IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Safe Low-Altitude Navigation in Steep Terrain With Fixed-Wing Aerial Vehicles

Jaeyoung Lim, *Graduate Student Member, IEEE*, Florian Achermann, *Member, IEEE*, Rik Girod, Nicholas Lawrance, *Member, IEEE*, and Roland Siegwart, *Fellow, IEEE*

***Abstract*—Fixed-wing aerial vehicles provide an efficient way to navigate long distances or cover large areas for environmental monitoring applications. By design, they also require large open spaces due to limited maneuverability. However, strict regulatory and safety altitude limits constrain the available space. Especially in complex, confined, or steep terrain, ensuring the vehicle does not enter an inevitable collision state (ICS) can be challenging. In this letter, we propose a strategy to find safe paths that do not enter an ICS while navigating within tight altitude constraints. The method uses periodic paths to efficiently classify ICSs. A sampling-based planner creates collision-free and kinematically feasible paths that begin and end in safe periodic (circular) paths. We show that, in realistic terrain, using circular periodic paths can simplify the goal selection process by making it yaw agnostic and constraining yaw. We demonstrate our approach by dynamically planning safe paths in real-time while navigating steep terrain on a flight test in complex alpine terrain.**

***Index Terms*—Aerial systems: Perception and autonomy, field robots, motion and path planning.**

I. Introduction

SMALL uncrewed aerial systems (sUASs) have become a crucial tool for information-gathering applications such as search and rescue [1], mapping and inspection [2], and environmental monitoring [3], [4], [5], [6]. In particular, fixed-wing and hybrid vertical takeoff and landing (VTOL) type vehicles are popular due to their aerodynamic efficiency, resulting in long endurance and extensive area coverage.

Operating fixed-wing vehicles near terrain would enable close-up information-gathering tasks traditionally performed by less efficient multi-rotor vehicles, such as high-resolution near-infrared photogrammetry [6]. Further, near-surface operations may become necessary as recent regulations require sUASs to stay within 120 m from the closest point of the terrain surface [7]. Modern fixed-wing sUASs have limited online mission planning capabilities, which limit operations to large open areas. Due to the use of aerodynamic forces for maneuvering and power constraints, fixed-wing vehicles are limited by minimum turn radius and rate of climb (RoC). The limited maneuverability poses a significant challenge in preventing the vehicle from entering an inevitable collision state (ICS), which are regions of the state space where all control inputs eventually will result in a collision. ICSs are more likely to occur in confined environments, e.g. flying below 120 m in mountainous regions, with terrain steeper than the maximum RoC. An unsafe flight can be challenging for a human operator to correct since the vehicle can enter an ICS long before the event of a collision. Addressing these issues requires an autonomous mission planner that can ensure safety when operating fixed-wing vehicles near steep terrain.

Previous works have demonstrated practical sampling-based planners for fixed-wing navigation in confined environments [8], [9], [10], [11], [12]. However, these works only focus on planning collision-free paths between start and goal states without consideration of the ICS, significantly reducing their applicability for deployment in altitude-constrained real-world scenarios. This work introduces a computationally efficient evaluation of ICSs using periodic paths for fixed-wing vehicle navigation.

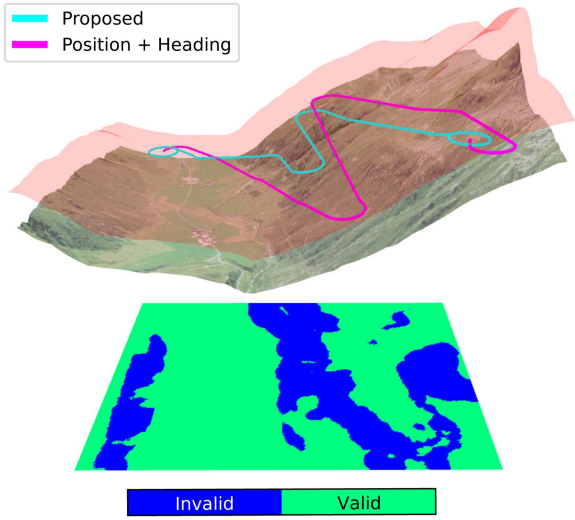


Fig. 1. Example of a planned path for a fixed-wing aerial vehicle above complex terrain with an altitude limit of 120 m above the ground (red surface). The 2D binary projection below the terrain shows the *Valid Loiter Positions* (green) proposed by this letter. Blue regions contain no kinematically feasible safe periodic states. Planned paths using the safe periodic states proposed in this letter (cyan) compared to the conventional goal position and heading (magenta) are shown.

Manuscript received 6 October 2023; accepted 31 January 2024. Date of publication 22 February 2024; date of current version 8 April 2024. This letter was recommended for publication by Associate Editor R. Zufferey and Editor G. Loianno upon evaluation of the reviewers' comments. This work was supported by ETH Research Grant AvalMapper under Grant ETH-10 20-1. *(Corresponding author: Jaeyoung Lim.)*

Jaeyoung Lim, Florian Achermann, Rik Girod, and Roland Siegwart are with Autonomous Systems Lab, ETH Zürich, 8092 Zürich, Switzerland (e-mail: jalim@ethz.ch; acfloria@ethz.ch; brik@ethz.ch; rsiegwart@ethz.ch).

Nicholas Lawrance is with Robotic Perception and Autonomy, CSIRO Data61, Eveleigh, NSW 4069, Australia (e-mail: nicholas.lawrance@csiro.au).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3368800, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3368800

Using periodic paths as terminal segments simplifies the infinite horizon collision checks required to evaluate whether a state is an ICS. We present circular periodic paths for this particular planning problem, which we pre-compute in the narrow flight corridors above the digital elevation map (DEM), as seen in Fig. 1. Our approach offloads the operator's workload of selecting a safe goal state by making the goal state yaw agnostic and by further constraining the altitude. Also, the periodicity of terminal paths allows the operator to replan the next mission without time constraints. We demonstrate our approach on a real-world flight navigating a steep valley in the Swiss Alps.

Ultimately, our work enables safe fixed-wing vehicle operation in alpine terrain. The key contributions of this letter are:

- An efficient method for evaluating ICSs with sampling-based path planning for fixed-wing navigation.
- Evaluation of planner performance on diverse terrains for real-time planning.
- A low-altitude flight demonstration in a challenging alpine environment.
- An open-source real-time Dubins RRT* planner implementation for fixed-wing sUAS.

II. Related Work

Previous works have explored practical approaches for navigating confined environments with fixed-wing vehicles [8], [9], [9], [10], [11], [12] demonstrated a fixed-wing vehicle operating in confined indoor environments using online trajectory optimization with Dubins polynomials. [11] use motion primitives to plan acrobatic maneuvers to navigate in cluttered environments. However, trajectory optimization approaches can be sensitive to disturbances, especially in the presence of wind.

Geometric path planning approaches split the problem into path planning and path following. Path planning uses smooth path representations, to represent the dynamic constraints of the vehicle such as Dubins curves [9], Bezier curves [12], and splines [10]. A guidance controller is then deployed for following the planned path. Therefore, geometric planning approaches are more robust against disturbances, as the reference path is purely geometric and is stabilized around the path with a guidance controller [13].

In this letter, we use the Dubins airplane model [14], where [9], [9] show that planning in a geometric Dubins airplane space [14] is an effective solution to incorporate the constrained climb rate and curvature for planning collision-free paths in alpine environments. These works formulate the planning problem of finding a collision-free path from start to goal. Therefore, the operator is still responsible for selecting a safe goal, i.e., not entering an ICS on arrival at the final state [15]. However, as fixed-wing vehicles are never stationary, identifying future collisions can be challenging.

Our work ensures the complete path, including the start and goal, is safe. [16] showed that a collision-free path between two safe states is always safe. Therefore, besides computing a collision-free path, our planner must verify that the start and goals are safe. Computing ICSs explicitly is challenging since infinite horizon collision checks are required [17]. We take inspiration from works that use periodic paths as terminal states to evaluate safety for emergency maneuvers [18] or robust invariant sets [19]. We use circular loiter patterns, a typical periodic flight mode for fixed-wing vehicles, to make an approximate ICS verification more efficient.

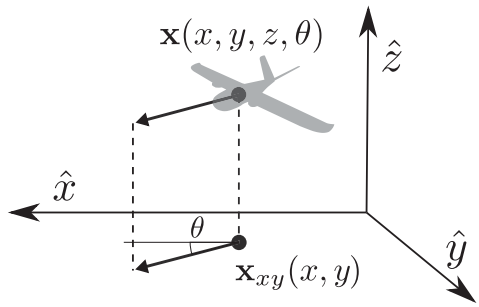


Fig. 2. State space of Dubins airplane model.

III. Problem Formulation

We consider the problem of finding the shortest path $\boldsymbol{\eta}^*(s) : \mathbb{R}^+ \to \mathcal{X}$, where s is the length along the path between any pair of elements from given sets of start and goal states $\mathcal{X}_{start}, \mathcal{X}_{goal} \subset \mathcal{X}$. The complete workspace $\mathcal{X}$ is divided into permitted or free-space states $\mathcal{X}^+ \subseteq \mathcal{X}$, and occupied or prohibited space $\mathcal{X}^- := \mathcal{X} \setminus \mathcal{X}^+$, which are in collision. The permitted workspace $\mathcal{X}^+$ is further partitioned into safe states $\mathcal{X}_{safe} \subseteq \mathcal{X}^+$ and inevitable collision states $\mathcal{X}_{ICS} := \mathcal{X}^+ \setminus \mathcal{X}_{safe}$.

The safe planning problem can be formally written as

$$\begin{aligned} \boldsymbol{\eta}^*(s) &= \text{argmin}_{\boldsymbol{\eta}} \int_0^S \frac{\partial \boldsymbol{\eta}(s)}{\partial s} ds \\ \text{s.t.} \quad & \boldsymbol{\eta}(s) \in \mathcal{X}_{safe}, && \forall s \in [0, S], \\ & \frac{\partial \boldsymbol{\eta}(s)}{\partial s} = f(\boldsymbol{\eta}(s)) && \forall s \in [0, S], \\ & \boldsymbol{\eta}(0) \in \mathcal{X}_{start}, \quad \boldsymbol{\eta}(S) \in \mathcal{X}_{goal}, && (1) \end{aligned}$$

where S denotes the path length. The complete path must only contain states in the set of safe states $\mathcal{X}_{safe}$. Also, the path should be kinematically feasible, satisfying the kinematic constraints $f(\cdot)$.

A. Dubins Airplane Model

The Dubins airplane [14] describes our kinematic model, where the aircraft state consists of a three-dimensional position and heading angle $\boldsymbol{x} = \{x, y, z, \theta\} \in \mathbb{R}^3 \times \text{SO}(2)$, as illustrated in Fig. 2. We adopt the kinematic definitions from [14] to a time-independent form based on distance along the path $\mathbf{s}$. Given a Dubins airplane path $\boldsymbol{\eta}(s)$ the path can be reparameterized as (2), where $\gamma = \tan^{-1}(\frac{\partial z}{\partial s} / \sqrt{\frac{\partial x}{\partial s}^2 + \frac{\partial y}{\partial s}^2})$ is the flight path angle, and κ is the curvature.

$$\frac{\partial \boldsymbol{\eta}(s)}{\partial s} = f(\boldsymbol{\eta}(s)) = \begin{pmatrix} \cos(\gamma)\cos(\theta) \\ \cos(\gamma)\sin(\theta) \\ \sin(\gamma) \\ \kappa\cos(\gamma) \end{pmatrix} \qquad (2)$$

The minimum and maximum flight path angle $\gamma \in [\gamma_{min}, \gamma_{max}]$ and path curvature limits $\kappa \in [-\kappa_{max}, \kappa_{max}]$ constrain the maneuverability of the Dubins airplane.

B. Map Representation

We use a multi-layered surface map [20], which consists of multiple layers of 2.5D elevation maps to represent the terrain surface and offset collision surfaces. Elevation maps are widely used in robotics [21] and geographic information systems communities [22] to represent large environments. Note that the use of elevation maps might limit the application of representing environments with complex geometry, e.g. indoor or cluttered industrial settings [20].

An elevation map $H : \mathcal{M} \mapsto \mathbb{R}$ maps a grid cell position $\boldsymbol{m} \in \mathcal{M}$ to the terrain height $h \in \mathbb{R}$, where $\mathcal{M} \subset \mathbb{R}^2$ is the set of all grid cell positions.

$$h = H(\boldsymbol{m}), \quad \boldsymbol{m} \in \mathcal{M} \tag{3}$$

The planner aims to generate a path in the band between a specified minimum and maximum distance to the closest point on the surface. We define the offset collision surface to represent these constraints.

Definition 3.1 (Offset Collision Surface): The offset collision surface $D_d(\cdot)$ is an elevation map that defines a surface above the terrain where the shortest Euclidean distance to the terrain is d for all points on the surface.

$$\begin{aligned} D_d(\boldsymbol{m}) = \ & \min z \\ s.t. \ \ & \forall \boldsymbol{m}_j \in \mathcal{M} \ \ \sqrt{(\boldsymbol{m}-\boldsymbol{m}_j)^2 + (z - H(\boldsymbol{m}_j))^2} \geq d, \\ & z > H(\boldsymbol{m}_j) \end{aligned} \tag{4}$$

We denote the offset collision surface with minimum distance as D^- and maximum distance as D^+. A state $\boldsymbol{x} = (x, y, z, \theta)$ is considered colliding if the height z is above the maximum offset collision surface D^+ or below the minimum offset collision surface D^- at $\boldsymbol{x}_{xy}$, where $\boldsymbol{x}_{xy}$ is the projection of the state $\boldsymbol{x}$ onto the xy plane.

$$\mathcal{X}^- = \left\{ \mathbf{x} \mid z > D^+(\mathbf{x}_{xy}) \text{ or } z < D^-(\mathbf{x}_{xy}) \right\} \tag{5}$$

IV. Safe Periodic Sets

A. Inevitable Collision States

ICSs were first proposed in [15], defining a set of states where the vehicle will inevitably result in a collision regardless of all feasible inputs. We adapt the original definition, which uses trajectories, to operate on kinematically feasible paths.

Definition 4.1 (Inevitable Collision State [15]): Define $\Gamma(\mathbf{x}, S)$ as the set of all kinematically feasible paths starting from initial state $\mathbf{x} \in \mathcal{X}^+$ with length S. $\boldsymbol{\eta} \in \Gamma(\mathbf{x}, S)$ is a single path starting from $\mathbf{x}$. An inevitable collision state is defined as

$$\mathcal{X}_{ICS} = \left\{ \mathbf{x} \mid \forall \boldsymbol{\eta} \in \Gamma(\mathbf{x}, \infty) \ \ \exists s \in [0, \infty) \ \ \boldsymbol{\eta}(s) \in \mathcal{X}^- \right\}.$$

In simple terms, if all possible paths from $\mathbf{x}$ will eventually result in passing through a forbidden state, then $\mathbf{x} \in \mathcal{X}_{ICS}$. Conversely, if there exists a kinematically feasible, collision-free path from $\mathbf{x}$, then $\mathbf{x}$ is not an ICS.

B. Safe Periodic Path

An infinite horizon path must be evaluated to ensure a particular state is not in $\mathcal{X}_{ICS}$ [17]. Since evaluating an infinite horizon path is typically impractical, we use periodic paths to find a subset of all possible safe states $\mathcal{X}_{safe}$. A path is periodic if any state on the path is repetitively visited with a certain period λ as in the following condition:

$$\forall s \in [0, \infty) \quad \boldsymbol{\eta}(s) = \boldsymbol{\eta}(s + \lambda)$$

The key idea is that if we can find a collision-free path to a periodic state, the path is safe. If a state $\boldsymbol{x}$ is on a periodic path $P(\cdot)$, it is sufficient to check the collision of one period of the path to check whether the state is safe [17].

Corollary 4.1 (Safe Periodic Path): If a periodic path $P(\cdot)$ is not in a collision within a single period, all its states are not in an ICS.

$$\forall s \in [0, \lambda) \ \ P(s) \notin \mathcal{X}^- \iff \forall s \in [0, \infty) \ \ P(s) \notin \mathcal{X}_{ICS}$$

While the method is applicable for all periodic paths, we choose a circular path s, as the isotropic distance to the center of the loiter simplifies collision checking. Consider a level circular path parameterized by the center position $\boldsymbol{c} = (c_x, c_y, c_z) \in \mathbb{R}^3$ and radius R. The heading θ is omitted for brevity.

$$P(\boldsymbol{c}, R) = \{ \boldsymbol{x} \mid z = c_z, \|\boldsymbol{x}_{xy} - \boldsymbol{c}_{xy}\| = R \}$$

To determine whether the circular path is collision-free, we check whether the states on the path lie between the two offset collision surfaces, D^+ and D^-. Therefore, a state $\boldsymbol{x}$ on a circular path $P(\boldsymbol{c}, R)$ is safe if it satisfies the condition (6).

$$\forall \boldsymbol{x} \in P(\boldsymbol{c}, R) \ \ D_d^+(\boldsymbol{x}) > x_z > D_d^-(\boldsymbol{x}). \tag{6}$$

C. Efficient Safety Evaluation

In (6), we must iterate through all position states along the circular path for collision-checking. We simplify the computation by replacing the circle with a disk and comparing two horizontal offset surfaces instead. The underlying sufficient condition of a collision-free circular path is to check whether the horizontal distance from the circle's center to the offset collision surface D^+, D^- is larger than the path radius. Thus, we offset D^+ and D^- horizontally by radius R, resulting in the horizontal offset surfaces $\mathcal{H}_R^+(\boldsymbol{m})$ and $\mathcal{H}_R^-(\boldsymbol{m})$.

$$\begin{aligned} \boldsymbol{m} \in \mathcal{M} \ \mathcal{H}_R^-(\boldsymbol{m}) &= \max_{\forall \boldsymbol{m}' \in \{\boldsymbol{m}' \mid \|\boldsymbol{m}' - \boldsymbol{m}\| \leq R\}} D^-(\boldsymbol{m}') \\ \mathcal{H}_R^+(\boldsymbol{m}) &= \min_{\forall \boldsymbol{m}' \in \{\boldsymbol{m}' \mid \|\boldsymbol{m}' - \boldsymbol{m}\| \leq R\}} D^+(\boldsymbol{m}') \end{aligned} \tag{7}$$

If the altitude of the center position $\mathbf{c} = (c_x, c_y, c_z)$ satisfies $c_z > \mathcal{H}^-(\boldsymbol{c}_{xy})$ and $c_z < \mathcal{H}^+(\boldsymbol{c}_{xy})$, a circular path $P(\mathbf{c}, R)$ is safe. We define the set of *valid loiter positions* over the whole map as follows.

Definition 4.2 (Valid Loiter Position): A collision-free circular path exists at $\boldsymbol{m}$ if $\mathcal{H}^+(\boldsymbol{m}) > \mathcal{H}^-(\boldsymbol{m})$.

$$\mathcal{M}_{valid} = \left\{ \mathbf{m} \in \mathcal{M} | \mathcal{H}^+(\boldsymbol{m}) > \mathcal{H}^-(\boldsymbol{m}) \right\} \tag{8}$$

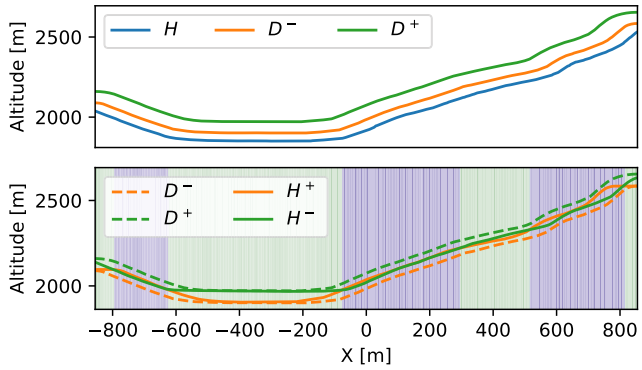


Fig. 3. Visualization of the terrain and maximum(120 m), minimum distance(50 m) offset collision surfaces of the example of Fig. 1 (top). Valid loiter positions according to Eq. (8) are marked as green and invalid regions are marked as blue (bottom).

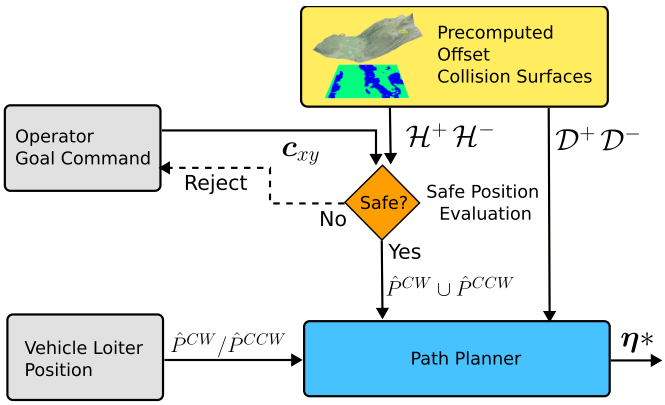


Fig. 4. Overview of the proposed method. The operator commands a 2D goal position c_{xy}, in which the safety is evaluated using the offset collision surface $\mathcal{H}^+, \mathcal{H}^-$. If the commanded goal position is safe, a planning problem is solved for path $\eta*$ using the start and goal circular path and the collision surface $\mathcal{D}^+, \mathcal{D}^-$.

Following the definition, the permitted loiter positions, used to set the start and goal, are deterministic and can be precomputed for a fixed loiter radius and terrain map. Fig. 3 shows that valid loiter positions exist only in flatter terrain.

V. Path Planning

In Section IV, we showed how the safety of a periodic path can be evaluated using simple geometric operations. This section explains how we find safe routes using sampling-based planners and safe periodic sets. The information flow of the planning approach is shown in Fig. 4

A. Start and Goal States

A sufficient safety condition is that if a path $\boldsymbol{\eta}(s)$ is collision-free and the final state is not in $\mathcal{X}_{ICS}$, the whole path is not in $\mathcal{X}_{ICS}$ [16]. Therefore, if we can evaluate the safety of $\mathcal{X}_{goal}$ using the strategy proposed in Section IV, we can solve the problem of finding a collision-free path $\boldsymbol{\eta}(s)$ from $\mathcal{X}_{start}$ to $\mathcal{X}_{goal}$. We assume that $\mathcal{X}_{start}$ is safe.

From (7), a horizontal goal position is valid if its circle center $\boldsymbol{c}_{xy} \in \mathcal{M}_{valid}$. A valid altitude of the circle center $\boldsymbol{c}$ is any value between the two offset surfaces $\mathcal{H}_R^+(\cdot)$, $\mathcal{H}_R^-(\cdot)$. We simplify the altitude selection by choosing the average between the two surfaces, as shown in (9).

$$c_z = \frac{\mathcal{H}^+(\boldsymbol{c}_{xy}) + \mathcal{H}^-(\boldsymbol{c}_{xy})}{2} \tag{9}$$

This results in choosing the circular path with the maximum clearance from the offset collision surfaces. Also, the automatic altitude selection offloads the operator's workload. The valid loiter position check and automatic altitude selection reduce the operation complexity from selecting a four-dimensional goal position and heading $\{x, y, z, \theta\}$ to a two-dimensional circle center selection, guaranteeing safety.

We generate the start and goal sets $\mathcal{X}_{start}$, $\mathcal{X}_{goal}$ by discretizing the circular path $P(\mathbf{c}, R)$ into N discrete sets of states $\hat{P}(\mathbf{c}, R)$, for $i \in \{1, \ldots, N\}$, as described in (10).

$$\hat{P}^{CW}(\mathbf{c}, R) = \left\{ \boldsymbol{x}_i | \theta_i = \left(\frac{i}{N} + \frac{1}{2} \right) \pi \right\} \tag{10}$$

$$\hat{P}^{CCW}(\mathbf{c}, R) = \left\{ \boldsymbol{x}_i | \theta_i = \left(\frac{i}{N} - \frac{1}{2} \right) \pi \right\}$$

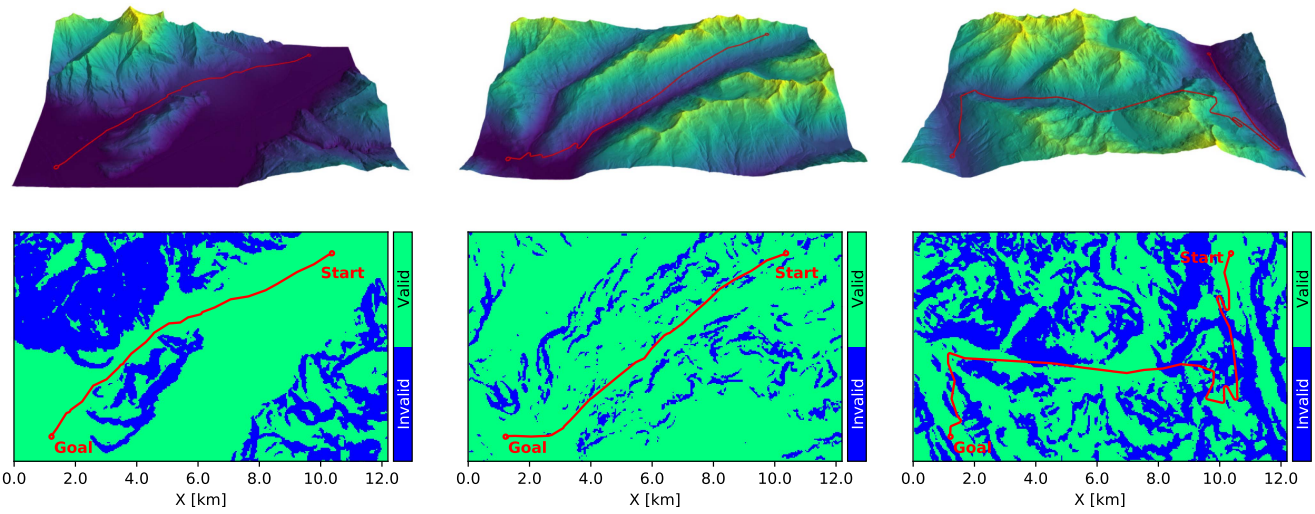
$$s.t. \quad \boldsymbol{x}_i = \begin{pmatrix} Rcos(\theta_i) + c_x \\ Rsin(\theta_i) + c_y \\ c_z \\ \theta_i \end{pmatrix} \tag{11}$$

Where $\hat{P}^{CW}(\mathbf{c}, R)$, $\hat{P}^{CCW}(\mathbf{c}, R)$ are clockwise and counter-clockwise circular paths. While the start loiter direction would depend on the vehicle state, both loiter directions are valid for the goal. We define the goal set as the union of both directions.

B. Planning

We plan using RRT*, a probabilistically complete, asymptotically optimal sampling-based planner [23]. The planner samples in the Dubins airplane space, which is defined by a quasi-distance metric $d_D(x_i, x_j)$ such that d_D is the length of the Dubins airplane path between two arbitrary states $x_i, x_j \in \mathcal{X}$ [14]. In this work, we use the minimum turn radius to be identical with the terminal circular path radius explained in Section V-A, since using the minimum turn radius maximizes valid loiter positions. However, the radius of the terminal circular path can be set independently of the minimum turn radius of the Dubins airplane space. Note that the distance metric is a quasi-metric, given that the distance may not be symmetric between the two states x_i, x_j.

To calculate the shortest path, we use [24], which divides the Dubins distance in the horizontal plane and altitude differences into three cases to determine whether an ascending/descending helix is required to reach the goal while satisfying the flight path angle constraints. As in [25], we take the sub-optimal Dubins distance to remove the need for a line search in the *medium goal altitude* case. To make the Dubins curve calculation more efficient, we use the set classification method proposed in [26] to prevent exhaustive computation of different Dubins path types.



(a) *Sargans* (b) *Dischma Valley* (c) *Gotthard Pass*

Fig. 5. (above) 3D visualization of the elevation data and an example of a planned path and (below) two-dimensional projection of the valid loiter position set $\mathcal{M}_{valid}$ (green) in three environments: (a) *Sargans*, (b) *Dischma Valley*, and (c) *Gotthard Pass*.

VI. Planner Evaluations

We run two quantitative evaluations for the proposed approach. First, we illustrate how terrain ruggedness reduces the safe terminal state space and how our approach relaxes this more complex planning problem on an operational level. Second, we show that our planner is capable of onboard real-time path planning, even in mountains with narrow passages.

A. Setup

The approach is evaluated in three different terrain environments: *Sargans*, *Dischma Valley*, and *Gotthard Pass*. Fig. 5 shows the selected DEMs from SwissAlti3D [27] with the extent of 12.2 km $\times$ 7.48 km with 10 m lateral resolution. The DEM uses CH1903/LV03 coordinates, with Bessel 1841 as the vertical datum. All terrains are identical in extent but have increasingly rugged topography. The *Gotthard Pass* environment is the most difficult to traverse, containing only a narrow pass. The minimum and maximum distance to the terrain was 50 m and 120 m, corresponding to European regulations [7]. The vehicle's minimum turn radius is $R = 66.67$ m. The maximum flight path angle is $\gamma = 8.6°$. The planners were implemented in OMPL [28] and a Robot Operating System (ROS). Map representations use the `grid_map` toolbox for efficiently representing 2.5D surfaces [21]. We executed all benchmarks on a 2.9 GHz Intel Core i7-10700 CPU with 32 GB memory.

B. Safe Periodic Goal Selection

A major difficulty when planning fixed-wing missions in mountainous regions is the selection of the goal state heading. A poorly selected heading may lead to crashing into the mountain upon arrival or an overly complicated approach path as illustrated in Fig. 1. Our safe periodic goal selection relaxes this problem by making the terminal state yaw agnostic with periodic paths.

To demonstrate, we compare the reachable subset of the map when selecting a terminal circle with a *Valid Loiter Position*. Fig. 5 shows the valid *Valid Loiter Position* locations marked green for the three environments. Loitering is generally possible in shallow regions, while steep regions prevent loitering due to the narrow space available. This effect becomes more severe as the environment becomes more rugged, effectively making the planning problem harder. The portion of *Valid Loiter Position* location of each environment is 77% for *Sargans*, 79% for *Dischma Valley*, 73% for the *Gotthard Pass*, with the *Gotthard Pass* environment being very jagged. Furthermore, the red paths show that the *Valid Loiter Position* selection ensures a smooth goal approach. Note that for the *Gotthard Pass* environment, the path needs to pass through invalid loiter positions, meaning that the vehicle would not be able to initiate a loiter safely.

In summary, the proposed safe periodic goal selection makes the goal selection process simpler and lowers the workload by eliminating the need to specify the heading as a terminal state.

C. Onboard Planner Convergence

The proposed safe periodic goal selection provides an opportunity for simpler on-the-fly replanning. We show that our approach can quickly find a valid path onboard the vehicle. For this experiment, we fix the start and goal circle at $\boldsymbol{c}_{start} = (2992, -4720, \cdot)[m]$, $\boldsymbol{c}_{goal} = (-2992, 4880, \cdot)[m]$ relative to the center of the map for all maps, with both radii of $R = 66.67$ m. We repeat the planning 50 $\times$ with a compute time budget of 500 s for each planning loop. The altitude of the start and goal positions are omitted since it is defined by the terrain using (9).

Fig. 6 shows that the planner always finds a solution. The median time to find the initial solution is 1.00 s for *Sargans*, 0.90 s for *Dischma Valley*, and 9.72 s for the more difficult *Gotthard Pass*. A vehicle flying at 15 ms^{-1} would take 28 s to fly a full loiter with 66.7 m radius. Therefore, the vehicle can

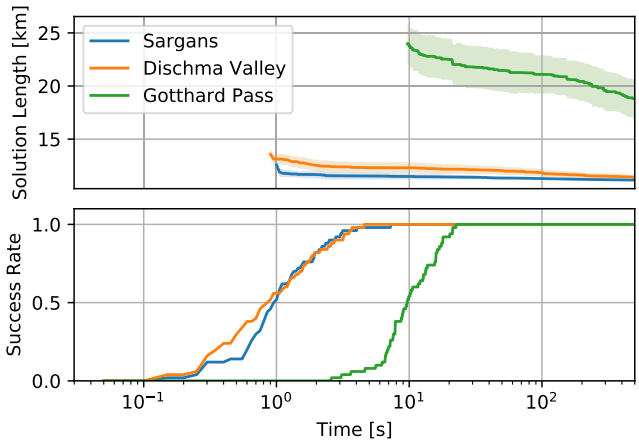


Fig. 6. Planner success rates and solution path length using the proposed planning approach.

find a safe path within a fraction of the time required to execute one loiter maneuver.

Despite the start and goal having the same Euclidean distance in all three test cases, the valid path length is significantly longer for *Gotthard Pass*. Also, the solution length does not converge within 500 s. Both of these results are due to the steep mountain faces that make it difficult to connect Dubins airplane path segments, especially in the presence of climb rate, curvature, and altitude constraints. However, the proposed planner always finds a valid initial solution within 30 s in all test cases. Here, the relaxed goal state positively influences the success rate, simplifying goal state connection. However, since the planning approach is identical except for the larger goal set, the difference in convergence was minimal.

VII. Real World Flight Tests

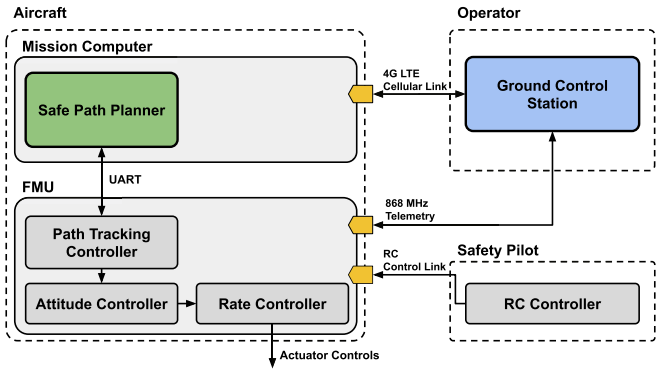
We conducted an actual flight experiment on and sUAS. The primary goal of this experiment was to demonstrate that our planner enables safe and legal flights in steep environments. Secondary, we validate our onboard replanning capabilities.

A. Setup

The test was conducted in the valley of Riemenstalden, Switzerland. A video of the demonstration can be found in the supplementary materials. Fig. 8(a) shows a terrain excerpt provided by SwissAlti3D [27]. The terrain is characterized by a relatively steep and narrow valley with a larger flat area at the bottom and a few plateau-like areas on the slopes. In the experiment, the operator dynamically provided four arbitrary goal positions from the safe set shown to the operator through a graphical user interface. The sUAS autonomously replanned during loitering and executed the new missions.

The experiment platform was a tiltrotor VTOL aircraft with a mass of 5.7 kg, and a wingspan of 2300 mm based on the Makefly Easy Freeman, shown in Fig. 7(a). The platform hovers during takeoff and landing with the tractor motors in front of the wing tilted upwards. During the remaining flight, the vehicle flies as a fixed-wing vehicle with the front rotors tilted forward. The

(a) Tiltrotor VTOL platform in flight during take off in Riemenstalden, Switzerland.



(b) The system consists of a Mission computer, flight management unit (FMU) which are controlled by an operator and a safety pilot.

Fig. 7. Overview of the system that was used for flight testing.

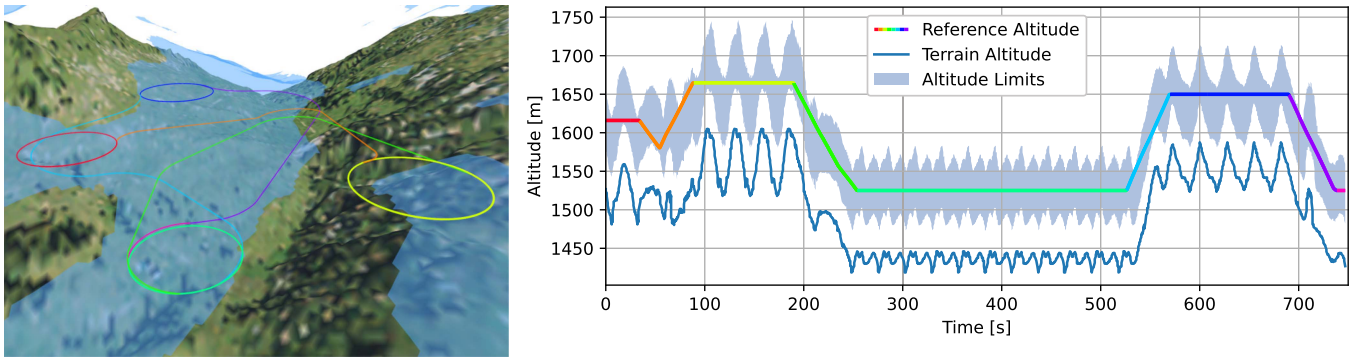
minimum turning radius of the vehicle is assumed to be 80 m. The maximum flight path angle is 8.6°. The planner's lower altitude bound was 50 m to remain safely above the canopy of the trees. According to the legal requirements, the upper altitude bound was 120 m [7].

The platform carries an Intel NUC, equipped with a 3.5 GHz Intel Core i7-7567 U CPU, which runs the proposed planning method on demand. Fig. 7(b) shows the communication flow. The operator communicates to the mission computer through a cellular connection. Upon user approval, the mission computer sends setpoints to a FMU at 10 Hz. The FMU runs stock PX4 autopilot with regular GNSS navigation. The ground station communicates with the autopilot through a cellular connection. For example, telemetry data from the plane is visualized on the DEM for planning a new mission. An additional 868 MHz telemetry connection is used for redundancy, and an RC link connects a safety pilot to the vehicle to abort the mission in case of an emergency.

B. Smoothed Dubins Path Tracking

The mission computer continuously sends path-tracking reference commands $\mathbf{r} = [\mathbf{p}, \mathbf{v}, \kappa]$ from the closest point on the Dubins airplane path $\mathbf{p}$, tangent $\mathbf{t}$, and curvature κ to the FMU. The reference is passed to a nonlinear path following guidance controller based on [13].

Given that Dubins curves consist of arcs and line segments, the curvature is a discrete set, which results in discontinuous



(a) 3D visualization of reference path over terrain. (b) Time record of vehicle altitude, terrain altitude, and offset collision surfaces D^+ and D^-.

Fig. 8. Visualization of the vehicle path during flight tests in the valley of Riemenstalden, Switzerland. Mission legs are color-coded. *Valid loiter position* surfaces are shaded blue. The altitude limits are defined by the offset collision surface D^+ and D^-. Due to small vertical tracking errors the vehicle altitude overlaps with the reference, thus was omitted from this figure and instead refer to Fig. 9.

jumps in reference curvature. The jumps degrade the tracking performance of the path-tracking controller. Therefore, a linear curvature blending strategy smoothes the curvature reference for better tracking performance. Given the current segment i, with curvature κ_i, and the next segment $i+1$, with curvature κ_{i+1}, we blend the two segments with the portion $\psi \in [0, 1]$, such that

$$\kappa = \psi\kappa_i + (1-\psi)\kappa_{i+1}, \tag{12}$$

where $\psi = \max(1.0, l/\bar{l})$, l is the remaining distance to the end of the segment, and $\bar{l}$ is the threshold where the segment is blended when nearing the end of the segment. For the experiments, $\bar{l}$ was 10 m, which was tuned with respect to the vehicle's roll time constant.

C. Safe and Legal Flight Test

Fig. 8(a) shows a total of four individual mission legs. The operator commanded all circular goal centers to lie within the safe goal surface. The safe regions are generally located in the flatter parts of the valley, and the vehicle can traverse through steep slopes only for climbing and descending. Exemplary is the transition from the yellow to the green circle. The platform cannot go straight to the goal because of its limited descending rate and tight altitude bounds. Instead, it has to perform a large detour along the slope to reach the valley.

Fig. 8(b) gives more insight into the planned path. The plot reveals that the planned reference path always remains between the minimum and maximum offset collision surface constraints. The path passes as close as 0.56 m to the maximum distance constraint and 0.915 m to the minimum distance constraint. Notice the significant periodic variation in altitude limits during the yellow loiter circle between 100 s to 200 s. This is due to the level, circular flight over the steep slope, stressing the necessity of terrain-aware mission planning to remain within altitude limits in non-flat environments.

Finally, we investigate the path tracking error to determine how well the flight controller handles the Dubins path approximation. Fig. 9 shows good altitude tracking, where the tracking

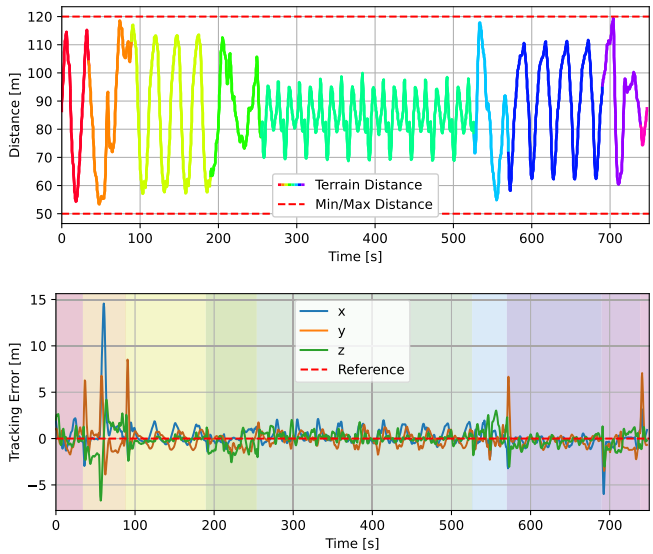


Fig. 9. Visualization of the distance of the vehicle to terrain with segment colors (top) and the path tracking error with segment colors overlaid (bottom) during the flight test.

errors were (0.65 ± 0.73) m with a maximum of 6.69 m. On the other hand, the lateral tracking errors were (1.06 ± 1.37) m, with a maximum of 14.6 m. Both the maximum altitude and lateral tracking errors occur in the middle of the orange segment (∼ 60 s), where a full right-handed turn and climb occur after a long descent. We attribute these errors to the curvature discontinuity in the Dubins path discussed in Section VII-B. Despite the tracking errors, the platform remains within the terrain distance constraints. The larger deviation occurred when the clearance to both distance constraints was large.

Our experiment shows that for our system and this environment, the planner parametrization was well chosen. In the case of a less performant tracking controller or steeper terrain, the altitude limits could be tightened, the climb rate reduced, or the minimum turning radius enlarged to increase safety at the expense of a smaller workspace.

VIII. Conclusion

This letter investigates autonomous fixed-wing flight in altitude-constrained mountainous regions. Based on the concept of an inevitable collision state (ICS), we propose a safe planning approach that uses circular periodic paths to verify the safety of a goal state. We show that circular periodic paths enable us to simplify the goal selection process, compared to conventional start and goal states specified by a singular position and heading. We incorporate this into a sampling-based planning framework and demonstrate the planner in real-world experiments. The experiments highlight the necessity for terrain-aware mission planning in the mountains, where climb rate, curvature, and altitude constraints drastically limit the search space.

To the authors' knowledge, this is the first demonstration of a fixed-wing sUASs autonomously navigating in alpine terrain, complying with the European regulations [7]. The main limitation of this work is that the path planning is done in the Dubins Airplane space, which is only able to represent a small subset of possible paths. Also, environmental effects such as wind are ignored from the planning problem. Future work may include safe, any-time replanning due to prioritized air traffic, handling wind and tracking uncertainty, or deviation from the nominal path.

Acknowledgment

The authors would like to thank Yves Bühler and Elisabeth Hafner at the WSL-Institute for Snow and Avalanche Research (SLF) for their expertise and support in avalanche mapping and modeling. The authors would like to thank Silvan Fuhrer and Thomas Stastny at Auterion AG for supporting the hardware used for the real flight tests. The authors would also like to thank David Rohr at ETH Zurich, for being a reliable safety pilot for countless flight tests.

References

[1] P. Oettershagen et al., "Robotic technologies for solar-powered UAVs: Fully autonomous updraft-aware aerial sensing for multiday search-and-rescue missions," *J. Field Robot.*, vol. 35, no. 4, pp. 612–640, 2018.

[2] A. Bircher et al., "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Auton. Robots*, vol. 40, pp. 1059–1078, 2016.

[3] G. Jouvet, Y. Weidmann, E. V. Dongen, M. P. Lüthi, A. Vieli, and J. C. Ryan, "High-endurance UAV for monitoring calving glaciers: Application to the Inglefield Bredning and Eqip Sermia, Greenland," *Front. Earth Sci.*, vol. 7, 2019, Art. no. 209.

[4] P.-H. Lin and C.-S. Lee, "The eyewall-penetration reconnaissance observation of typhoon longwang (2005) with unmanned aerial vehicle, aerosonde," *J. Atmospheric Ocean. Technol.*, vol. 25, no. 1, pp. 15–25, 2008.

[5] K. Shah, G. Ballard, A. Schmidt, and M. Schwager, "Multidrone aerial surveys of penguin colonies in antarctica," *Sci. Robot.*, vol. 5, no. 47, 2020, Art. no. eabc3000.

[6] Y. Bühler, M. S. Adams, A. Stoffel, and R. Boesch, "Photogrammetric reconstruction of homogenous snow surfaces in alpine terrain applying near-infrared UAs imagery," *Int. J. Remote Sens.*, vol. 38, no. 8/10, pp. 3135–3158, 2017.

[7] E. Union, "Commission implementing regulation (EU) 2019/947 of 24 may 2019 on the rules and procedures for the operation of unmanned aircraft," *Official J. Eur. Union*, vol. 62, pp. 45–71, 2019.

[8] A. Bry, C. Richter, A. Bachrach, and N. Roy, "Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 969–1002, 2015.

[9] P. Oettershagen, F. Achermann, B. Müller, D. Schneider, and R. Siegwart, "Towards fully environment-aware UAVs: Real-time path planning with online 3D wind field prediction in complex terrain," 2017, *arXiv:1712.03608*.

[10] D. Lee, H. Song, and D. H. Shim, "Optimal path planning based on spline-RRT* for fixed-wing UAVs operating in three-dimensional environments," in *Proc. 14th Int. Conf. Control Autom. Syst.*, 2014, pp. 835–839.

[11] J. Levin, A. Paranjape, and M. Nahon, "Motion planning for a small aerobatic fixed-wing unmanned aerial vehicle," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 8464–8470.

[12] M. Seemann and K. Janschek, "RRT-based trajectory planning for fixed wing UAVs using bezier curves," in *Proc. 41st Int. Symp. Robot.*, 2014, pp. 1–8.

[13] T. Stastny and R. Siegwart, "On flying backwards: Preventing run-away of small, low-speed, fixed-wing UAVs in strong winds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 5198–5205.

[14] H. Chitsaz and S. M. LaValle, "Time-optimal paths for a dubins airplane," in *Proc. 46th IEEE Conf. Decis. Control*, 2007, pp. 2379–2384.

[15] T. Fraichard and H. Asama, "Inevitable collision states – A step towards safer robots?" *Adv. Robot.*, vol. 18, no. 10, pp. 1001–1024, 2004. [Online]. Available: https://inria.hal.science/inria-00182063

[16] S. Petti and T. Fraichard, "Partial motion planning framework for reactive planning within dynamic environments," in *Proc. 2nd Int. Conf. Informat. Control, Autom. Robot.*, 2005, pp. 199–204, doi: 10.5220/0001185401990204.

[17] K. E. Bekris, "Avoiding inevitable collision states: Safety and computational efficiency in replanning with sampling-based algorithms," in *Proc. Workshop Guaranteeing Safe Navigation Dyn. Environ. Int. Conf. Robot. Autom.*, 2010. [Online]. Available: https://safety2010.inrialpes.fr/proceedings/index.html

[18] S. Arora, S. Choudhury, D. Althoff, and S. Scherer, "Emergency maneuver library-ensuring safe navigation in partially known environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 6431–6438.

[19] D. Althoff, M. Althoff, and S. Scherer, "Online safety verification of trajectories for unmanned flight with offline computed robust invariant sets," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 3470–3477.

[20] R. Triebel, P. Pfaff, and W. Burgard, "Multi-level surface maps for outdoor terrain mapping and loop closing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 2276–2282.

[21] P. Fankhauser and M. Hutter, "A universal grid map library: Implementation and use case for rough terrain navigation," *Stud. Comput. Intell.*, vol. 625, pp. 99–120, 2016.

[22] L. Polidori and M. El Hage, "Digital elevation model quality assessment methods: A critical review," *Remote Sens.*, vol. 12, no. 21, 2020, Art. no. 3522, doi: 10.3390/rs12213522.

[23] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," May 2010, *arXiv:1005.0416*.

[24] T. McLain, R. W. Beard, and M. Owen, "Implementing dubins airplane paths on fixed-wing UAVs," 2014. [Online]. Available: https://scholarsarchive.byu.edu/facpub/1900

[25] D. Schneider, "Path planning for fixed-wing unmanned aerial vehicles," Master's thesis, ETH Zürich, Zürich, Switzerland, 2016.

[26] A. M. Shkel and V. Lumelsky, "Classification of the dubins set," *Robot. Auton. Syst.*, vol. 34, pp. 179–202, 2001.

[27] "Swiss Federal Office of Topography swisstopo," 2023. [Online]. Available: https://www.swisstopo.admin.ch/en/height-model-swissalti3d. [Online]. Available: https://www.swisstopo.admin.ch/en/height-model-swissalti3d

[28] I. A. Şucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robot. Automat. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012. [Online]. Available: https://ompl.kavrakilab.org