{v \cos(\Psi)}{\omega} \quad (8)$$

where  $v$  is the linear velocity of the robot and  $\Psi = \arcsin(\frac{R - \min(H, R)}{R})$  is the angle of attack of the collision. The angle  $\Psi$  depends on the elevation of the obstacle and the radius  $R$  of the wheel. To be conservative, we take the maximum difference in elevation  $H$  of the obstacles that lie in the transverse axis of the robot path  $\mathcal{P}$ . Finally, the risk function is defined as the maximum potential energy that will be absorbed in the wheels at the time of collision and is computed by

$$r(X) = \frac{1}{2} k_r l_m^2 \quad (9)$$

As such, the risk function  $r(X)$  has a clear physical meaning and is expressed in Joule. In the next section, we use this risk assessment to construct safe paths in 3D environments.

## VI. PATH PLANNING

The goal of the robot is to follow a local reference path  $\mathcal{P}_{ref} \subset \mathbb{R}^3$  that is assumed to be known. However, unforeseen events, detected by the robot's sensors, can make this path impassable. The aim of our path planning framework is to find the optimal path  $\mathcal{P}^* \subset \mathbb{R}^3$  that is the closest to the reference path  $\mathcal{P}_{ref}$ , while maintaining a reasonable level of risk along the path. Namely, a path  $\mathcal{P}$  is accepted only if the risk undertaken by the robot along the path is below a user-defined risk threshold  $r_{threshold}$ , consistent with the preferences of the user and the robot abilities. The optimal path  $\mathcal{P}^*$  also minimizes the traversal time of the robot.

To address the problem, we rely on a Nonlinear Model Predictive Control (NMPC) approach, where we use the Ackermann steering model as the prediction model. Using a small sample time  $\Delta t$ , the kinematic equations can be discretized into

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \Delta t \begin{bmatrix} v_k \cos \theta_k \\ v_k \sin \theta_k \\ v_k \tan \delta_k / L \end{bmatrix}, \quad (10)$$

where  $k$  represent the discrete time,  $L$  is the length of the wheelbase of the robot,  $(x_k, y_k)$  and  $\theta_k$  are respectively the centre position of the rear axle and the yaw angle of the robot in the absolute frame. Finally,  $v_k$  is the linear velocity of the robot, and  $\delta_k$  the steering angle of the robot. The compact form of Equation 10 is written as  $\mathbf{x}_{k+1} = f_d(\mathbf{x}_k, \mathbf{u}_k)$ , where  $\mathbf{x}_{k+1} = [x_{k+1}, y_{k+1}, \theta_{k+1}]^\top$  is the predicted state vector,  $\mathbf{x}_k = [x_k, y_k, \theta_k]^\top$  is the state vector, and  $\mathbf{u}_k = [v_k, \delta_k]^\top$  is the control vector. The reference trajectory associated with the path  $\mathcal{P}_{ref}$  is defined as a set of desired states  $[\mathbf{x}_0^r, \dots, \mathbf{x}_{N_p}^r]^\top$ , where  $\mathbf{x}_k^r$  is the desired state at time  $k$ .

According to our objective and constraints, we want to minimize the errors between the predicted and the reference paths over a finite prediction horizon of length  $N_p$  while considering kinematic, control, and risk constraints. We also

want to minimize the traversal time and penalize the error in the final state. The penalization of this latter error is used to force the robot to approach the farthest reference goal  $\mathbf{x}_{N_p}^r$  located in the perceived and therefore secure navigable area. As a result, the risk-aware navigation based on NMPC consists in computing a trajectory  $[\mathbf{x}_0, \dots, \mathbf{x}_{N_p}]^\top$  by the following quadratic constraint optimization:

$$\begin{aligned} \min \quad & Z = Z_1 + Z_2 + Z_3 \\ \text{s. t.} \quad & \mathbf{x}_{k+1} = f_d(\mathbf{x}_k, \mathbf{u}_k) \\ & \mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max} \\ & \mathbb{E}[r(X)] \leq r_{threshold} \end{aligned} \quad (11)$$

where  $Z$  is the cost function of the optimization problem, defined as a combination of three penalty terms described below, and  $(\mathbf{u}_{\min}, \mathbf{u}_{\max})$  are respectively the lower and upper bounds of the control vector.

The first penalty term is used to prevent the robot from deviating too far from the reference path  $\mathcal{P}_{ref}$ :

$$Z_1 = \sum_{k=0}^{N_p-1} (\mathbf{x}_k - \mathbf{x}_k^r)^\top \mathbf{Q} (\mathbf{x}_k - \mathbf{x}_k^r), \quad (12)$$

where  $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$  is the weight matrix used to penalize the state component parts. The second penalty term ensures that the robot approaches to the farthest desired position:

$$Z_2 = (\mathbf{x}_{N_p} - \mathbf{x}_{N_p}^r)^\top \mathbf{Q}_N (\mathbf{x}_{N_p} - \mathbf{x}_{N_p}^r), \quad (13)$$

where  $\mathbf{x}_{N_p}$  is the last predicted state, and  $\mathbf{Q}_N \in \mathbb{R}^{3 \times 3}$  is the weight matrix used to penalize the final state component parts. The last penalty term is used to prioritize a minimum traversal time:

$$Z_3 = \sum_{k=0}^{N_p-1} w_v (v_k - v_{max})^2, \quad (14)$$

where  $v_{max}$  is the maximum velocity the robot can reach and  $w_v \in \mathbb{R}$  is the weighting coefficient that penalizes low velocities. The final constraint  $\mathbb{E}[r(X)] \leq r_{threshold}$  forces the robot to find a trajectory that is safe to traverse. Therefore, any trajectory that exceeds the risk threshold  $r_{threshold}$  will be rejected. Note that we do not add the risk to the cost function, as we would lose its physical meaning. The risk is here defined as a hard constraint to ensure that it is never exceeded. Finally, the optimal path  $\mathcal{P}^*$  is the one that minimizes the cost function in respects to the risk and the control constraints.

## VII. VALIDATION OF THE FRAMEWORK

To show the applicability of this framework, we ran three scenarios in the environment depicted in Figure 1, in which three different risk thresholds were considered. We used real data for the perception part and simulated the path planning with a simulation model of the robot seen in Figure 1. The simulations were performed on a computer equipped with an 11th Gen Intel Core i7-11850H processor and an NVIDIA GeForce RTX 3080 graphics card. In all scenarios, the robot starts in front of the speed bump with zero steering angle and



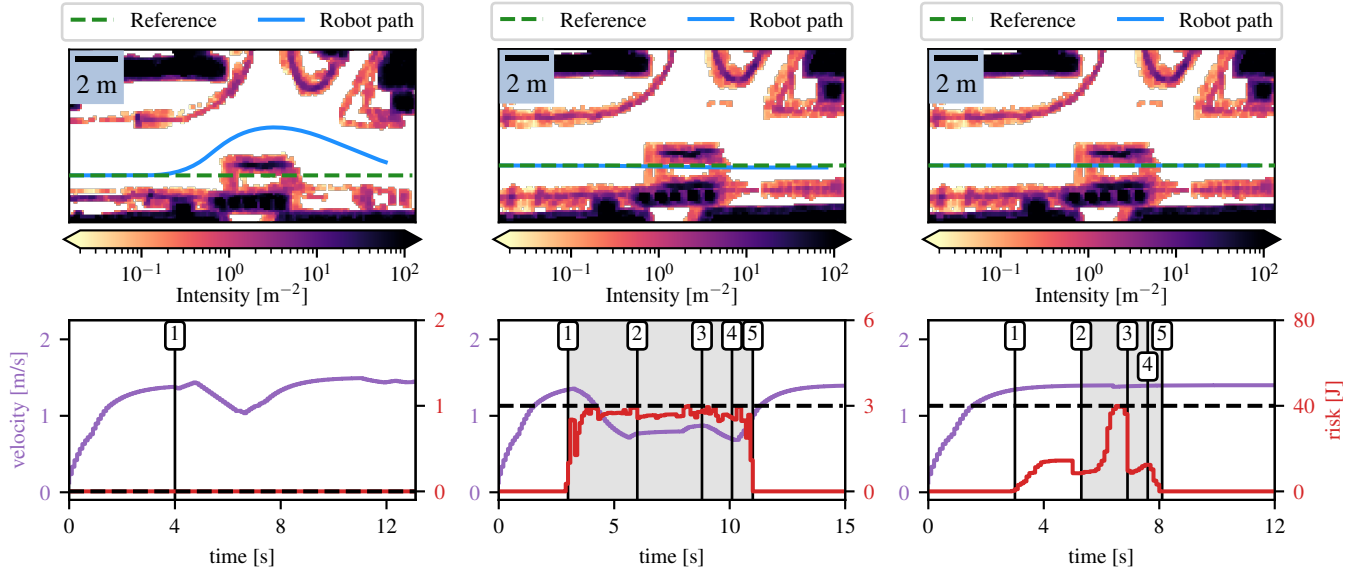


Fig. 4. Example of an environment where an unexpected obstacle (speed bump) occurred on the reference path (dashed green). We investigated three risk thresholds, 0 J (left), 3 J (middle) and 40 J (right). For each threshold, the path taken by the robot is in blue. Its velocity and risk are depicted in purple and red. Events: (1): Speed bump came into the robot's perception field. (2): Robot started to climb the speed bump. (3): Robot reached the speed bump top. (4): Robot started to get down. (5): Robot reached the asphalt. The gray shaded areas show the duration when the robot traversed the speed bump.

zero speed. The Lambda-Field construction involves setting the difference in elevation threshold  $H_{\text{safe}}$  to 5 cm and the  $e$  value to  $1 \text{ cm}^2$ . Each Lambda-Field grid map consist of  $200 \times 200$  cells, where each cell has a size of  $10 \times 10 \text{ cm}$ , resulting in a map of  $20 \times 20 \text{ m}$ . The simulated robot has a wheel radius  $R$  of 25 cm and a mass  $m$  of 50 kg. The stiffness  $k_r$  of each robot wheel is set to 150 000 N/m. The maximum velocity  $v_{\text{max}}$  of the robot is set to 1.5 m/s, while the steering angle range  $(\delta_{\text{min}}, \delta_{\text{max}})$  is set to  $\pm 11^\circ$ . The sampling time  $\Delta t$  is set to 100 ms. The weight matrix  $\mathbf{Q}$  is a diagonal matrix of elements  $[0.05, 0.05, 0.05]$ . The weight matrix  $\mathbf{Q}_N$  is a diagonal matrix of elements  $[1, 1, 1]$ . Therefore, if an unexpected obstacle prevents the robot from tracking the reference path, but the reference goal can be reached by a path deviating from the reference path, then the path planner will be able to choose this path. Finally, the weight coefficient  $w_v$  is equal to 0.1.

We show the results in Figure 4, where each column corresponds to the result of a scenario. The first row in Figure 4 shows the path of the robot as well as the reference path, which is a straight line passing through the speed bump. The second row in Figure 4 shows the velocity profile and the risk during the traversal. The graphs are labeled with numbers corresponding to different times of interests:  $t_1$ : the speed bump is detected;  $t_2$ : the robot ascend the speed bump;  $t_3$ : the robot reaches the top speed bump;  $t_4$ : the robot descends the speed bump; and  $t_5$ : the robot comes back to the road.

In the first scenario, the risk threshold is set to zero, meaning that the robot is not allowed to take any risk. As long as the robot does not see the speed bump, the reference path is tracked. At time  $t_1$ , the robot can no longer track the reference path, as crossing the speed bump is risky. The path

planning formulation leads the robot to go around the speed bump. One can note that throughout the traversal, the robot maximizes its velocity without ever crossing a potential risky cell, as no risk is allowed. This scenario shows that setting the risk to zero is the same as assessing the risk of collision and imposing that the robot goes around the obstacles.

However, in some cases, avoiding may not be possible (for example, the left lane is forbidden due to traffic rules or occupied by another vehicle) and if no risk is allowed the robot would then stop in front of the speed bump. With our approach, it is possible to reach the goal by managing the risk threshold in accordance with the capability of the robot. In the scenario shown in the second column in Figure 4, we set the risk threshold  $r_{\text{threshold}}$  to 3 J, meaning in our case that 6 mm compression of the wheel is the maximum allowed. One can see that the robot passes through the speed bump, but at a low speed. The robot slows down at time  $t_1$ , as the speed bump is detected by the robot. Then, from time  $t_2$  to time  $t_5$ , our path planner regulates the speed of the robot to stay below the risk threshold. The path planner maintains an adequate speed to cross the speed bump safely, finding a balance between minimizing the traversal time and the risks taken by the robot. From time  $t_3$  to  $t_4$ , the robot is on the speed bump. Before leaving it, the robot again detects hazardous events on the reference path, leading the robot to slow down. At time  $t_4$ , the robot starts to descend the speed bump at a reasonable speed to manage the risk that has appeared on the path. Finally, at time  $t_5$ , the robot leaves the obstacle, accelerating again. This scenario shows that the robot can handle a consistent amount of risk in complicated situations.

In the last scenario, we increase the risk threshold to 40 J, meaning that we accept a 23 mm maximum tire's

compression. Intuitively, the robot goes faster than in the previous scenario. At time  $t_1$ , the speed bump is detected. The planner stabilizes the velocity of the robot to traverse the speed bump, while not expecting a risk exceeding 40 J. Until it reaches its target, the robot tracks the reference path at its greatest allowable velocity. As such, our framework is able to produce meaningful paths by considering the potential hazards associated with navigating through 3D obstacles.

## VIII. CONCLUSION

In this paper, we presented a method of risk-aware navigation in unknown 3D environments. We propose a generalization of the work of Laconte *et al.* [3], taking into account the traversal of small obstacles such as road curbs or speed bumps. We showed how to compute the associated Lambda-Field, using 3D lidar measurements. The resulting map is used to evaluate the expected maximum potential energy that the robot's wheels will absorb in the event of a collision along a given path. Finally, we presented a path planning formulation that take the risk into account as a hard constraint. Using our formulation, we showed that the risk function is well-fitted for risk-aware navigation in urban environment. While being able to mimic a standard path planning approach by setting the risk threshold to zero, our framework can also generate a path going over obstacles if the risk is tolerable. This work focused on explaining the theoretical framework and its applicability.

Future works will focus on improving the path planning part and demonstrating the pertinence of the framework in larger-scale experiments. The exploration of alternative risk metrics, such as CVaR, will be pursued to account for tail events. We intend to extend our framework to enable the robot to perform long-term missions. Indeed, providing mobile robots with such capabilities will augment the autonomy of intelligent vehicles in rural environments. A global path planner based on OpenStreetMap [19] will be included to the framework. Furthermore, we will conduct extensive experiments on the framework, incorporating quantitative evaluations. Finally, we will investigate the possibility of adding several risks to further constrain the path-planning algorithm, such as the risk of crossing a continuous lane marking.

## ACKNOWLEDGMENT

This work has been supported by the AURA Region and the European Union (FEDER) through the LOG SSMI project of CPER 2020 MMaSyF challenge.

## REFERENCES

- [1] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.
- [2] Y. Shi, K. Jiang, J. Li, J. Wen, Z. Qian, M. Yang, K. Wang, and D. Yang, "Grid-centric traffic scenario perception for autonomous driving: A comprehensive review," *arXiv preprint arXiv:2303.01212*, 2023.
- [3] J. Laconte, C. Debain, R. Chapuis, F. Pomerleau, and R. Aufrère, "Lambda-field: A continuous counterpart of the bayesian occupancy grid for risk assessment," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 167–172.
- [4] B. Patle, G. Babu L, A. Pandey, D. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, no. 4, pp. 582–606, 2019.
- [5] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, "Bayesian occupancy filtering for multitarget tracking: An automotive application," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, 2006.
- [6] J. Schroder, T. Gindele, D. Jagszent, and R. Dillmann, "Path planning for cognitive vehicles using risk maps," in *2008 IEEE Intelligent Vehicles Symposium*, 2008, pp. 1119–1124.
- [7] A. A. Pereira, J. Binney, B. H. Jones, M. Ragan, and G. S. Sukhatme, "Toward risk aware mission planning for autonomous underwater vehicles," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3147–3153.
- [8] S. Primates, G. Guglieri, and A. Rizzo, "A risk-aware path planning strategy for UAVs in urban environments," *Journal of Intelligent & Robotic Systems*, vol. 95, pp. 629–643, 2019.
- [9] A. Majumdar and M. Pavone, "How should a robot assess risk? towards an axiomatic theory of risk in robotics," in *Robotics Research: The 18th International Symposium ISRR*, Springer, 2020, pp. 75–84.
- [10] R. T. Rockafellar, S. Uryasev, *et al.*, "Optimization of conditional value-at-risk," *Journal of risk*, vol. 2, pp. 21–42, 2000.
- [11] R. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of Banking & Finance*, vol. 26, no. 7, pp. 1443–1471, 2002.
- [12] 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.
- [13] D. D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Agha-Mohammadi, "Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation," *arXiv preprint arXiv:2103.02828*, 2021.
- [14] 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.
- [15] A. Koval, S. Karlsson, and G. Nikolakopoulos, "Experimental evaluation of autonomous map-based spot navigation in confined environments," *Biomimetic Intelligence and Robotics*, vol. 2, no. 1, p. 100035, 2022.
- [16] J. Laconte, A. Kasmi, F. Pomerleau, R. Chapuis, L. Malaterre, C. Debain, and R. Aufrère, "A novel occupancy mapping framework for risk-aware path planning in unstructured environments," *Sensors*, vol. 21, no. 22, 2021.
- [17] J. Laconte, E. Randriamiarintsoa, A. Kasmi, F. Pomerleau, R. Chapuis, C. Debain, and R. Aufrère, "Dynamic lambda-field: A counterpart of the bayesian occupancy grid for risk assessment in dynamic environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4846–4853.
- [18] Y. Pan, X. Xu, Y. Wang, X. Ding, and R. Xiong, "Gpu accelerated real-time traversability mapping," in *2019 IEEE International Conference on Robotics and Biomimetics (RO-BIO)*, 2019, pp. 734–740.
- [19] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.