Risk-aware Motion Planning for Collision-tolerant Aerial Robots subject to Localization Uncertainty

Paolo De Petris, Mihir Dharmadhikari, Huan Nguyen, and Kostas Alexis

***Abstract*— This paper contributes a novel strategy towards risk-aware motion planning for collision-tolerant aerial robots subject to localization uncertainty. Attuned to the fact that micro aerial vehicles are often tasked to navigate within GPS-denied, possibly unknown, confined and obstacle-filled environments the proposed method exploits collision-tolerance at the robot design level to mitigate the risks of collisions especially as their likelihood increases with growing uncertainty. Accounting for the maximum kinetic energy with which an impact is considered safe, alongside the robot dynamics, the planner builds a set of admissible uncertainty-aware and collision-inclusive paths over a horizon involving multiple motion steps. The first step of the best path is executed by the robot, while the procedure is then repeated in a receding horizon manner. Evaluated in extensive simulation studies and experimental results with a collision-tolerant flying robot, the planner successfully considers the interplay between uncertainty and the likelihood of a collision, balances the risks of possible impacts and enables to navigate safely within highly cluttered environments.**

I. INTRODUCTION

Aerial robots and especially Micro Aerial Vehicles (MAVs) are being utilized in an ever increasing set of applications including infrastructure inspection [1, 2], exploration [3], transportation and delivery [4], or even off-world scientific discovery [5]. In all these tasks, the ability of the robot to safely navigate its environment is essential. Motivated by this fact, the research community has contributed seminal works for collision-free motion planning [6–9]. However, MAVs are often tasked to navigate within highly confined, obstacle-filled environments and operate subject to conditions of severe perceptual degradation and lack of GPS which –in their combination– tend to increase their localization and mapping uncertainty [10–17]. In such conditions, confidently identifying a collision-free path is far from trivial –even when one exists– especially due to the effects of uncertainty [10]. Collision-tolerant flying robots have been developed to mitigate some of the risks and have been successfully utilized in industrial applications through teleoperation. A niche community has focused on the problem of motion planning that accounts for collisions [18–22] with some of the works focusing on flying systems.

Responding to the above facts and challenges, in this work we contribute a novel risk-aware motion planning algorithm that provides a unified treatment to the threats and hazards posed to a collision-tolerant aerial robot due to unavoidable localization uncertainty and the impact of possible collisions in highly cluttered environments. By allowing the set of safe and admissible solutions to contain paths that present collisions –within a limit of kinetic energy at impact– the planner can balance the risks posed to the robot when selecting a path subject to uncertainty and offer the ability to fly fast –but safely– in challenging densely obstacle-filled environments such as industrial facilities, underground networks, vessel ballast tanks or forests. The provided motion plans account for the dynamics of MAVs, while to enhance the robustness of the solution –especially in initially unknown environments– the method operates in a receding horizon fashion by identifying a path that involves multiple motion steps but executing only the first of those and then iteratively updating its solution. To verify the proposed strategy, a comprehensive set of metric-driven simulation studies are presented, alongside an experiment using the collision-tolerant RMF-Owl [23] MAV developed for the DARPA Subterranean Challenge [24] and depicted in Figure 1.

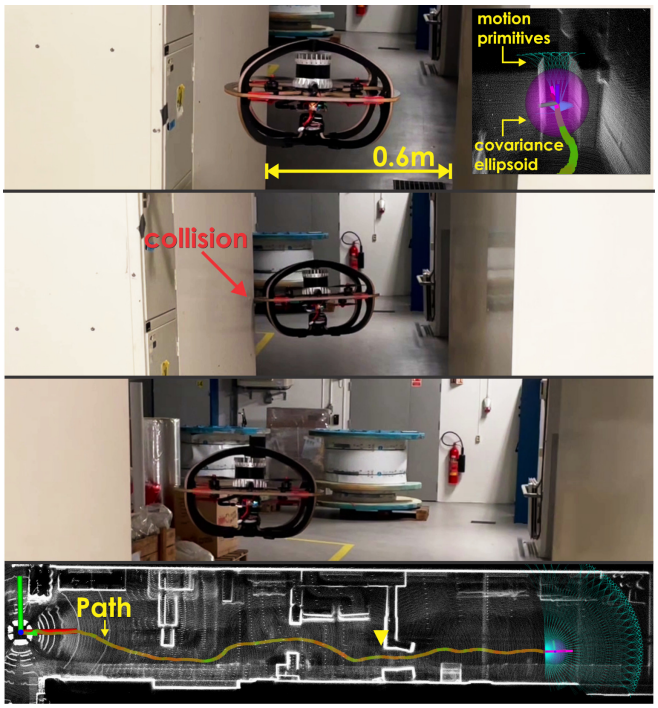

Fig. 1. Instance of the proposed risk-aware motion planner for collision-tolerant aerial robots subject to uncertainty.

The remainder of this paper is organized as follows: Section II presents related work, followed by the problem statement in Section III. The proposed risk-aware motion planner is presented in Section IV. Evaluation studies are detailed in Section V, followed by conclusions in Section VI.

This material was supported by the Research Council of Norway under project SENTIENT (grant number 321435).

The authors are with the Autonomous Robots Lab, Norwegian University of Science and Technology (NTNU), O. S. Bragstads Plass 2D, 7034, Trondheim, Norway paolo.de.petris@ntnu.no

II. RELATED WORK

The proposed contribution has relevance to the domains of uncertainty-aware motion planning, as well as the niche domain of collision-aware path and trajectory generation. To account for uncertainty in path generation, the domain of "belief–space" planning has emerged [10–17]. The authors in [10] contribute the Rapidly-exploring Random Belief Trees method on motion planning under uncertainty. To allow for computational tractability, they restrict the motion plan to a nominal trajectory stabilized using a linear estimator and controller. Building on top, the work in [13] evaluates offline multiple paths in a known map to choose the one with the motion required to support estimation. The authors in [14] present the method of Batch Belief Trees that visit regions that offer information to reduce the system's uncertainty. Exploiting photometric information, the work in [12] uses information for the scene's visual appearance to reduce the pose uncertainty over planned trajectories. Our previous work in [15] focused on receding horizon localization uncertainty-aware exploration path planning on volumetric maps considering a robot running visual-inertial odometry. Exploiting Model Predictive Control, the work in [25] unifies control and planning with respect to action and perception objectives.

The idea of exploring or even exploiting collisions in robot path planning has been researched by a niche community [18–22]. Relying on the principles of mixed integer programming, the works in [19, 20] investigate a framework that allows for planned trajectories to be inclusive of collisions. The method incorporates a hybrid model to represent collision dynamics as constraints. In [21] the authors focus on harnessing the possible benefits of collisions-aware trajectory optimization for spacecrafts, and extend and experimentally verify their previous contributions to demonstrate the potential benefits (e.g., in power limiting) of collision-inclusive planning. Contributing a new optimal steering paradigm, the authors in [22] demonstrate that collisions can be beneficial events as contact may help, for example, to possibly reduce the state estimation uncertainty. The method is validated with the use of an omni-directional wheeled robot. Considering aerial robots, and specifically one with tensegrity structure, the work in [18] exploits collisions for sampling-based motion planning and demonstrates that collisions-inclusive paths can have benefits in confined geometries.

Compared to the previous body of work, our contribution a) combines information on uncertainty and collision-tolerance in a unified risk-aware planning paradigm, b) embeds collision-inclusive paths in a versatile planning framework exploiting an efficient volumetric representation that does not make simplifying assumptions for the geometry of the environment and does not assume any prior knowledge, c) accounts for the robot dynamics, d) enhances robustness through a receding horizon policy, and e) is extensively verified through both simulations and experimental studies with a collision-tolerant flying robot operating completely autonomously in obstacle-filled GPS-denied environments.

III. PROBLEM FORMULATION

The risk-aware motion planning problem, as considered in this work, is that of identifying safe paths that allow a collision-tolerant aerial robot, specifically a multirotor MAV, to navigate towards a reachable destination within an initially unknown map, while accounting for localization uncertainty, and the impact of potential collisions with the environment.

Let $\mathbb{M}$ be a 3D occupancy map of the environment which is iteratively explored as the robot navigates using an onboard depth sensor $\mathbb{S}$ with horizontal and vertical Field Of View (FOV) $[F_H, F_V]^\circ$ and maximum effective range $d_{\max}$. The map consists of voxels m that belong to three categories, namely a) mapped and free, b) mapped but occupied, and c) unknown. At every iteration k, the set of voxels being free or occupied, represent the known subset of the map $\mathbb{M}_K^k$, while the rest is the unknown subset $\mathbb{M}_U^k = \mathbb{M}^k - \mathbb{M}_K^k$, where $\mathbb{M}^k$ is the map at its current status. Considering a holonomic vehicle, let us define the reachable set of position configurations $\mathcal{R}$, i.e., the set of coordinates that the robot can reach in the map. At every time instance t of its motion, the robot is subject to localization uncertainty modeled using a multivariate Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}_t, \mathbf{\Sigma}_t)$. As collisions lead to significant forces applied to the robot's frame, a known (e.g., from prior experimental or simulation studies) maximum kinetic energy $E_K^{\max}$ with which an impact may safely take place is considered. Given the above, the overall risk-aware motion planning problem for collision-tolerant aerial robots under uncertainty can be cast globally as:

Problem 1 (Risk-Aware Motion Planning Problem) Given an initial position configuration $\boldsymbol{\xi}_0 = [x_0, y_0, z_0]$ and a reference configuration $\boldsymbol{\xi}_r = [x_r, y_r, z_r]^T \in \mathcal{R}$, find an admissible safe path $\boldsymbol{\chi}$ that reaches this destination, while minimizing the risks imposed by possible collisions - especially as those arise due to the underlying localization uncertainty of the system as captured by a Gaussian process. The solution must further account and optimize for the dynamics of the system belonging to the class of multirotor MAVs. An admissible safe path is one that either experiences no collisions or experiences collisions with the robot flying at speeds such that the kinetic energy of the impact is below $E_K^{\max}$.
It is noted that although the problem is here defined globally, its solution is necessarily derived iteratively as the map of the environment is initially considered to be unknown and gets explored as the robot navigates and utilizes the onboard depth sensor $\mathbb{S}$ alongside localization information. In any iteration, solutions are planned within the known subset of $\mathbb{M}^k$.

IV. PROPOSED APPROACH

This work contributes a new Risk-Aware Motion planner (RAMPlanner) tailored to collision-tolerant MAVs that are subject to localization uncertainty. The method offers a unified treatment of the risks posed by uncertainty and by the impact of a collision to the flying robot allowing for collision-inclusive paths that are still safe but more efficient in their ability to negotiate highly confined environments.

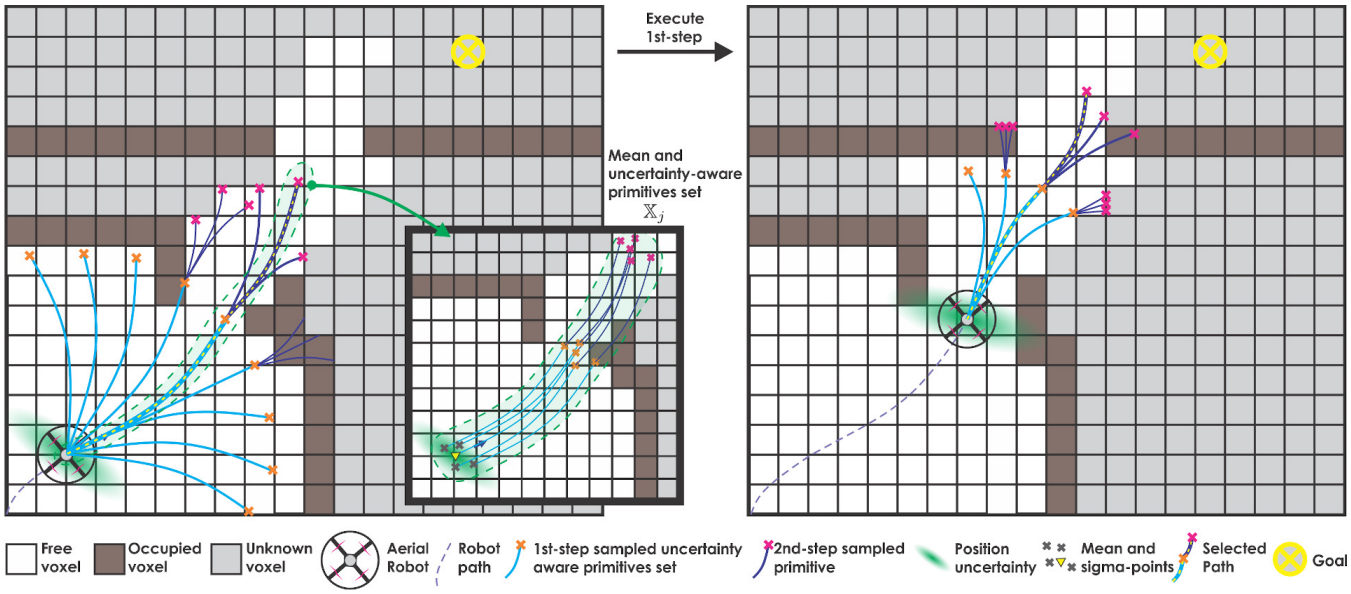

Fig. 2. Illustration of key steps of the proposed risk-aware motion planner for collision-tolerant aerial robots subject to localization uncertainty. The method samples a set of sets of motion primitives over a prediction horizon (here $N = 2$), while for the first step of each motion primitive set, $\mathbb{X}_j$, it accounts for the position uncertainty of the robot by sampling $2L$ sigma-points additional to the mean value (here $L = 2$ as this is a 2D visualization but $L = 3$ in the method). The algorithm balances the importance of moving towards the destination and the impact of possible collisions, which may be likely given the uncertainty, and selects the best path. Only the first step of that path is executed by the robot, while the whole procedure is then repeated in a receding horizon fashion. In this depiction few paths are shown, and certain steps are not continued over the horizon in order to allow for visual clarity.

A. Environment Representation

The environment is represented as a 3D volumetric occupancy map $\mathbb{M}$ using the open-sourced VDB-EDT method [26] implementing an efficient Euclidean Distance Transform (EDT) through the VDB data structure [27]. VDB is a memory-efficient, hierarchical data structure allowing to represent sparse, time-varying volumetric maps that are discretized over a 3D grid. Named after its close relation to the dynamic volumetric $B+$ tree, the method utilizes spatial coherence of time-varying data and permits to encode data values and grid topology in a compact and memory-efficient fashion. Importantly for robotics, the method does not impose any constraint on the sparsity of the volumetric data and provides fast, on average $O(1)$, random and sequential access patterns during insertion, retrieval, and deletion operations. VDB-EDT runs on the grid map and generates the associated distance field encoding the distance value against the obstacles of the environment, thus facilitating motion planning. The strategy is particularly efficient, both in memory needs, and in terms of computation time for updating the map and it is thus suited for computationally constrained robotic platforms. As the robot navigates in the initially unknown environment, more of it gets explored and at every planning iteration k its status is reflected in $\mathbb{M}_k$. In this work a voxel size of $0.2m$ is used.

B. Receding Horizon Motion Primitives Planning

The proposed RAMPlanner, outlined in Figure 2, performs planning based on motion primitives that respect the dynamics of multirotor MAVs and operates in a receding horizon fashion towards enhanced robustness against the imperfections introduced by the partial knowledge of the map, as well as the robot's localization uncertainty. Using the 3D position as the planning state $\boldsymbol{\xi} = [x, y, z]^T$, at every iteration the method builds a library of motion primitives by sampling a set of waypoints $\{\boldsymbol{\xi}\}^j$ within a "frustrum" with horizontal and vertical parameters $[P_H, P_V]^\circ$ for each primitive $\boldsymbol{\chi}^j$. More specifically, the planning frustrum is discretized with a fixed resolution (s_H, s_V) over the horizontal and vertical axes respectively and for every point in this discretization, the method samples N points in the $\boldsymbol{\xi}$-space with each being a tunable distance d_s apart. To probabilistically enhance the ability of the algorithm to find admissible solutions, an additional population of N-point sequences is also derived with each of these N-points being a random configuration in the $\boldsymbol{\xi}$-space. Subsequently, each motion primitive, based on fixed discretization or random sampling, is generated by calculating a minimum snap trajectory passing through these waypoints, with the current configuration of the robot $\boldsymbol{\eta}_k = [x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k]^T$ as the initial condition. Each dimension $d \in \{x, y, z\}$ of the motion primitive $\boldsymbol{\chi}^j$ is represented by a sequence of N 7^{th} order polynomials

$$\{P_n^{j,d}(t) = \sum_{l=0}^{7} c_n^{l,d} t^l\}, \; n = 0...N-1 \tag{1}$$

where $c_n^{l,d}$ is the coefficient of t^l for dimension d. The proposed method derives paths with N steps but only the first step of the selected motion primitive will be executed by the robot thus giving rise to a receding horizon strategy.

The numerical calculation of these minimum snap trajectories is based on the open-sourced work of [28] on fast minimum snap trajectory generation using double descriptions of polynomials. Building upon the success of minimum snap

trajectories for quadrotor MAVs [29], the method considers a set of waypoints and the planned times assigned to each of them and offers minimum snap trajectory solutions with linear-complexity. The algorithm is extremely efficient which is key to the ability of our approach to derive a large number of motion primitives over a horizon. We derive the planned time to each waypoint based on set maximum values for the allowable velocity $v^{\max}$ and acceleration $a^{\max}$. Furthermore $v^{\max}$ is adjusted in each planning iteration based on the number of motion primitives experiencing unsafe collisions in the previous iterations as explained in Section IV-F.

C. Accounting for Uncertainty

Operating in unknown and GPS-denied environments requires the robot to estimate its state onboard which is susceptible to drift over time. Flying in cluttered and confined areas may further increase the uncertainty in the localization due to obstructed visibility of the onboard sensors. Furthermore, aerial robots are often tasked to operate in conditions of perceptual degradation [30] which increases the likelihood of localization drift. Simultaneously, inaccurate localization results in poor trajectory tracking. Hence, it is important to account for the localization uncertainty in the motion planning problem. However, this is a particularly demanding task and the complexity of the underlying partially observable markov decision process is so high that a solution to the full problem is practically intractable [31, 32]. Focusing on the effect of uncertainty in increasing the risk of a collision and emphasizing a computationally-efficient approach, in this work we consider the uncertainty only in the current position estimate $\hat{\boldsymbol{\xi}}^k$, which we model as a Gaussian distribution $\mathcal{N}(\hat{\boldsymbol{\xi}}^k, \boldsymbol{\Sigma}^k_{\xi\xi})$, derived from $\mathcal{N}(\hat{\mathbf{x}}_t, \boldsymbol{\Sigma}_t)$ at time $t = t_k$, and investigate the effect of the uncertain robot state when a new planning iteration starts. Since the planning $\boldsymbol{\xi}$-space only includes the robot position, this approximation is considered as a good choice for the risk-aware planning problem considered.

Based on this model for the robot position state and uncertainty, we can evaluate a candidate motion primitive $\boldsymbol{\chi}^j, j = 1...\mathrm{n_{prim}}$ and account for the possible collisions resulting from it in an uncertainty-aware manner. Specifically, a set of points are sampled from the covariance, including the mean estimate of the position, and the possible trajectories are calculated with these points as the starting state. To sample the covariance over the planning state $\boldsymbol{\xi}$ we derive a set of $2L + 1$ sigma-points computed from the density $\mathcal{N}(\hat{\boldsymbol{\xi}}^k, \boldsymbol{\Sigma}^k_{\xi\xi})$ according to

$$\boldsymbol{\xi}^S_0 = \hat{\boldsymbol{\xi}}^k \tag{2}$$

$$\boldsymbol{\xi}^S_i = \hat{\boldsymbol{\xi}}^k + \sqrt{L\boldsymbol{\Sigma}^k_{\xi\xi}} \;\; (0 < i \leq L) \tag{3}$$

$$\boldsymbol{\xi}^S_i = \hat{\boldsymbol{\xi}}^k - \sqrt{L\boldsymbol{\Sigma}^k_{\xi\xi}} \;\; (L < i \leq 2L) \tag{4}$$

where $L = \dim(\hat{\boldsymbol{\xi}}^k)$ and the sampled points allow to numerically reconstruct $\mathcal{N}(\hat{\boldsymbol{\xi}}, \boldsymbol{\Sigma}_{\xi\xi})$ [33]. Please note that for the simplicity of notation, from now on we focus only to the current iteration and drop the k superscript.

Given the sigma-points $\boldsymbol{\xi}^S_i, i = 0...2L$, the possible trajectories $\{\boldsymbol{\chi}_j^{\boldsymbol{\xi}^S_i}\}$ (hereafter referred to as the uncertain primitives) starting from the sigma-points are computed and associated with each primitive $\boldsymbol{\chi}_j^{\boldsymbol{\xi}^S_0}$ (hereafter referred to as the mean primitive). As the sigma-points only contain the position, the velocity and the higher order derivatives for the uncertain primitives will remain the same as the mean primitive. Hence, only the coefficients for the constant terms in the polynomial formulation of the uncertain primitives will differ from those of the mean primitive. The vector of the constant term coefficients for each state element, $\boldsymbol{c}^0_{n,i} = [c^{0,x}_{n,i}, c^{0,y}_{n,i}, c^{0,z}_{n,i}]^T$, of the n^{th}, $n = 0...N-1$, polynomial segment of the uncertain primitive $\boldsymbol{\chi}_j^{\boldsymbol{\xi}^S_i}$ corresponding to $\boldsymbol{\xi}^S_i$ is calculated as:

$$\boldsymbol{c}^0_{n,i} = \boldsymbol{c}^0_n + \boldsymbol{\xi}^S_i - \boldsymbol{\xi}^S_0 \;\; (i = 0...2L) \tag{5}$$

where, $\boldsymbol{c}^0_n$ is the coefficient vector for the mean primitive. Overall, for each mean primitive we derive a set of mean and uncertainty-aware primitives $\mathbb{X}_j = \{\boldsymbol{\chi}_j^{\boldsymbol{\xi}^S_i}\}, i = 0...2L$.

D. Accounting for Collisions

Classical motion planning methods [6] aim to avoid all collisions thus ensuring the safety of the robot. However, this approach may render navigation in highly confined and obstacle-filled environments inefficient [18]. Importantly, when localization uncertainty is taken into consideration, it may turn that a collision-free path is hard and statistically unlikely to be possible to be derived (especially in cluttered environments). However, embedding collision-tolerance into the design of a robot opens the way for a new risk-aware planning paradigm in which collisions may be treated as affordable events if the kinetic energy at the time of impact (on which the collision forces depend) is below a set threshold. Accordingly safe collision-inclusive paths can be generated that factor the combined risks raised by the uncertainty in estimation and the impact of a collision.

At a sampling iteration k and provided the sampled set of mean and uncertainty-aware motion primitive sets $\mathbb{X}_j$, the method identifies the instances within each path that it collides with the environment map $\mathbb{M}_k$. To perform this test, the robot is modeled as a box with dimensions $D_L \times D_W \times D_H$ and identifying if voxels at its corners intersect the mapped occupied map in $\mathbb{M}^k$. For each path $\boldsymbol{\chi}_j^{\boldsymbol{\xi}^S_i}$, the method first identifies the possible collision points in each step $n = 0...N-1$ of the motion primitive horizon. Then, for each step n it identifies the possible collision point with maximum velocity $v^{c,\max}_{j,i,n}$ and associates its kinetic energy $E_K^{j,i,n} = \frac{1}{2}m(v^{c,\max}_{j,i,n})^2$ with that step. If no collision is experienced in a path segment, then $E_K^{j,i,n} = 0$. As for every collision-inclusive path, its risk should be weighted according to the probability $p_{j,i}$ of the mean or uncertainty-aware starting point $\boldsymbol{\xi}^S_i$ given the distribution $\mathcal{N}(\hat{\boldsymbol{\xi}}^k, \boldsymbol{\Sigma}^k_{\xi\xi})$, the collision-cost of the j-th primitive takes the form

$$J_j^c = \sum_{i=0}^{2L} \sum_{n=0}^{N-1} w_n^c p_{j,i} \underbrace{\frac{1}{2} m (v_{j,i,n}^{c,\max})^2}_{E_K^{j,i,n}} \tag{6}$$

where $w_n^c > 0$ is a tunable weight for each step and the earlier steps of the motion primitive carry more weight with respect to a possible collision. Notably, any motion primitive j that contains collision events with velocities $v_{j,i,n}$ such that $E_K^{j,i,n} \geq E_K^{\max}$ within the first step ($n = 0$) is pruned as it represents an unsafe path involving unacceptable collisions. Also, the primitives for which after a certain point all the path is in collision, are pruned up to the point where the non-continuously colliding path ends.

E. Risk-aware Objective Formulation

Provided the non-pruned safe sampled uncertainty-aware and collision-inclusive motion primitives, as well as the defined collision cost metrics, we can formulate an overall risk-aware objective that allows a collision-tolerant aerial robot to identify an optimized path towards the reference destination that is admissible and safe within the currently known map $\mathbb{M}^k$. To that end, we first define a sub-objective reflecting how close a primitive $\boldsymbol{\chi}_j$ reaches to the reference destination $\boldsymbol{\xi}_r = [x_r, y_r, z_r]^T$ over all steps of its horizon as

$$J_j^d = \sum_{i=0}^{2L} \sum_{n=0}^{N-1} p_{j,i} \left\| \boldsymbol{\xi}_r - \boldsymbol{\xi}_{j,i}^n \right\|^2 \tag{7}$$

where $\boldsymbol{\xi}_{j,i}^n$ is the end point of the n-th step of the mean ($i = 0$) or uncertain primitives ($i = 1...2L$) of the j-th primitive.

Then we can define a collective objective that accounts for both the extent to which a motion primitive optimizes the goal of reaching a destination and eliminates or reduces the impact of a collision as

$$J_j = w_d J_j^d + w_c J_j^c \tag{8}$$

where $w_d, w_c > 0$ are tunable weights that balance the importance of moving towards the desired destination and employing as safe as possible paths, within the subset of motion primitives that do not violate the constraint on the maximum kinetic energy of an affordable collision. Importantly, beyond tuning w_d, w_c the method implements a prioritization policy that allows paths that experience no collision to be selected with preference. In particular, all the cost values J_j^d are normalized and before optimizing J_j as defined above, the method optimizes for a simplified $J_i = J_j^d$ as long as there are collision-free paths which present J_j^d values no larger than the smallest $\lambda\%$ of the overall population of safe collision-free or collision-inclusive primitives.

F. Risk-aware Planning Algorithm

The proposed RAMPlanner operates in a receding horizon fashion to find uncertainty-aware and collisions-inclusive safe paths that move towards a reachable destination while respecting the dynamics of the robot and a set maximum on the kinetic energy of a collision so that it is rendered as an acceptable event. Algorithm 1 outlines all the involved steps per iteration. Finally, it is noted that an adaptive policy with respect to the maximum velocity of a path segment is also implemented. Specifically, for all the set of motion primitives per iteration, the method finds the percentage β of those that present collisions. It then uses a small sliding window of r such values β and derives an average value $\tilde{\beta}$ based on which it then reduces the maximum velocity $v^{\max}$ as $\tilde{\beta}$ increases.

Algorithm 1 RAMPlanner

$\boldsymbol{\xi}_r \leftarrow \mathbf{getGoalState}()$
$k \leftarrow 0$
$\boldsymbol{\xi}_k \leftarrow \mathbf{getCurrentPosition}()$
while $||\boldsymbol{\xi}_r - \boldsymbol{\xi}_k|| > \delta_{th}$ **do**
  $\mathbf{X} \leftarrow \mathbf{generateMotionPrimitives}()$
  $\hat{\boldsymbol{\xi}}^k, \boldsymbol{\Sigma}_{\xi\xi}^k \leftarrow \mathbf{getMeanPositionAndCovariance}()$
  $\boldsymbol{\Xi} \leftarrow \mathbf{sampleSigmaPoints}(\hat{\boldsymbol{\xi}}^k, \boldsymbol{\Sigma}_{\xi\xi}^k)$
  $J_{best} \leftarrow \infty, \boldsymbol{\chi}_{best} \leftarrow \emptyset$
  for all $\boldsymbol{\chi}_j \in \mathbf{X}$ **do**
    $\mathbb{X}_j \leftarrow \mathbf{calculateUncertainPrimitives}(\boldsymbol{\Xi})$
    $\{v_{j,n}^{c,\max}\} \leftarrow \mathbf{collisionVelocities}(\boldsymbol{\chi}_j)$
    $E_K^{j,0} \leftarrow \frac{1}{2} m (v_{j,0}^{c,\max})^2$
    if $E_K^{j,0} > E_K^{max}$ **then**
      continue
    for all $\boldsymbol{\chi}_j^{\boldsymbol{\xi}_i^S} \in \mathbb{X}_j$ **do**
      $\{v_{j,i,n}^{c,\max}\} \leftarrow \mathbf{collisionVelocities}(\boldsymbol{\chi}_j^{\boldsymbol{\xi}_i^S})$
    $J_j^c \leftarrow \mathbf{computeCollisionCost}()$
    $J_j^d \leftarrow \mathbf{computeDistanceCost}(\boldsymbol{\xi}_r, \mathbb{X}_j)$
    $J_j \leftarrow \mathbf{computeObjectiveFunction}(J_j^d, J_j^c)$
    if $J_j < J_{best}$ **then**
      $J_{best} \leftarrow J_j$
      $\boldsymbol{\chi}_{best} \leftarrow \boldsymbol{\chi}_j$
  $\mathbf{executeFirstStep}(\boldsymbol{\chi}_{best})$
  $\boldsymbol{\xi}_k \leftarrow \mathbf{getCurrentPosition}()$
  $k \leftarrow k + 1$

V. EVALUATION STUDIES

In order to evaluate the performance of the proposed RAMPlanner, we present a set of simulation studies as well as an experimental evaluation in an indoor environment with varying obstacle density.

A. Simulation Studies

We evaluate the performance of the planner in two simulation tests, first (Simulation1) consisting of an obstacle filled environment with increasing density of obstacles, and the second (Simulation2) requiring the robot to pass through a corridor with circular openings of decreasing size. All the simulations were done in the RotorS [34] simulator using a model of the RMF-Owl [23] aerial robot carrying an OUSTER OS0-64 LiDAR sensor having FOV $[360, 90]^\circ$ and $d_{\max} = 50$m. The modeled robot dimensions are $38 \times 38 \times 24$cm$(L \times W \times H)$, perfectly matching the real one.

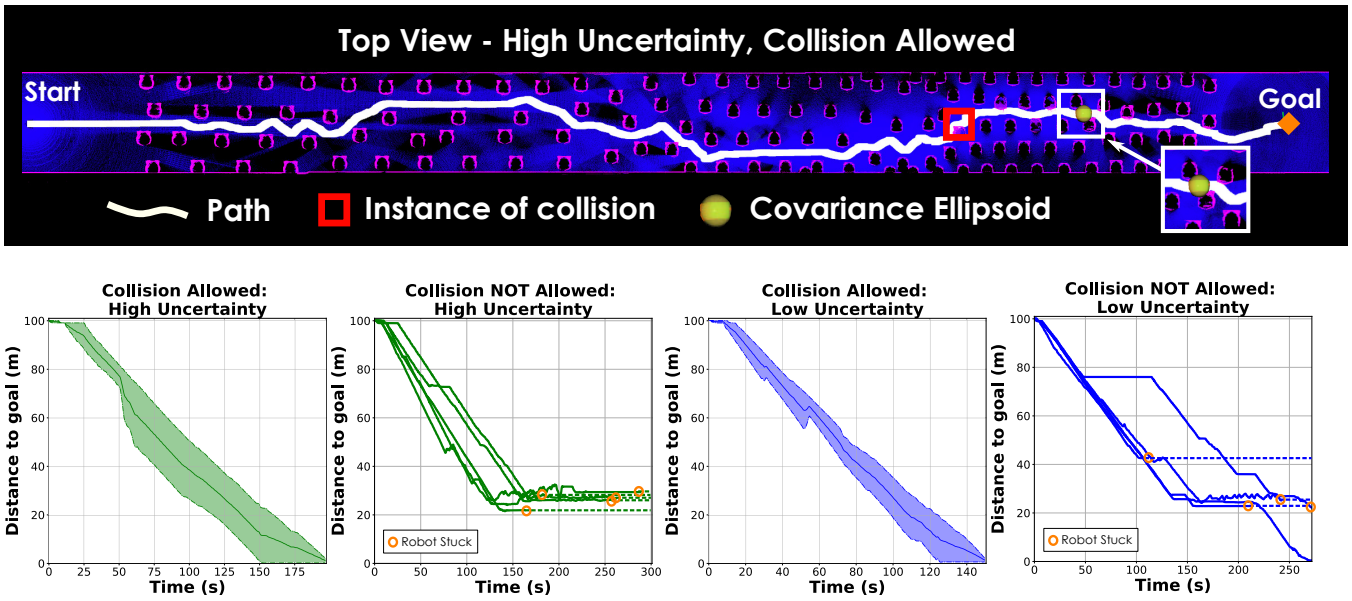

Fig. 3. This figure shows a simulation study in randomly generated environments of increasing density of obstacles. The RAMPlanner was tested with two configurations, namely a) allowing collisions within limits and b) a "naive" version allowing only completely collision-free navigation, as well as with two different levels of localization uncertainties ($\mathbf{\Sigma}_{\xi\xi}^{High} = \text{diag}(0.03, 0.03, 0.03)$, $\mathbf{\Sigma}_{\xi\xi}^{Low} = \text{diag}(0.0003, 0.0003, 0.0003)$). The plots show the remaining distance to the goal as a function of time over 5 missions in each of the four cases. It is evident that without allowing (safe) collisions-inclusive paths the robot did not reach the goal when considering high localization uncertainty, and only reached once under low uncertainty. The map on the top shows one example with the environment along with the path followed by the robot for the case of high localization uncertainty and allowing collisions. The 3σ covariance ellipsoid is also shown.

Parameter	Simulation1	Simulation2
$[P_H, P_V]$	$[360, 40]^\circ$	$[360, 60]^\circ$
$s_H[n_0, n_1]$	$[10, 10]^\circ$	$[10, 10]^\circ$
$s_V[n_0, n_1]$	$[10, 10]^\circ$	$[10, 10]^\circ$
$v^{\max}$	2.0 m/s	1.0 m/s
$a^{\max}$	3.0 m/s^2	3.0 m/s^2
$v_{col}^{\max}$	1.0 m/s	1.0 m/s
n_{prim}	1080	1512
t_p	220 ms	400 ms
$\ell_n[n_0, n_1]$	$[1.5, 2.5]$ m	$[1.2, 3.0]$ m
w_d, w_c	0.7, 0.5	0.9, 0.8

TABLE I

RAMPLANNER PARAMETERS USED IN SIMULATION.

The first simulation study evaluates the performance of the RAMPlanner in the presence of varying obstacle density and two different levels of localization uncertainty. The environment consists of three regions with increasing obstacle density (average free space between the obstacles being 2.0m, 1.4m, and 0.8m). The robot starts outside the least dense region and is commanded to go through the obstacle field to reach a goal on the other side. The performance of the planner is evaluated in two configurations: a) allowing collisions within safety limits and b) completely collision-free navigation. Both the configurations were tested in 5 runs each for two different levels of localization uncertainty, totalling of 10 runs each. The results of these evaluations are presented in Figure 3. As shown, without allowing safe collisions the robot is unable to reach the goal under high localization uncertainty and only reaches once under low uncertainty condition. The subfigure at the top illustrates the environment in one of the missions showing the case of high localization uncertainty with safe collisions permitted.

In the second simulation study the robot is commanded

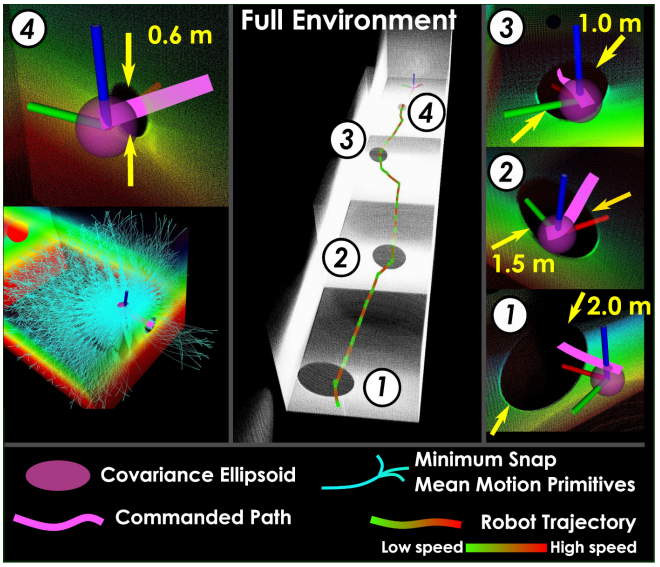

Fig. 4. Instances of the simulation study where the aerial robot is commanded to reach a goal that requires it to pass through four circular openings of decreasing diameter, namely 1) 2.0 m, 2) 1.5 m, 3) 1.0 m and 4) 0.6 m. The last three dimensions, with different shape, are recreated in the real environment.

to reach a goal that requires it to pass through four circular openings of decreasing diameter ranging from 2m to 0.6m. Due to the size of the robot and having introduced uncertainty in the localization, the planner has to chose a path such that at least one of the uncertain primitives or the mean primitive is in collision within the allowed limits. Figure 4 demonstrates the ability of the planner to successfully navigate through all four openings. The paths planned by the robot, generated motion primitives, and the

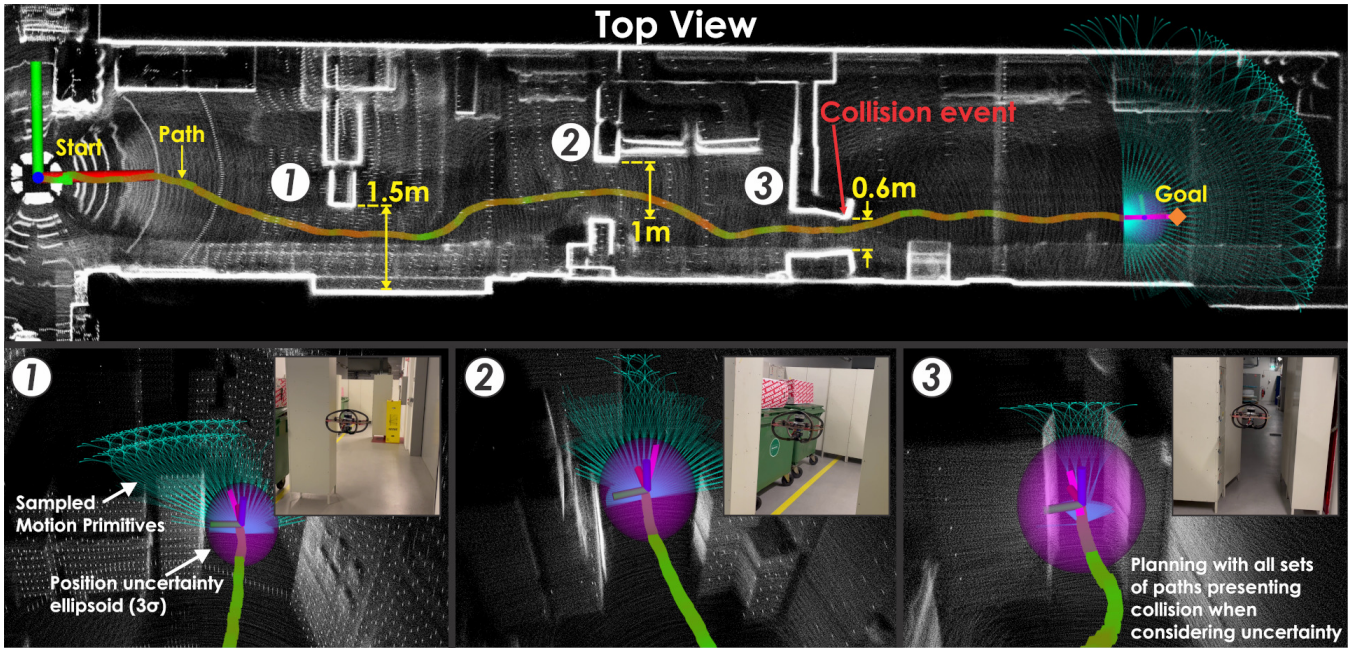

Fig. 5. Instances of autonomous traversal towards a reachable destination in an obstacle-filled corridor. The dimensions of the depicted passages are roughly 4, 2.5 and 1.5 times the size of the RMF-Owl robot $(1.5, 1.0, 0.6\text{m})$. The color of the path illustrates the traversed speed of the robot, red color representing faster speed while green color corresponds to lower speed. The robot follows planned collision-free paths to pass the first and second passages, slowing down when it goes through the narrower second gap. When RMF-Owl reaches the third gap, all the generated primitives are in collision when accounting for the localization uncertainty, thus the robot must accept a collision to be able to reach the final goal.

3σ covariance ellipsoid can also be seen in the figure.

Table I summarizes various parameters used in the two simulations. These include the parameters used for the generation of the motion primitives, tunable weights in Equation 8, maximum allowed velocity and acceleration $v^{\max}, a^{\max}$, maximum allowed safe collision velocity $v_{col}^{\max}$, number of motion primitives generated $\mathrm{n_{prim}}$, the average computation time t_p, and the step sizes (ℓ_n) of the two step (n_0, n_1) motion primitives used in the simulations.

B. Experimental Evaluation

To further evaluate the proposed risk-aware motion planner, an experiment was conducted using the RMF-Owl [23] aerial robot. The RMF-Owl is an MAV with a rigid collision-tolerant airframe fully built out of carbon-foam sandwich material. It integrates a PX4-based autopilot for attitude and thrust control, and integrates a Khadas VIM3 Single Board Computer incorporating $\times 4$ 2.2GHz Cortex-A73 cores, paired with $\times 2$ 1.8GHz Cortex-A53 cores, alongside a Neural Processing Unit (NPU). The robot's primary exteroceptive sensor is an OUSTER OS0-64 LiDAR with FOV $[360, 90]^\circ$ and $d_{\max}$ 50 m. The system's total weight is 1.4 kg, its dimensions are $38 \times 38 \times 24$ cm$(L \times W \times H)$, and its endurance slightly above 10 min. The position controller, LiDAR-based localization and mapping using the work in [35], as well as the overall pipeline of the proposed RAMPlanner run onboard the Khadas SBC. Accordingly, the computational efficiency of the method is particularly important for small aerial robots in both industrial and natural environments. In the proposed experiment the robot is commanded to reach a setpoint at the end of an obstacle-filled corridor in which three increasingly narrower passages have been placed. As depicted in Figure 5 the dimensions of these passages are roughly 4, 2.5 and 1.5 times the size of the RMF-Owl respectively $(1.5, 1.0, 0.6\text{m})$. After takeoff, the planner is triggered and the robot safely traverses the first of the three passages. While approaching the second gap the travelling speed is reduced as described in Section IV-F and increased again before the robot reaches the final gap. In this case all the generated paths result in collision when considering uncertainty and the robot must accept a possibly colliding path to be able to reach the goal location. Note that the covariance matrix provided by the onboard localization solution was inflated 3000 times to replicate a scenario with high localization uncertainty. For this experiment we use two step primitives generated using $[P_H, P_V] = [180, 5]^\circ$, with $s_H = s_V = 5^\circ$ for the first step and 8° for the second step, for a total number of 888 possible motions. The values used for w_d and w_c are 0.7 and 0.5 and the average computational cost on Khadas is 350ms.

VI. CONCLUSIONS

This work contributes a novel risk-aware planning method for flying robots subject to localization uncertainty. Motion primitives based on minimum-snap trajectories are evaluated based on the goal-reaching and collision costs of the mean and uncertain primitives, allowing the robot to traverse safely when collision-free paths are available and simultaneously being able to go through narrow environments where collisions with limited kinetic energy are unavoidable and tolerable with the collision-tolerance design. Extensive simulation and experimental results demonstrate the performance of the planner to navigate safely in cluttered environments with the uncertainty in position estimate.

REFERENCES

[1] A. Bircher *et al.*, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 6423–6430. [Online]. Available: https://github.com/ethz-asl/StructuralInspectionPlanner

[2] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[3] S. Nuske *et al.*, "Autonomous exploration and motion planning for an unmanned aerial vehicle navigating rivers," *Journal of Field Robotics*, vol. 32, no. 8, pp. 1141–1162, 2015.

[4] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70–85, 2016.

[5] J. Balaram, M. Aung, and M. P. Golombek, "The ingenuity helicopter on the perseverance rover," *Space Science Reviews*, vol. 217, no. 4, pp. 1–11, 2021.

[6] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[7] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[8] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.

[9] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.

[10] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 723–730.

[11] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.

[12] G. Costante, J. Delmerico, M. Werlberger, P. Valigi, and D. Scaramuzza, "Exploiting photometric information for planning under uncertainty," in *Robotics research*. Springer, 2018, pp. 107–124.

[13] M. W. Achtelik, S. Lynen, S. Weiss, M. Chli, and R. Siegwart, "Motion-and uncertainty-aware path planning for micro aerial vehicles," *Journal of Field Robotics*, vol. 31, no. 4, pp. 676–698, 2014.

[14] D. Zheng and P. Tsiotras, "Batch belief trees for motion planning under uncertainty," *arXiv preprint arXiv:2110.00173*, 2021.

[15] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty-aware receding horizon exploration and mapping using aerial robots," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 4568–4575.

[16] R. P. de Figueiredo, J. le Fevre Sejersen, J. G. Hansen, M. Brandão, and E. Kayacan, "Real-time volumetric-semantic exploration and mapping: An uncertainty-aware approach," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9064–9070.

[17] A. Agha *et al.*, "Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge," *arXiv preprint arXiv:2103.11470*, 2021.

[18] J. Zha and M. W. Mueller, "Exploiting collisions for sampling-based multicopter motion planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7943–7949.

[19] M. L. Mote, J.-P. Afman, and E. Feron, "A framework for collision-tolerant optimal trajectory planning of autonomous vehicles," *arXiv preprint arXiv:1611.07608*, 2016.

[20] M. Mote, J. P. Afman, and E. Feron, "Robotic trajectory planning through collisional interaction," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 1144–1149.

[21] M. Mote, M. Egerstedt, E. Feron, A. Bylard, and M. Pavone, "Collision-inclusive trajectory optimization for free-flying spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 7, pp. 1247–1258, 2020.

[22] Z. Lu and K. Karydis, "Optimal steering of stochastic mobile robots that undergo collisions with their environment," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 668–675.

[23] P. De Petris *et al.*, "Rmf-owl: A collision-tolerant flying robot for autonomous subterranean exploration," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 527–584.

[24] Defense Advanced Research Projects Agency (DARPA), "DARPA Subterranean Challenge." [Online]. Available: https://subtchallenge.com/

[25] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "Pampc: Perception-aware model predictive control for quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.

[26] D. Zhu, C. Wang, W. Wang, R. Garg, S. Scherer, and M. Q.-H. Meng, "Vdb-edt: An efficient euclidean distance transform algorithm based on vdb data structure," *arXiv preprint arXiv:2105.04419*, 2021.

[27] K. Museth, "Vdb: High-resolution sparse volumes with dynamic topology," *ACM transactions on graphics (TOG)*, vol. 32, no. 3, pp. 1–22, 2013.

[28] Z. Wang, H. Ye, C. Xu, and F. Gao, "Generating large-scale trajectories efficiently using double descriptions of polynomials," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7436–7442.

[29] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 2520–2525.

[30] M. Tranzatto *et al.*, "Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge," *Field Robotics*, 2021.

[31] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[32] D. Zheng, J. Ridderhof, P. Tsiotras, and A.-a. Agha-mohammadi, "Belief space planning: A covariance steering approach," *arXiv preprint arXiv:2105.11092*, 2021.

[33] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of 1995 American Control Conference-ACC'95*, vol. 3. IEEE, 1995, pp. 1628–1632.

[34] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors-a modular gazebo mav simulator framework," in *Robot Operating System (ROS)*. Springer, 2016, pp. 595–625.

[35] S. Khattak, H. Nguyen, F. Mascarich, T. Dang, and K. Alexis, "Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1024–1029.