

Article

Not peer-reviewed version

Gradient Guided Search for Autonomous Contingency Landing Planning

[H. Emre Tekaslan](#)^{†,‡} and [Ella M. Atkins](#)^{*,†,‡}

Posted Date: 11 August 2025

doi: 10.20944/preprints202508.0709.v1

Keywords: assured contingency landing management; advanced air mobility; discrete search; real-time risk mitigation; path planning



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Gradient Guided Search for Autonomous Contingency Landing Planning

H. Emre Tekaslan ^{*,†,‡}  and Ella M. Atkins ^{†,‡} 

Kevin T. Crofton Aerospace and Ocean Engineering Department, Virginia Tech, Blacksburg, VA 24061, USA; ematkins@vt.edu

* Correspondence: tekaslan@vt.edu

† Current address: 1600 Innovation Dr., Blacksburg, VA 24061, USA.

‡ These authors contributed equally to this work.

Abstract: The growing reliance on autonomy in crewed and uncrewed air vehicles necessitates a real-time solution for assured contingency landing management during in-flight emergencies. This paper presents a novel gradient-guided search algorithm for risk-aware emergency landing trajectory generation with a fixed-wing aircraft loss of thrust use case. This framework integrates a compact four-dimensional discrete search space with aircraft kinematic and ground risk cost. A multi-objective cost function is employed, combining flight envelope feasibility, optimal descent, and overflown population risk terms. To ensure discrete search convergence, a constrained hypervolume definition is introduced around the destination. A holding pattern identification algorithm is defined to minimize risk during the necessary flight path angle constrained descent to final approach. Planner effectiveness is validated through randomly generated case studies over a region of Long Island, NY under steady wind conditions. Benchmark comparisons with a 3D Dubins solver demonstrate the approach's improved risk mitigation and acceptable real-time computation overhead. Future development will focus on integrating air traffic avoidance into the discrete search-based landing planner.

Keywords: assured contingency landing management; advanced air mobility; discrete search; real-time risk mitigation; path planning

1. Introduction

Advanced Air Mobility (AAM) will support a combination of Uncrewed Aircraft Systems (UAS) and larger air taxi platforms flying low over densely populated urban areas. Although electric Vertical Takeoff and Landing (eVTOL) aircraft are expected to offer a Lift-Cruise configuration [1,2], a variety of conditions ranging from propulsion module failure to unexpectedly low battery energy or fuel can render any aircraft unable to remain aloft long-term. A Lift-Cruise eVTOL with low battery energy can safely fly longer, with more margin, if it remains in cruise configuration rather than converting to vertical flight. Whether operating in wing lift (cruise) mode due to a fixed-wing configuration or due to a low energy or eVTOL propulsion unit failure condition, a contingency landing plan can be safest for aircraft, any onboard occupants (if present), and people on the ground below when this aircraft relies on wing lift throughout a stable planned descent to land on a nearby runway. Because aircraft can glide without consuming fuel in wing lift mode, it is the most energy efficient configuration. Gliding is also feasible regardless of which propulsion unit fails. Therefore, this paper models the aircraft in wing lift mode during contingency descent to landing. While this work is motivated in part by larger AAM platforms such as eVTOL air taxis, the underlying approach is equally applicable to smaller-sized UAS capable of operating in wing-lift mode. The algorithm's reliance on aerodynamic gliding and discrete ground risk data enables autonomous onboard planning for drones conducting logistics, surveillance, or emergency response missions - applications that are becoming increasingly common in densely populated areas [3–5].

During urgent or emergency landings, the distressed aircraft has priority over other traffic but must deal with whatever winds and weather are present. This work therefore focuses on consideration

of steady wind and flight envelope margin and appropriately assumes other aircraft will avoid collision with the distressed aircraft as it lands. Path planning for urgent or emergency landing requires a four-dimensional solution that meets strict spatiotemporal constraints and minimizes safety risks. Along with implementation of degraded aircraft performance constraints for flight safety, ground risk is quantified by employing discrete population census data in this paper.

Fixed-wing (wing lift/cruise configuration) aircraft have non-holonomic motion constraints that require maintaining forward motion at a minimum (stall) speed and limiting turn radius. Urgent landing descent paths must avoid obstacles, minimize risk, and align with suitable runways while accounting for steady winds in planning and adjusting to actual variable winds during plan execution. There are several methods for aircraft path planning. However, not all of them are suitable for constrained minimum-risk path planning in complex environments. For instance, although they are fast to compute, geometric solutions such as Dubins paths [6–8] solely incorporate path length and curvature metrics. Despite the success of optimal control (OC) and model predictive control (MPC), these methods suffer from initial guess dependency, non-convergence, and low computational efficiency in complex environments [9–13]. In addition, it is difficult to adapt inherently discrete ground risk metrics to OC and MPC as solutions may become stuck at local minima that may not meet feasibility constraints. Similarly, Hamilton-Jacobi-Bellman (HJB) reachability does not scale efficiently due to the curse of dimensionality [14]. Moreover, the HJB value function requires spatial continuous differentiability [15], a requirement that discrete ground risk cost violates. Even though such metrics can be approximated by projecting data onto smooth hyperplanes (e.g., Fourier series expansion) this task increases complexity while reducing the data resolution. Note that dynamic cost metric updates such as ground risk values derived from mobile phone activity [16] and real-time ground traffic data [17] render data projection computationally unaffordable for real-time risk quantification as part of contingency planning.

Tree-search-based path planners are well-suited for diverse, discrete cost metrics such as ground risk. As example related works, Ref. [18] applies a modified A* algorithm to find a shortest feasible path. In [19] and [20], Monte Carlo Tree Search was exploited to plan UAS flight paths. A rapidly-exploring-random tree (RRT) was used in [21–23] to safely navigate a space with obstacles. Ref. [24] used pre-calculated maneuvers for emergency landing planning for a gliding aircraft. Markov Decision Process (MDP) formulations in Refs. [25,26] focused on uncertainty in aircraft performance and prognostics-based margin, respectively, in emergency landing planning. Tree-search over feasible post-failure maneuver primitives for emergency landing planning was proposed in [27] as an extension of a Dubins-based geometric solver in [28]. A three-dimensional A* variant in [29] balanced flight path and landing site risk but faced significant complexity challenges when path length was substantial. This work advances previous search-based and MDP approaches by integrating risk, kinematic, and flight envelope constraints within a solver that efficiently generates feasible long-distance 4D landing paths. This paper's approach accounts for operational altitude ranges with a minimum-risk loitering approach and wind effects while maintaining a compact heading-constrained search space.

This article is a revised and expanded version of a paper entitled Gradient Guided Search for Aircraft Contingency Landing Planning [30] which was presented at the IEEE International Conference on Robotics and Automation, Atlanta, USA, 19-23 May 2025. The discrete search contingency landing planner in this paper adopts a priority queue and operators inspired by the Theta* search algorithm [31] to accommodate flight envelope constraints as a function of steady wind. By fusing feasible path and population risk cost metrics, the presented method simultaneously considers safety of the landing (e.g., path margin) and risk due to a distressed aircraft passing over an urban population. The multi-objective cost function forms a gradient field that significantly reduces state-space exploration thus to minimize computational overhead to support real-time implementation. A constrained four-dimensional (4D) position and heading volume around approach fix is introduced to guarantee search convergence. Since ground risk is negligible at high altitude, the planner incorporates a three dimensional (3D) Dubins path solver with a holding point identification scheme to efficiently generate a path to a

minimum-risk holding pattern that dissipates altitude. The algorithm's effectiveness is demonstrated through use cases under steady wind at different initial states. Search-based planning results are benchmarked against 3D Dubins path solvers, evaluating overflown population risk, deviations from optimal gliding trajectories, and computational overhead. The key contributions of this work include:

1. Efficient and feasible contingency landing planning with a compact 4D discrete search framework guided by a cost function with a constraint margin gradient field.
2. A constrained hypervolume definition around an approach fix to ensure discrete search convergence.
3. A real-time minimum-risk aircraft holding pattern placement algorithm and its integration into contingency landing planning.
4. Assured contingency landing plan generation within a prescribed time limit.

This paper is organized as follows. Section 2 introduces fundamental concepts that this work builds upon including gliding flight envelope determination of an aircraft in fixed-wing mode, reachability, and geometric path planning. Section 4 presents the key components of the contingency landing planner such as the building blocks of the gradient-guided search incorporating flight feasibility and landing risk assessment with a convergence guarantee. Below, preliminaries and methodology are followed by use case studies over a Long Island, NY region with a Cessna 182 fixed-wing performance model serving as a surrogate to a four-passenger AAM aircraft operating in fixed-wing mode.

2. Preliminaries

This section summarizes the principles of fixed-wing aircraft flight, reachability, and geometric path planning. These concepts are essential for understanding and developing emergency landing planning procedures.

2.1. Fixed-Wing Aircraft Performance

A fixed-wing aircraft is exposed to three fundamental forces [32]: aerodynamic, gravitational, and propulsive, as shown in Figure 1.

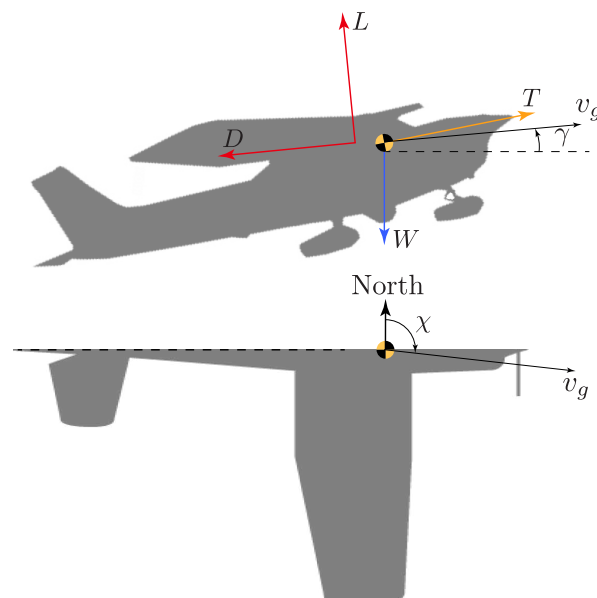


Figure 1. Fundamental forces acting in the aircraft longitudinal plane.

High wing lift L to drag D ratio is the key to efficient fixed wing or "cruise" mode eVTOL flight [33]. Lift is mainly produced by the pressure difference between upper and lower wing surfaces and requires forward flight above a minimum stall speed. In cruise flight, forward thrust T generated by propulsion units must counter drag D while lift L balances total weight W . The four primary forces (L, D, T, W)

define flight path angle γ , the angle between the ground velocity v_g and the horizontal plane, with $\gamma = 0$ in level flight. Aircraft course angle, denoted by χ , is the angle between North and the horizontal projection of v_g .

Best glide flight path angle in forward flight $\gamma_{bg} \in \mathbb{R}^+$ (1) defines the most shallow descent possible without thrust. It is defined by the fixed-wing aircraft nonlinear equations of motion $f(\mathbf{x}, \mathbf{u})$ where \mathbf{x} and \mathbf{u} denote state and control input vectors, respectively.

$$\gamma_{bg} = \arg \max_{L \in [0, L_{max}]} \frac{L}{D} \quad \text{s.t.} \quad \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \quad T = 0 \quad (1)$$

Best glide for a two-minute standard constant radius turn $\gamma_{bg,t} \in \mathbb{R}^+$ under steady wind conditions is as formulated in Equation (2) where bank angle $\mu \neq 0$. An additional inequality constraint [34] on maximum bank angle μ_{max} limits the maximum load factor L/W to meet structural integrity constraints.

$$\gamma_{bg,t} = \arg \max_{C_L \in [0, C_{L,max}]} \frac{L}{D} \quad \text{s.t.} \quad \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \quad T = 0, \quad |\mu| - |\mu_{max}| \leq 0 \quad (2)$$

The best-glide angle is important for any emergency situation where an aircraft needs to descend rapidly, even if the engines are still operational but delivering reduced maximum output.

A good landing begins with a good approach, so it is important to define a stabilized approach speed. Wing flaps are extendable only below maximum flap extended speed v_{FE} . The steepest flight path angle $\gamma_{max} \in \mathbb{R}^+$ for straight gliding (3) and turning (4) flight are identified to assure airspeed v_a is upper bounded by v_{FE} .

$$\gamma_{max} = \min -\gamma \quad \text{s.t.} \quad \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \quad T = 0, \quad v_a - v_{FE} = 0 \quad (3)$$

The steepest gliding angle for turning $\gamma_{max,t} \in \mathbb{R}^+$ is per Equation (3) where $\mu \neq 0$.

$$\gamma_{max,t} = \min -\gamma \quad \text{s.t.} \quad \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \quad T = 0, \quad v_a - v_{FE} = 0, \quad |\mu| - |\mu_{max}| \leq 0 \quad (4)$$

The best-glide and steepest descent flight path angle pair of a Cessna 182 (C182) [35] are defined as functions of wind speed v_w and direction χ_w , and obtained as in [17]. Given no forward thrust ($T = 0$), best-glide angle γ_{bg} represents the shallowest gliding angle an aircraft can fly at without stalling. While aerodynamically efficient, this angle is susceptible to stall due to disturbances such as wind gusts. Conversely, flying at steepest glide angle γ_{max} requires a higher approach speed, leading to a longer deceleration path. To balance safety and efficiency during emergency approaches, flight path angle γ_{opt} is defined per Equation (5) that lies halfway between γ_{bg} and γ_{max} . γ_{opt} offers an optimal safety margin. This value can be adjusted to provide flexibility in the safe landing path descent profile.

$$\begin{aligned} \gamma_{opt}(v_w, \chi_w) &= \frac{1}{2} [\gamma_{bg}(v_w, \chi_w) + \gamma_{max}(v_w, \chi_w)] \\ \gamma_{opt,t}(v_w, \chi_w) &= \frac{1}{2} [\gamma_{bg,t}(v_w, \chi_w) + \gamma_{max,t}(v_w, \chi_w)] \end{aligned} \quad (5)$$

For clarity, flight path angle notation is simplified by omitting their input arguments, i.e., $\gamma = \gamma(v_w, \chi_w)$. Note that all flight path angles are functions of steady wind conditions as used to define aircraft turn radius R per Equation (6) [33] for path planning.

$$R = \frac{v_a^2 \cos \gamma}{g \cos \mu} \quad (6)$$

Here, g and μ are gravitational acceleration and wing bank angle, respectively.

2.2. Reachable Footprint

For emergency landing, potential landing sites must be identified within a reachable area. This area, known as the reachable footprint [28], is determined by the aircraft's capabilities and the specific emergency. For example, a complete loss of thrust limits the reachable footprint to the gliding range, while a medical emergency prioritizes nearby landing sites that are also close to healthcare facilities to minimize total transit time. Various methods have been proposed to determine the reachable area for distressed aircraft landing. Aircraft dynamics analysis [36,37], optimal control theory [38], Hamilton-Jacobi methods [39,40], and attitude optimization [41] have been employed with different cost/benefit trade-offs. In this paper, reachability of the target destination is assumed confirmed to focus this research effort on path planning to a designated site.

2.3. Geometric Aircraft Path Planning

The Dubins path [6] provides a geometric solution for transitioning between two states while respecting a minimum turning radius constraint. Originally formulated for two-dimensional paths, assuming constant altitude, this solution has been extended to three-dimensional aircraft path planning applications [8,42,43]. The literature offers several contingency landing management frameworks with Dubins paths. Ref. [7] modified the kinematic model of Dubins paths to include transient effects such as translational and rotational acceleration in emergency path planning. Ref. [28] extends Dubins paths by incorporating real-time updates for dynamically feasible landing plans under degraded aircraft performance. Dubins path and RRT* are fused in [44] to create emergency landing procedures. Ref. [17] incorporated datalink and Dubins path to coordinate distressed aircraft and ground traffic for road-based emergency landings.

Conventional Dubins path solutions are formed by an initial minimum radius turn segment, a final minimum radius turn segment, and a straight segment connecting tangents of the initial and final turns. There are four options for turn-straight-turn type of paths: right-straight-right (RSR), right-straight-left (RSL), left-straight-right (LSR), and left-straight-left (LSL). Dubins [6] also defined solutions with intermediate turns (RLR and LRL) but these are not practical for long distance traversals between arbitrary configurations. Most Dubins solvers typically evaluate all four path types and select the shortest assuming constant velocity. Path geometry and length derivations can be found in [32].

To enhance discrete search-based path planning, the proposed algorithm incorporates 3D Dubins path solutions. For instance, Dubins paths can be employed to establish a connection between an initial emergency state and a designated holding point, as well as to link a discrete search-generated trajectory to final approach to the landing runway. Also, S-turn Dubins paths [28] are employed for benchmarking.

Aircraft state is defined as a tuple $s = (\varphi, \lambda, h, \chi)$ where $\varphi \in \Phi$, $\lambda \in \Lambda$, $h \in \mathbb{R}_{\geq 0}$, and $\chi \in [0, 2\pi)$ respectively correspond to latitude, longitude, altitude above mean sea level (MSL), and course angle. The symbols Φ and Λ denote valid sets of latitude and longitude. The set of all possible aircraft states S are defined per Equation (7).

$$S = \{(\varphi, \lambda, h, \chi) \mid \varphi \in \Phi, \lambda \in \Lambda, h \in [0, \infty), \chi \in [0, 2\pi)\} \quad (7)$$

Let the initial and final states respectively be $s_0 \in S$ and $s_N \in S$. A contingency landing path $\mathcal{P} \subset S$ consists of a sequence of states from s_0 to s_N where $\mathcal{P} = \{s_0, s_1, \dots, s_N\}$. A set of Dubins path solutions between arbitrary states s_i and s_j are denoted by $\mathcal{D}(s_i, s_j)$ per Equation (8).

$$\mathcal{D}(s_i, s_j) = \{\mathcal{P}_{\text{type}} \mid \text{type} \in \{\text{RSR}, \text{RSL}, \text{LSR}, \text{LSL}\}\} \quad (8)$$

The shortest Dubins solution \mathcal{P}_D is defined in Equation (9) where $\mathcal{L}(\mathcal{P})$ is the length of path \mathcal{P} .

$$\mathcal{P}_D = \begin{cases} \arg \min_{\mathcal{P} \in \mathcal{D}(s_i, s_j)} \mathcal{L}(\mathcal{P}), & \mathcal{D}(s_i, s_j) \neq \emptyset, \\ \emptyset, & \mathcal{D}(s_i, s_j) = \emptyset \end{cases} \quad (9)$$

Note that $\mathcal{D}(s_i, s_j)$; thus \mathcal{P}_D , can be an empty set if s_j is unreachable given s_i .

3. Problem Statement

Ensuring a safe and feasible contingency landing for fixed-wing aircraft in emergency scenarios is a critical problem that demands both efficiency and reliability. The objective of this paper is to present and analyze a contingency landing planner that minimizes risk to the overflowed population while ensuring timely definition of a feasible path. This problem presents several key challenges. First, the feasibility of the landing path is constrained by the distressed aircraft's flight envelope, necessitating a careful assessment of its aerodynamic limitations given degraded performance. Second, minimizing risk along the trajectory requires accurate evaluation and incorporation of risk data. Another fundamental challenge lies in the computational complexity of search-based path planning. The need to explore a continuous search space with theoretically infinite extent complicates real-time decision-making and solution convergence guarantees. This trade-off between runtime efficiency, search completeness, and trajectory risk must be carefully managed to ensure a practical implementation. For this study, the following assumptions are made. First, designated landing sites are assumed to be within the aircraft reachable footprint. Second, this paper assumes a known steady wind throughout the descent to landing.

4. Methodology

This section introduces the contingency landing planner and its building blocks including a search-based path planning algorithm, a solution identification criterion, and a method to define a minimum-risk holding pattern.

4.1. Contingency Landing Path Planning

Because a Dubins solution is purely geometric, risk metrics cannot be considered. A low-flying aircraft in distress poses risk to an overflowed population. However, an aircraft at high altitude poses negligible risk to people on the ground below because it will fly well past its current position before reaching the ground even in a steep descent. Discrete search with a population risk metric is therefore only useful when the aircraft descends below a crossover altitude $h_c \in \mathbb{R}_{>0}$. The handling of high and low altitude cases is illustrated in Figure 2.

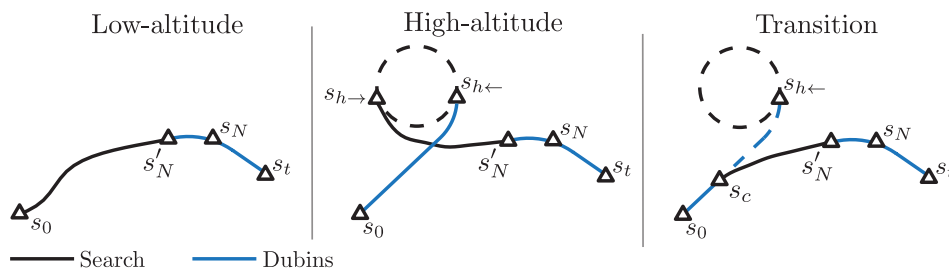


Figure 2. Contingency landing paths for low-altitude, high-altitude, and transition cases. A circular holding pattern is used for high-altitude cases.

In a low-altitude emergency, a discrete search planner generates a low-risk, energy-managed path $\mathcal{P}_s(s_0, s'_N)$ from the initial emergency state s_0 to the final search state s'_N . Final search state s'_N is constrained to lie in a predefined hypervolume around s_N . Then, a final turn to join a final approach path to touchdown state s_t is computed with a 3D Dubins path $\mathcal{P}_D(s'_N, s_t)$ considering s_N as a fly-by waypoint. Note that the exact initial and final states of the Dubins path for the final turn are respectively computed by the Dubins path solver between s'_N and s_N as well as s_N and s_t , to ensure feasibility with respect to aircraft dynamics and kinematic continuity. Formally, a low-altitude solution is defined as:

$$\mathcal{P} = \mathcal{P}_s(s_0, s'_N) \cup \mathcal{P}_D(s'_N, s_t) \quad (10)$$

In a high-altitude emergency, the distressed aircraft must descend at a flight path angle near γ_{opt} before initiating an approach to landing. Altitude dissipation is achieved by defining a circular holding pattern with inbound state $s_{h\leftarrow}$ and outbound state $s_{h\rightarrow}$. The planner identifies the optimal holding pattern location using a utility function specified below. It connects s_0 to $s_{h\leftarrow}$ with the shortest Dubins path $\mathcal{P}_D(s_0, s_{h\leftarrow})$. The holding pattern is traversed until reaching $s_{h\rightarrow}$ where discrete search is initiated to find a low-risk low-altitude landing path to s'_N . A high-altitude solution can be expressed as:

$$\mathcal{P} = \mathcal{P}_D(s_0, s_{h\leftarrow}) \cup \mathcal{P}_s(s_{h\rightarrow}, s'_N) \cup \mathcal{P}_D(s'_N, s_t) \quad (11)$$

A transition case arises if $h_0 > h_c$ yet $s_{h\leftarrow}$ is unreachable above h_c . Similar to high-altitude cases, a Dubins path is generated from s_0 to $s_{h\leftarrow}$. The crossover state $s_c \in \mathcal{P}_D(s_0, s_{h\leftarrow})$ at the crossover altitude on the path to $s_{h\leftarrow}$ is identified. The search-based planner then generates a low-risk path from s_c to s'_N . Then, the solution is finalized with a 3D Dubins path $\mathcal{P}_D(s'_N, s_t)$. Solution of a transition case is given by:

$$\mathcal{P} = \mathcal{P}_D(s_0, s_c) \cup \mathcal{P}_s(s_c, s'_N) \cup \mathcal{P}_D(s'_N, s_t) \quad (12)$$

4.2. Search-Based Path Planning

An aircraft flight envelope defines its safe operating limits. These limits include but are not limited to stall for controllability, load factor and maximum airspeed for structural integrity. Let degraded aircraft flight envelope \mathbb{A} be defined as:

$$\mathbb{A} = \{(v_a, R, \gamma, \gamma_t) \mid \text{all within safe limits}\} \quad (13)$$

For a given aircraft flight envelope \mathbb{A} , s_0 , and s_N , the search space per Equation (7) can be identified using a reachability method. In this study, \mathcal{S} is an infinite, continuous, and non-Euclidean space where the distance between arbitrary states are geodesic distances. Feasible actions for transition from a parent state to successor states are given by $a(s) = \{a^j(s) \mid j \in \mathbb{Z}_{>0}\}$ where $[j]$ is the branching factor. Each action a^j is composed of course angle change $\Delta\chi$, path segment length which is great-circle distance between the parent and successor state ℓ , and optimal straight gliding and turning flight path angles along segment γ and γ_t given a known steady wind.

$$a^j(s) = (\Delta\chi, \ell, \gamma, \gamma_t \mid \Delta\chi \in \mathcal{X}, \ell \in \mathbb{R}^+, \gamma \in \mathbb{R}^+, \gamma_t \in \mathbb{R}^+) \quad (14)$$

where $\mathcal{X} \subseteq [0, 2\pi]$ is a set of feasible course angle changes. The set of all feasible actions \mathcal{A} is per Equation (15) where $|\mathcal{A}| < \infty$.

$$\mathcal{A} = \{a(s) \mid s \in \mathcal{S}\} \quad (15)$$

For a given aircraft flight envelope \mathbb{A} , set of actions \mathcal{A} , initial and goal states s_0 and s_N , state cost function $g(s)$, and steady wind with magnitude v_w and direction χ_w , the 4D search-based path planning problem \mathcal{T} is formulated as in Equation (16) where \mathcal{P}_s is a non-ascending emergency landing path solution.

$$\mathcal{P}_s = \{s \mid s = \mathcal{T}(\mathbb{A}, \mathcal{A}, g(s), s_0, s_N, v_w, \chi_w)\} \quad (16)$$

4.3. Feasible Actions

The action set must consist only of feasible actions considering potentially degraded flight envelope constraints. For example, while symmetrical lateral maneuverability can be used for left and right turns considering airspeed and load factor for a single-engine aircraft experiencing loss-of-thrust, asymmetrical course angle rate limits will be present for aircraft with deteriorated lateral controllability caused by conditions such as control surface jam [45], structural damage [46], and uneven icing [47]. The minimum segment length for a given arbitrary set of course angle changes is further defined below.

A sketch of sequential turns in the horizontal plane is shown in Figure 3. The minimum segment length ℓ_{min} that can accommodate sequential turns between state s_i and s_{i+1} is expressed by Equation (17). When expanding state s_i , the next state and action are unknown; hence, $\Delta\chi_{i+1}$ can be substituted by the highest possible course change value to ensure feasibility.

$$\ell_{min} = R \left(\tan \frac{\Delta\chi_i}{2} + \tan \frac{\Delta\chi_{i+1}}{2} \right) \quad (17)$$

The search algorithm uses an adaptive segment length (18) that reduces search resolution at higher altitudes where ground risk is minimal to improve real-time performance. This function uses h_i , the altitude at s_i , $\underline{h} \in \mathbb{R}^+$, the floor altitude change at the minimum segment length, and $m \in \mathbb{R}^+$, the rate of segment length change per unit altitude.

$$\ell_i = \begin{cases} \ell_{min} + m(h_i - \underline{h}), & h_i > \underline{h} \\ \ell_{min}, & h_i \leq \underline{h} \end{cases} \quad (18)$$

Per Equation (5), γ_{opt} and $\gamma_{opt,t}$ are determined within \mathbb{A} for given steady wind conditions. Therefore, a feasible action a^j at state s is a function of wind speed v_w and wind direction χ_w .

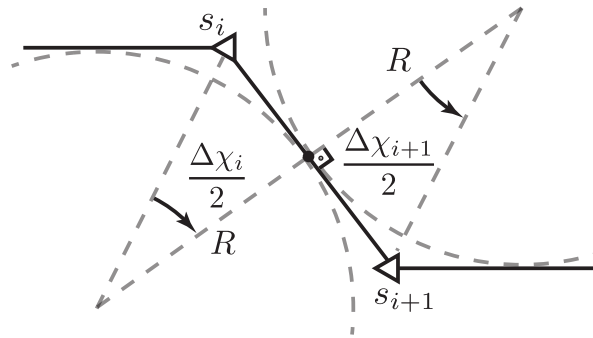


Figure 3. Planar geometry of sequential turns.

4.4. State Expansions

Successor states are generated by expanding parent states. A successor of parent s resulted from j -th feasible action a^j of s is denoted by $s^j = (\varphi^j, \lambda^j, h^j, \chi^j)$. Course angle of a successor state χ^j is

$$\chi^j = |\chi + \Delta\chi^j|_{2\pi} \quad (19)$$

where $\chi \in s$ and $\Delta\chi^j \in a^j$. The operator $|\cdot|_\alpha$ wraps an angle at α . Coordinates of a successor state are computed by using World Geodetic System 1984 (WGS84) [48] model $\mathcal{W} : \mathbb{R}_{>0} \times \mathbb{R} \times \mathcal{S} \rightarrow \Phi \times \Lambda$ per Equation (20). Let $c(s) : \mathcal{S} \rightarrow \Phi \times \Lambda$ be an operator such that $c(s) = [\varphi \ \lambda]^\top$. Simply, \mathcal{W} is a function that returns the destination coordinates given a reference state and action variables.

$$c(s^j) = \mathcal{W}(\ell^j, \chi^j | s) \quad (20)$$

The altitude of the successor state h^j (21) is computed considering turning and straight gliding traversals where $\gamma^j \in a^j$.

$$h^j = h - \ell_{turn} \tan \gamma_{opt,t}^j - (\ell - \ell_{turn}) \tan \gamma_{opt}^j \quad (21)$$

Turning traversal is denoted by ℓ_{turn} and calculated as in Equation (17). For convenience, the transition model $E : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is specified to apply an action to a state, which composes relations (19)-(21). The set of successors for parent state s is

$$C(s) = \{s^j | s^j = E(s | a^j), a^j \in \mathcal{A}\}. \quad (22)$$

Reciprocally, the parent of a successor state is $s = \pi(s^j)$.

4.5. Cost Functions with a Constraint Margin Gradient Field

The path planning cost function is constructed by four distinct terms: (1) a gradient function to prioritize state expansion with an optimal descent path angle, (2) a gradient function to guide search toward the target location, (3) a gradient function that guides search toward the desired course angle as the target location is approached, and (4) a term that discourages flight over densely populated urban areas. The total cost of a state $g(s)$ is given in Equation (23).

$$g(s) = w_1[w_d(s)g_{d,1}(s)] + w_2[(1 - w_d(s))g_{d,2}(s)] + w_3g_\chi(s) + w_4g_p(s), \quad \sum_{i=1}^4 w_i = 1 \quad (23)$$

Here, $g_{d,1}$, $g_{d,2}$, g_χ , and g_p are respectively optimal descent, direct distance, course angle, and population cost functions. Terms $w_i \in \mathbb{R}_{\geq 0}$ are weight coefficients while $w_d(s) : \mathcal{S} \rightarrow [0, 1]$ is an adaptive weighting factor. First, common terms used in cost functions are introduced. An approximate shortest path length to the goal is calculated based on a combination of turn and straight sections.

$$d_{min}(s) = d_{gc}(s) + R|\theta(s, s_N) - \chi|\pi \quad (24)$$

The function $d_{gc}(s) : \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$ represents the great-circle distance from state s to s_N , where $\theta : \mathcal{S} \times \mathcal{S} \rightarrow [0, 2\pi)$ returns the bearing angle between two states. The remaining best-glide range $d_{bg}(s) \in \mathbb{R}_{\geq 0}$ given the altitude loss from s to s_N is defined as:

$$d_{bg}(s) = \frac{h - h_N}{\tan \gamma_{bg}} \quad (25)$$

States with $d_{bg}(s) < d_{min}(s)$ indicates unreachability and are pruned from further expansion. d_{opt} is defined as an approximate optimal distance to be flown to descend to goal altitude h_N from initial altitude h_0 at optimal glide slope γ_{opt} initially estimated based on the relative wind between s_0 and s_N .

$$d_{opt} = \frac{h_0 - h_N}{\tan \gamma_{opt}} \quad (26)$$

4.5.1. Optimal Gliding Cost

Two cost functions are employed to encourage progress toward the goal state region. The first function, $g_{d,1}(s) : \mathcal{S} \rightarrow [0, 1]$ (27), prioritizes the necessary altitude loss to reach the goal state. It serves as a measure of how efficiently the estimated total path length (the first two terms in the numerator) compares to the optimal descent path length. Consequently, it penalizes deviations from the optimal path length caused by deviation from the optimal flight path angle using the heuristic $d_{min}(s)$.

$$g_{d,1}(s) = \frac{|\ell(s) + d_{min}(s) - d_{opt}|}{d_{opt}} \quad (27)$$

Here, $\ell : \mathcal{S} \cup \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is the total path length leading s_0 to s as a function of either state s or time t to reach s per Equation (28).

$$\ell(x) = \begin{cases} \ell_s(s), & x \in \mathbb{R}^4, \\ \ell_t(t), & x \in \mathbb{R}_{\geq 0}. \end{cases} \quad (28)$$

The cost $g_{d,1}(s)$ is proven bounded per Lemma 1.

Lemma 1. For a non-ascending landing path with $\gamma_{bg} \leq \gamma$, and an aircraft whose optimal glide angle satisfies $\tan \gamma_{opt} \leq 2 \tan \gamma_{bg}$, the cost function $g_{d,1}(s)$ is bounded within the closed interval $[0, 1]$, $\forall s \in \mathcal{S}$.

Proof of Lemma 1. The boundedness of $g_{d,1}(s)$ can be established by analyzing the limits of $\ell(s) + d_{min}(s)$, given that d_{opt} is a constant. Given the path slope γ , $\gamma_{bg} \leq \gamma$, a feasible landing path cannot exceed the best-glide range d_{bg} ; hence, the upper bound is:

$$\ell(s) + d_{min}(s) \leq \frac{h_0 - h_N}{\tan \gamma_{bg}} = d_{bg}. \quad (29)$$

Defining $\Delta h = h_0 - h_N$ and substituting d_{bg} into the expression for $g_{d,1}(s)$, one obtains:

$$\begin{aligned} g_{d,1}(s) &= \frac{|\ell(s) + d_{min}(s) - d_{opt}|}{d_{opt}} \\ &\leq \frac{|d_{bg} - d_{opt}|}{d_{opt}} \\ &= \frac{\Delta h / \tan \gamma_{bg} - \Delta h / \tan \gamma_{opt}}{\Delta h / \tan \gamma_{opt}} \\ &= \frac{\tan \gamma_{opt}}{\tan \gamma_{bg}} - 1. \end{aligned} \quad (30)$$

Given aircraft flight envelope with $\tan \gamma_{opt} \leq 2 \tan \gamma_{bg}$, it follows that:

$$\frac{\tan \gamma_{opt}}{\tan \gamma_{bg}} - 1 \leq 1 \quad (31)$$

Thus, $g_{d,1}(s) \leq 1$ is found. Next, the lower bound is examined. The quantity $\ell(s) + d_{min}(s)$ attains its minimum if and only if the initial and goal states coincide in latitude and longitude, i.e., $c(s_0) = c(s_N)$. Given these conditions, $\ell(s) = 0$ and $d_{min}(s) = 0$, leading to:

$$g_{d,1}(s) = \frac{|-d_{opt}|}{d_{opt}} = 1 \quad (32)$$

By equations (31) and (32), $g_{d,1}(s)$ is found upper bounded by 1. Noting that $g_{d,1}(s)$ is non-negative due to the absolute value term in its definition, it is concluded that:

$$g_{d,1}(s) \in [0, 1], \quad \forall s \in \mathcal{S}. \quad (33)$$

Thus, $g_{d,1}(s)$ is bounded within the closed interval $[0, 1]$, completing the proof. \square

4.5.2. Direct Distance Cost

Using only $g_{d,1}(s)$ ensures the path follows the optimal length using a heuristic, but it can lead to wasted potential energy by expanding nodes away from the goal, particularly in high-altitude emergency cases. To address this, a second function, $g_{d,2}(s) : \mathcal{S} \rightarrow [0, 1]$, a heuristic estimate for the shortest path to the goal normalized by d_{opt} , is introduced per Equation (34). This function prioritizes states that are closer to the goal regardless of their descent profile.

$$g_{d,2}(s) = \frac{d_{min}(s)}{d_{opt}} \quad (34)$$

Due to reachability limitations, the optimal flight path angle γ_{opt} cannot be determined per Equation (5) if $d_{min}(s_0) > d_{opt}$. In such cases, a shallower flight path angle - optimal for the specific emergency scenario - must be used to ensure reachability such that $d_{bg}(s_0) \leq d_{min}(s_0) \leq d_{opt}$. Therefore, the condition $0 \leq d_{min}(s) \leq d_{opt} < \infty, \forall s \in \mathcal{S}$ inherently guarantees the boundedness of $g_{d,2}(s)$ on closed interval $[0, 1]$, eliminating the need for a separate proof.

By weighting $g_{d,1}(s)$ and $g_{d,2}(s)$, prioritization of expanding states towards the goal and establishment of the optimal descent profile are balanced. To achieve this balance, upper and lower bound radii around s_N , respectively \bar{r} and \underline{r} , are specified to calculate $w_d(s)$ to limit outward branching. Equation (35) defines \bar{r} while \underline{r} is a predetermined constant.

$$\bar{r} = \frac{1}{2}(d_{opt} + d_{min}(s_0)) \quad (35)$$

The weighting factor is

$$w_d(s) = \begin{cases} 0, & d_{min}(s) \geq \bar{r} \\ \frac{\bar{r} - d_{min}(s)}{\bar{r} - \underline{r}}, & \underline{r} < d_{min}(s) < \bar{r} \\ 1, & d_{min}(s) \leq \underline{r} \end{cases} \quad (36)$$

Equation (36) guides state expansion toward s_N by adjusting the influence of $g_{d,1}(s)$ and $g_{d,2}(s)$ based on the state's location in the search space. Specifically, according to Equation (23), when s is farther than \bar{r} from s_N , $g_{d,1}(s)$ is set to zero, preventing outward branching and conserving excess altitude. This allows the aircraft to later use its potential energy to navigate around densely populated areas. Conversely, when s is closer than \underline{r} to s_N , $g_{d,2}(s)$ is eliminated, prioritizing the optimal descent path by managing altitude loss.

To facilitate further analysis, let $g_d(s)$ be:

$$g_d(s) = w_d(s)g_{d,1}(s) + (1 - w_d(s))g_{d,2}(s) \quad (37)$$

Now, the boundedness of $g_d(s)$ is established in Theorem 1.

Theorem 1. *The weighted sum of $g_{d,1}(s)$ and $g_{d,2}(s)$ is bounded within the closed interval $[0, 1]$, $\forall s \in \mathcal{S}$.*

Proof of Theorem 1. By Lemma 1, $g_{d,1}(s)$ is bounded within $[0, 1]$. Additionally, by definition, both $g_{d,2}(s)$ and $w_d(s)$ are constrained within $[0, 1]$. Since $g_d(s)$ is a convex combination of these bounded functions, it follows that $g_d(s)$ remains within $[0, 1]$, $\forall s \in \mathcal{S}$. The constant weights w_1 and w_2 do not alter the boundedness of the linear combination. \square

Figure 4 demonstrates the evolution of $g_d(s)$ during path planning. The vector field visualizes the gradient of $g_d(s)$, $\nabla g_d(s) = [\partial g_d(s)/\partial \lambda \quad \partial g_d(s)/\partial \phi]^T$, indicating the preferred direction of state expansion.

Initially, in the leftmost figure where $\ell(s) = 0$ NM, the vector field around state s points away from the goal state. This outward direction of state expansion occurs as $g_d(s)$ forms a repellent, high-cost region around goal state s_N due to $d_{opt} > d_{min}(s_0)$. As the path planning progresses, the search captures and follows the optimal decent path as in the second and third figures from the left. Finally, it converges toward s_N by following the minimum cost isoline as depicted in the rightmost figures. If the initial altitude were set lower, the minimum cost region would lie between s_0 and s_N due to $d_{opt} \approx d_{bg}$, leading state expansion immediately toward s_N .

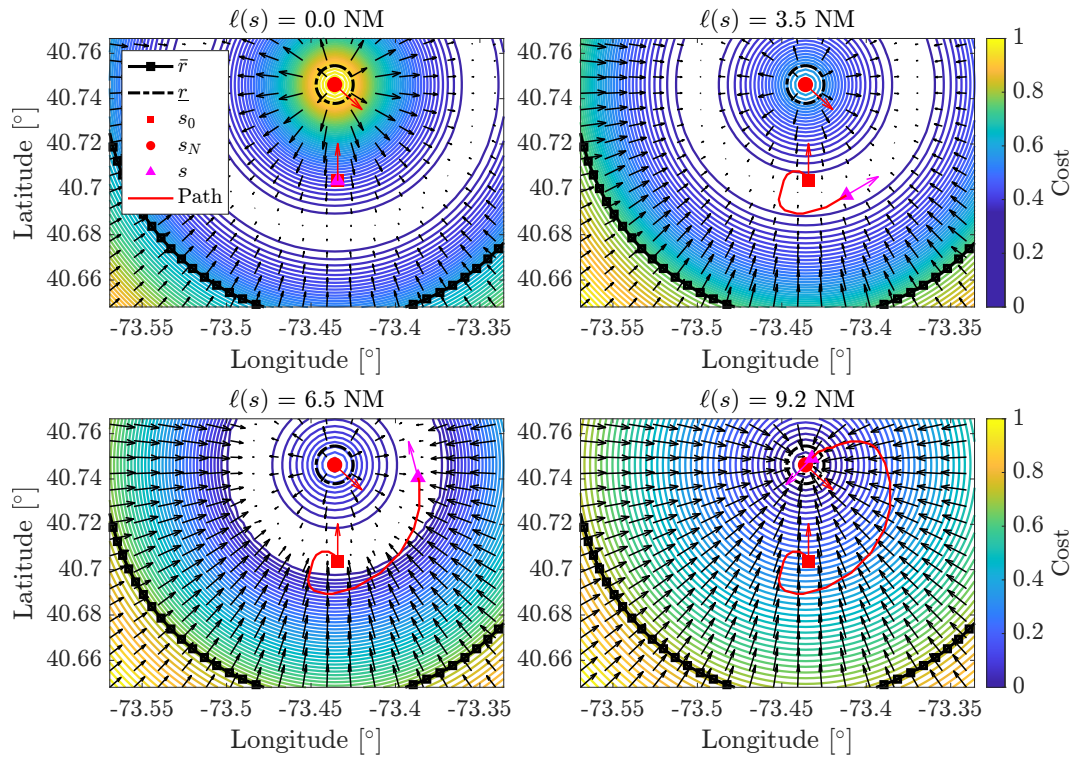


Figure 4. An evolution of $g_d(s)$ with a vector field around the goal state. Red arrows show course angle.

4.5.3. Course Angle Cost

The costs $g_{d_1}(s)$ and $g_{d_2}(s)$ enable search to reach the goal region yet they do not reward the desired course angle. It is challenging to match the exact target course with a discrete search. Explicitly incorporating the course angle difference in a cost function would eventually enforce states to only expand parallel to the goal course which impedes necessary turns to match the goal location. Thus, a course angle cost that represents the preferred direction for approaching the goal state in lieu of exactly matching the landing runway course angle is proposed.

A unit normal vector \vec{n}_{χ_N} represented in a North-East-Down (NED) coordinate system and aligned with the target course χ_N is used to partition the cost map into regions of higher and lower cost per Equation (38) where $\mathcal{R}_{(\varphi,\lambda)}^{NED} : \Phi \times \Lambda \rightarrow \mathbb{R} \times \mathbb{R}$ is the transformation operator from geodetic to NED coordinate system.

$$\tilde{g}_{\chi}(s) = \vec{n}_{\chi_N} \cdot \left[\mathcal{R}_{(\varphi,\lambda)}^{NED}(c(s)) - \mathcal{R}_{(\varphi,\lambda)}^{NED}(c(s_N)) \right] \quad (38)$$

It should be noted that the vector operations in Equation (38) must be in a local coordinate system to eliminate asymmetrical gradient field with respect to χ_N due to the Earth's curvature. The scalar product in Equation (38) evaluates the projection of the vector from an arbitrary state s to s_N onto \vec{n}_{χ_N} . A $\tilde{g}_{\chi}(s)$ value of zero indicates that s lies on the line perpendicular to \vec{n}_{χ_N} that passes through $c(s_N)$, bisecting the search map with a vertical plane. Such states require a 90° turn to align with the final course, a common practice in airport traffic patterns for general aviation aircraft. States with $\tilde{g}_{\chi}(s) < 0$ are located in the region behind the bisecting line and require a course change of less than 90° to reach the goal. Conversely, states with $\tilde{g}_{\chi}(s) > 0$ are penalized, as they necessitate a course angle change greater than 90° to align with the landing site. Equation (38) is exploited to further define the course angle cost $g_{\chi}(s)$ per Equation 39 where $\kappa_{\chi} > 0$ is a normalization constant to ensure the cost is bounded on the closed interval $[0, 1]$. The scalar product term is multiplied by distance terms to ensure that distant states can be explored in any direction, while nearby states are restricted to the

region behind the bisecting line. The $\max(\cdot, \cdot)$ operator is introduced to zero out the course angle cost of states with $\tilde{g}_\chi(s) < 0$.

$$g_\chi(s) = \frac{1}{\kappa_\chi} \max\left(0, \tilde{g}_\chi(s) \frac{d_{opt}}{d_{gc}(s)}\right) \quad (39)$$

The normalizer κ_χ can easily be determined based on the maximum course cost value in the search space \mathcal{S} per Equation (40).

$$\kappa_\chi = \sup_{s \in \mathcal{S}} \tilde{g}_\chi(s) \frac{d_{opt}}{d_{gc}(s)} \quad (40)$$

Overall, the $\max(\cdot)$ operator ensures a lower bound of zero while κ_χ normalizes to the upper bound to one, simplifying the need for a boundedness proof for $g_\chi(s)$ on the closed interval $[0, 1]$.

An illustration of Equation (39) for $\chi_N = 132^\circ$ is shown in Figure 5. The green arrow points in the goal course angle direction, and the vector field of $\partial g_\chi = [\partial g_\chi / \partial \lambda \quad \partial g_\chi / \partial \varphi]^\top$ encourages the search algorithm to expand states toward regions where $g_\chi(s) = 0$, enabling a feasible final approach.

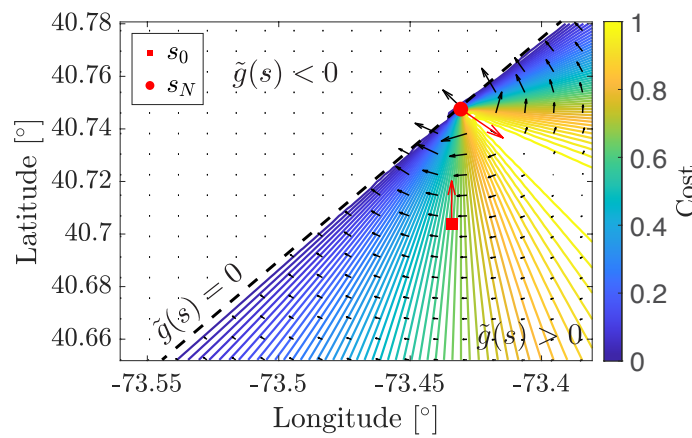


Figure 5. A map of the course angle cost. The black dashed line illustrates the state bisecting plane. Red arrows show course angle.

The function $g_\chi(s)$ alone is insufficient to guide the search to the solution, as it only provides guidance for one half of the horizontal cost map. However, stacking $g_\chi(s)$ on top of $g_{d,1}(s)$ and $g_{d,2}(s)$ alters the gradient field direction in the vicinity of the goal state, allowing state expansion in the right direction.

The combined $g_{d,1}(s)$, $g_{d,2}(s)$, and $g_\chi(s)$ are depicted in Figure 6. In contrast to the solution shown in Figure 4, the state expansion crosses the bisecting line from $\tilde{g}_\chi(s) > 0$ to $\tilde{g}_\chi(s) < 0$ while simultaneously capturing the optimal descent path.

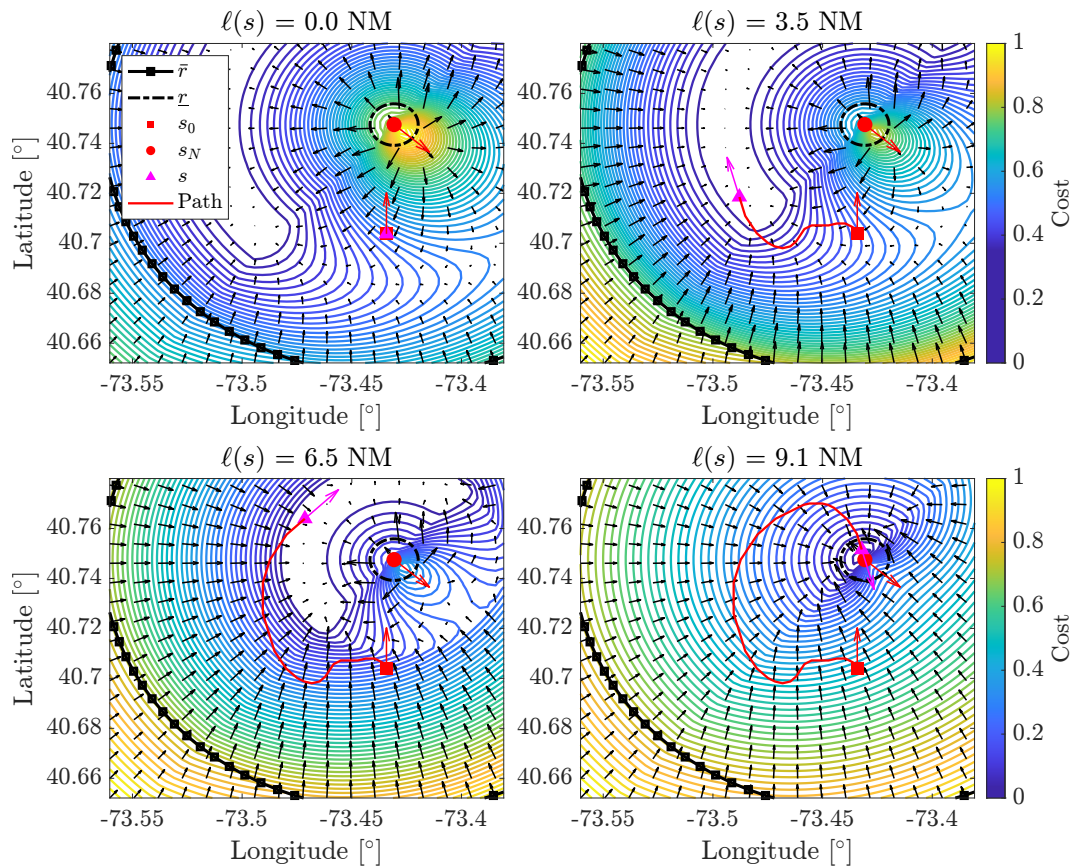


Figure 6. Evolution of the total distance and course angle costs with a vector field around a goal state. Red arrows show course angle.

4.5.4. Population Cost

Population cost is computed by integrating the overflowed population density over time. Let the population density function be $\tilde{\eta} : \Phi \times \Lambda \rightarrow \mathbb{R}_{\geq 0}$. The function $\tilde{\eta}$ is reformulated as a function of time for a known state per Equation (41), where \circ is the composition operator. In this paper, the population and land area dataset shared by the U.S. Census Bureau [49] is utilized to obtain population density as a function of time $\eta(t | s, v)$ which will be denoted by $\eta(t)$ for convenience. For population risk estimation, wind effect is ignored; thus, $v = v_a$.

$$\eta(t | s, v) = \tilde{\eta} \circ \mathcal{W}(vt, \chi | s) \text{ persons/m}^2 \quad (41)$$

To reflect true population risk, a weighting function $w_{p,1}(s)$ (42) is introduced to penalize η under a lower altitude path.

$$w_{p,1}(t) = \begin{cases} 1 - \frac{h(t)}{\min(h_0, h_c)}, & h(t) \leq h_c \\ 0, & h(t) > h_c \end{cases} \quad (42)$$

Here, the crossover altitude h_c is used for the ceiling altitude for population risk consideration. Therefore, population risk is not taken into account for path planning above h_c .

In case of an emergency, the population residing under a given destination is not a factor as the aircraft is supposed to be approaching and landing at an airport runway in this work. Thus, the impact of population risk is reduced proportionally to the remaining traversal using a quadratic cost $w_{p,2}(t)$ 43.

$$w_{p,2}(t) = \left(1 - \frac{\ell(t)}{d_{opt}}\right)^2 \quad (43)$$

Time t is the input argument to Equation (28) for integration purposes.

Let the cumulative overflowed population risk to reach s^j from s in flight time t be the integration of weighted (41) with respect to time per Equation (44).

$$g_p(s^j) = \frac{1}{\kappa_p} \int_0^t w_{p,1}(t) w_{p,2}(t) \eta(t) dt \quad (44)$$

Here, the flight time t is estimated by ℓ^j/v . The boundedness of $g_p(s)$ depends on the proper scaling by $\kappa_p \in \mathbb{R}_{\geq 0}$ (45).

$$\kappa_p = \kappa_w \kappa_\eta t \quad (45)$$

Per Equation (45), the population risk is time averaged and normalized by the upper bound of the weight product $\kappa_w = \sup w_{p,1}(t) w_{p,1}(t)$ and maximum population density κ_η within the search space \mathcal{S} (46).

$$\kappa_\eta = \sup_{s \in \mathcal{S}} \tilde{\eta}(c(s)) \quad (46)$$

The magnitudes of each cost term must be normalized and/or appropriately weighted in path planning for an unbiased solution. To prove the boundedness of $g_p(s)$ in numerical implementations, two lemmas are provided.

Lemma 2. Population density function $\eta(t)$ is Riemann integrable on $[0, T]$.

Proof of Lemma 2. A function is Riemann integrable on a closed interval $[0, T]$ if and only if it is bounded and the lower and upper integrals equal to each other [50].

First, the boundedness of $\eta(t)$ is established. Since $\eta(t)$ represents population density, a physical quantity, it is inherently bounded over finite time interval $[0, T]$. Next, the discontinuities of $\eta(t)$ are taken into account. By its nature, $\eta(t)$ is piecewise continuous. Consider a partition $P = \{t_0, t_1, \dots, t_n\}$ on $[0, T]$ such that $0 = t_0 < t_1 < \dots < t_n = T$. The norm of the partition P is

$$\|P\| = \max_{0 < i < n} \{t_i - t_{i-1}\} \quad (47)$$

Let the lower sum $\underline{S}(P, \eta(t))$ and upper sum $\bar{S}(P, \eta(t))$ of $\eta(t)$ given the partition P be

$$\begin{aligned} \underline{S}(P, \eta(t)) &= \sum_{i=1}^n \inf\{\eta(t) \mid t_{i-1} < t < t_i\} (t_i - t_{i-1}) \\ \bar{S}(P, \eta(t)) &= \sum_{i=1}^n \sup\{\eta(t) \mid t_{i-1} < t < t_i\} (t_i - t_{i-1}) \end{aligned} \quad (48)$$

The function $\eta(t)$ has a finite number of discontinuities corresponding to discrete transitions (e.g., entering or leaving census blocks). Each discontinuity has zero width. Therefore, it is deduced that

$$\lim_{\|P\| \rightarrow 0} \underline{S}(P, \eta(t)) = \lim_{\|P\| \rightarrow 0} \bar{S}(P, \eta(t)) = \int_0^T \eta(t) dt \quad (49)$$

$\eta(t)$ is Riemann integrable on $[0, T]$. \square

Lemma 3. There exists a point $\ell^*(t) \in (0, d_{opt})$ at which the product $w_{p,1}(t) w_{p,2}(t)$ attains its maximum value, denoted as $\kappa_w \in \mathbb{R}_{\geq 0}$, for $h_0 \leq h_c$.

Proof of Lemma 3. Substituting $h(t) = h_0 - \ell(t) \tan \gamma$ into $w_{p,1}(t)w_{p,2}(t)$, the product is expressed in terms of $\ell(t)$. To find its maximum, one can take the derivative and equate it to zero:

$$0 = \frac{\partial}{\partial \ell(t)} w_{p,1}(t)w_{p,2}(t) = \left(1 - \frac{\ell(t)}{d_{opt}}\right) \left(\frac{\tan \gamma}{h_0}\right) \left(\frac{d_{opt} - 3\ell(t)}{d_{opt}}\right) \quad (50)$$

Solving for $\ell^*(t)$:

$$\ell^*(t) = \frac{d_{opt}}{3} \quad (51)$$

Substituting $\ell^*(t)$ into $w_{p,1}(t)w_{p,2}(t)$, obtain the upper bound κ_w is obtained:

$$\kappa_w = \frac{4}{27} \frac{d_{opt} \tan \gamma}{h_0} \quad (52)$$

Thus, $w_{p,1}(t)w_{p,2}(t)$ is bounded by κ_w , completing the proof. \square

Theorem 2. The population cost $g_p(s)$ is bounded on the closed interval $[0, 1]$, $\forall s \in \mathcal{S}$.

Proof of Theorem 2. Since $w_{p,1}(t)$, $w_{p,2}(t)$, and $\eta(t)$ are nonnegative, it is concluded that $g_p(s) \geq 0$. To find the upper bound, consider the magnitude of (44).

$$|g_p(s)| = \left| \frac{1}{\kappa_p} \int_0^t w_{p,1}(t)w_{p,2}(t)\eta(t)dt \right| \quad (53)$$

The function $g_p(s)$ is Riemann integrable per Lemma 2; thus, the triangular inequality for integrals can be applied to Equation (53) [51]:

$$|g_p(s)| \leq \frac{1}{\kappa_p} \int_0^t |w_{p,1}(t)w_{p,2}(t)\eta(t)|dt \quad (54)$$

Using the Cauchy-Schwarz inequality,

$$|g_p(s)| \leq \frac{1}{\kappa_p} \int_0^t |w_{p,1}(t)w_{p,2}(t)| |\eta(t)|dt \quad (55)$$

Per Lemma 3 and Equation (46),

$$|g_p(s)| \leq \frac{1}{\kappa_p} \int_0^t \kappa_w \kappa_\eta dt = \frac{1}{\kappa_p} \kappa_w \kappa_\eta t = 1 \quad (56)$$

\square

This study implements the R*-Tree [52], a spatial indexing algorithm, to efficiently query population density in real time. The process begins by assigning a unique identification number (ID) to each polygon (i.e., census block) in the dataset. These polygons are then organized hierarchically in an R*-Tree. The intermediate nodes store bounding boxes that enclose multiple polygons, allowing for efficient spatial pruning while the leaf nodes contain the smallest bounding boxes along with the corresponding polygon IDs.

To perform a query, the R*-Tree first retrieves the smallest bounding box that contains the query coordinate, quickly eliminating large portions of the dataset. Then, a ray-casting algorithm is used to precisely determine which polygon contains the query point among the remaining candidates. Finally, once the correct polygon is identified, its population and land area values are retrieved from a lookup table using the polygon's ID. The normalized density values are then used in trapezoidal integration. This approach significantly reduces the computational cost of population density queries, making real-time overflowed population risk estimation feasible.

4.6. Feasible Solution Identification for Discrete Search

In a 4D discrete search space, reaching an exact goal state s_N cannot be guaranteed. Therefore, there are two parts of the proposed search algorithm: (1) discrete search to a feasible region around a goal state to identify s'_N , and (2) a Dubins solver that can always find a geometric solution from the end of search path to finalize the landing path. Let $\mathcal{Z} \subseteq \mathcal{S}$ be the solution identification set, defining the set of feasible final search states. A state s is considered final, s'_N , if and only if the following conditions are met:

1. First, $d_{gc}(s)$ must fall within a defined annulus per Equation (57).

$$R_i \leq d_{gc}(s) \leq R_o \quad \text{s.t.} \quad 2R_o \geq \ell \quad (57)$$

This annulus, centered on s_N , is independent of altitude. Ensuring that the outer diameter $2R_o$ is at least the length of one discrete search segment ℓ guarantees state expansion within the outer circle. The inner radius R_i ensures the feasibility of a Dubins path connecting the search solution to the final approach.

2. Second, the flight path angle of the remaining traversal γ_{rem} from s'_N to s_N should adhere to the constraints specified in Equation (58).

$$\gamma_{bg} \leq \tan^{-1} \frac{h_N - h'_N}{d_{gc}(s)} \leq \gamma_{max} \quad (58)$$

This constraint introduces altitude bounds to \mathcal{Z} .

3. Third, s must lie behind s_N such that $g_\chi(s) = 0$. This effectively excludes states requiring a significant course angle change to join the final approach.
4. Four, a Dubins path must be feasible from s'_N to s_t .

Formally, the solution identification volume \mathcal{Z} is per Equation (59).

$$\mathcal{Z} = \{s \mid s \in \mathcal{S}, R_i \leq d_{gc}(s) \leq R_o, \gamma_{bg} \leq \gamma_{rem} \leq \gamma_{max}, g_\chi(s) = 0, \mathcal{D}(s, s_t) \neq \emptyset\} \quad (59)$$

The first and third conditions are treated as hard constraints, while modifications to the flight paths are permitted if the second and fourth conditions are not met. If the flight path angle constraints are satisfied, the optimal flight path angle remains unchanged. Otherwise, the gliding angles of both the discrete search and Dubins solutions must be adjusted. However, any such adjustments must ensure that the resulting flight path angle remains within the aircraft's flight envelope for all segments of the flight. An example of solution identification set is demonstrated in Figure 7. The volume between steepest-glide and best-glide surfaces defines the states of \mathcal{Z} which holds potential final search state s'_N .

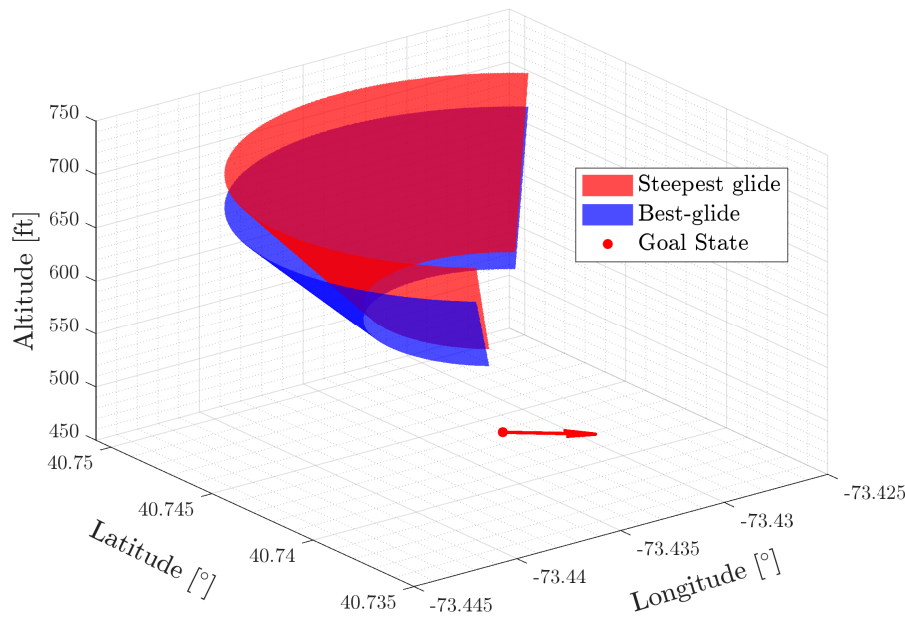


Figure 7. A representative solution identification set.

4.7. Minimum-Risk Holding Pattern Identification

Optimal holding points identification for aircraft emergencies is based on leveraging census population data and geospatial analysis. This process also ensures compliance with safety constraints, such as avoidance of no-fly zones within search space $\mathcal{N} \subset \mathcal{S}$ and sufficient distance from the goal to enable maneuvering. The developed approach has offline and online parts. The offline part begins by defining the circular area around the goal state, \mathcal{H} (60), to identify candidate holding patterns, considering optimal glide performance.

$$\mathcal{H} = \{\Delta p \mid \Delta p = \bigcup_{i=1}^n e_i, e_i = \{(\varphi_i, \lambda_i), (\varphi_j, \lambda_j)\}, A_{\Delta p} > \pi R^2, \Delta p \cap \mathcal{N} = \emptyset\} \quad (60)$$

Here, \mathcal{H} is a set of polygons Δp where e is an edge of polygon with n edges. Each edge is defined by two vertices in latitude and longitude. The algorithm merges small, scattered polygons into larger polygons with area $A_{\Delta p}$, ensuring that merged areas meet a minimum threshold defined by the aircraft turn radius so that the holding pattern is completely situated inside the polygon. It is worth mentioning that simply filtering out polygons with the minimum area constraint does not necessarily ensure that the remaining polygons can accommodate a holding pattern with a minimum radius of R due to their arbitrary shapes. However, this heuristic step prunes polygons that are unlikely to fit the required holding pattern, thereby reducing online computational overhead. A significant aspect of \mathcal{H} is that it can be processed offline and retrieved in real-time from the flight computer.

The online steps of holding point identification include a real-time optimization to find holding pattern center coordinates and utility calculation over candidate holding points. Let a circular left-turn holding pattern be identified with centering state $s_h \in \Delta p$ and $\Delta p \in \mathcal{H}$. A holding pattern is evaluated with a utility function per Equation (61). The maximum values of these metrics are found over the set \mathcal{H} .

$$U(s_h) = \frac{\eta_{\max} - \tilde{\eta}(c(s_h))}{\eta_{\max}} + \frac{d_{\max} - d_{gc}(s_h)}{d_{\max}} + \frac{r_h(s_h)}{r_{\max}} + \frac{g_{p,\max} - g_p(s_h)}{g_{p,\max}} \quad (61)$$

The coordinates of s_h , $c(s_h)$, are determined per Equation (62), which is a real-time optimization solved using [53] in this work.

$$c(s_h) = \arg \max \sum_{i=0}^n \|c(s_h) - e_i\| \quad (62)$$

The radius r_h refers to the largest inscribed circle in Δp (63).

$$r_h(s_h) = \min_{e_i \in \Delta p} \|c(s_h) - e_i\| \quad (63)$$

The estimated overflown population risk from s_h to s_N on a straight path is specified by $g_p(s_h)$ per Equation (44). The best holding pattern s_h^* is identified per Equation (64).

$$s_h^* = \arg \max_{s_h \in \Delta p, \Delta p \in \mathcal{H}} U \quad (64)$$

This approach ensures an efficient and robust selection of low-risk holding points for emergency scenarios over urban areas. The holding inbound state is specified in Equation 65.

$$\begin{aligned} \chi_{h\leftarrow} &= \theta(s_0, s_h^*), \quad c(s_{h\leftarrow}) = \mathcal{W}(R, |\chi_{h\leftarrow} + 0.5\pi|_{2\pi} \mid s_h^*) \\ h_{h\leftarrow} &= h_0 - \delta h(\mathcal{P}_D(s_0, s_{h\leftarrow})) \end{aligned} \quad (65)$$

The function $\delta h : \mathbb{P} \rightarrow \mathbb{R}_{>0}$ yields the altitude loss along $\mathcal{P} \in \mathbb{P}$. Figure 8 visualizes the hold inbound state determination.

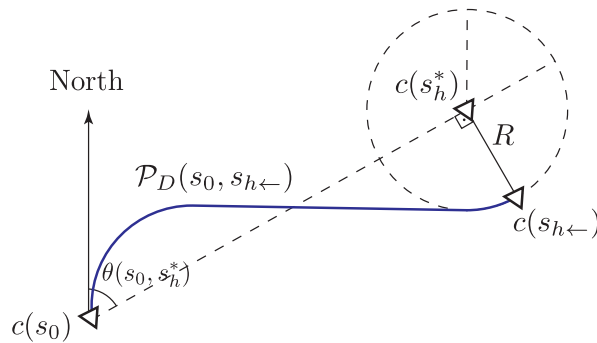


Figure 8. Hold inbound state determination.

First, the holding outbound state $s_{h\rightarrow}$ is built by determining its altitude which is set accounting for the maximum obstacle height \bar{h}_o within \mathcal{S} , optimal glide distance from s_h^* to s_N , the holding pattern floor altitude $\underline{h}_{h\rightarrow}$, and a buffer h_{buffer} . The obstacle height data can be found from aeronautical charts.

$$h_{h\rightarrow} = \max(\bar{h}_o, d_{gc}(s_h^*) \tan \gamma_{opt}, \underline{h}_{h\rightarrow}) + h_{buffer} \quad (66)$$

The number of turns n_{360} and net course change $\Delta\chi_{net}$ while holding are given in Equation (67), $\Delta h_{360} = 2\pi R \tan \gamma_{opt,t}$ being the altitude loss in a 360° turn.

$$n_{360} = \frac{h_{h\leftarrow} - h_{h\rightarrow}}{\Delta h_{360}}, \quad \Delta\chi_{net} = 2\pi(n_{360} - \lfloor n_{360} \rfloor) \quad (67)$$

The holding outbound coordinate for a left-hand hold is

$$c(s_{h\rightarrow}) = \mathcal{W}(R, |\theta(s_h^*, s_{h\leftarrow}) - \Delta\chi_{net}|_{2\pi} \mid s_h^*) \quad (68)$$

The outbound course angle, the last element of $s_{h\rightarrow}$, is found from $|\chi_{h\leftarrow} + \Delta\chi_{net}|_{2\pi}$.

5. Real-time Contingency Landing Planning Algorithms

This section presents search space discretization and contingency landing planning algorithms.

5.1. Search Space Discretization

Search space \mathcal{S} is a continuous infinite set, rendering the search problem \mathcal{T} computationally incomplete. Search over a continuous infinite set is not guaranteed to return a solution \mathcal{P}_s . To address this issue, \mathcal{S} can be discretized into \mathcal{S}_d using uniform cells, and reduced to a discrete finite set of states. Approaches for state-based discretization for search can be found in [54,55].

In this work, discretization is performed in horizontal plane, altitude, and course angle dimensions. Hexagonal prism cells are used to represent the 3D position discrete state-space $s_d \in \mathcal{S}_d$ as illustrated in Figure 9. This choice ensures efficient tiling of the space, guaranteeing full coverage of \mathcal{S} while providing an upper bound on the size of the discrete search space. However, it is important to note that each hexagonal prism does not correspond to a single state; rather, a single prism represents multiple states, each associated with a different discretized course angle. Each continuous state s within a discrete cell is mapped to a single unique discrete state s_d , which is defined as the centroid of the cell.

An upper bound to the total number of states can be estimated considering an optimal packing of hexagons into a circle, an approximate footprint. The number of layers of hexagonal cells required to cover the footprint with a radius of $r_{\mathcal{F}}$ is per Equation (69) where the circumradius of cells is $\ell/2$.

$$l = \left\lceil \frac{r_{\mathcal{F}} - \ell\sqrt{3}/4}{\ell\sqrt{3}/2} \right\rceil \quad (69)$$

The total number of latitude-longitude pairs needed to completely cover the footprint is:

$$n_{(\varphi,\lambda)} = 1 + 3l(l+1) \quad (70)$$

The number of discrete course angle and altitude ranges are

$$n_{\psi} = \left\lceil \frac{2\pi}{\Delta\psi} \right\rceil, \quad n_h = \left\lceil \frac{h_0 - h_N}{\Delta h_{360}} \right\rceil \quad (71)$$

Therefore, an upper bound to the total number of states is per Equation (72).

$$n_s = n_{(\varphi,\lambda)} \times n_{\psi} \times n_h \quad (72)$$

As a finite search space is obtained, a formal proof for guaranteed convergence can be given.

Theorem 3. *Given a finite discrete search space \mathcal{S}_d , a defined solution identification set $\mathcal{Z} \subset \mathcal{S}_d$, and assured reachability between arbitrary states $s_i \in \mathcal{S}_d$ and $s_j \in \mathcal{S}_d$, the discrete gradient-guided search from s_i to s_j is guaranteed to converge to a final state $s'_N \in \mathcal{Z}$.*

Proof of Theorem 3. The state-space \mathcal{S} is discretized into a finite set of states with $|\mathcal{S}_d| = n_s < \infty$, ensuring that the search process operates over a well-defined and bounded space. A solution from s_i to s_j exists given the reachability assurance.

$$\exists \mathcal{P}(s_i, s_j) \subset \mathcal{S}_d \quad (73)$$

Given that the search space is finite and a feasible solution exists, the algorithm is guaranteed to terminate in a discrete feasible state within \mathcal{Z} .

$$\lim_{k \rightarrow n_s} s_k = s'_N, \quad s'_N \in \mathcal{Z} \quad (74)$$

Thus, the discrete gradient-guided search is complete given the conditions. \square

Figure 9 illustrates an example of \mathcal{S}_d for underestimated best-glide ranges with $n_h = 3$ and $n_\psi = 1$. These values are intentionally chosen for visualization purposes and do not reflect the actual space discretization used in the algorithm. The red, green, and blue cells represent increasing altitudes, with $n_{(\varphi, \lambda)}$ values are 3, 4, and 5, respectively.

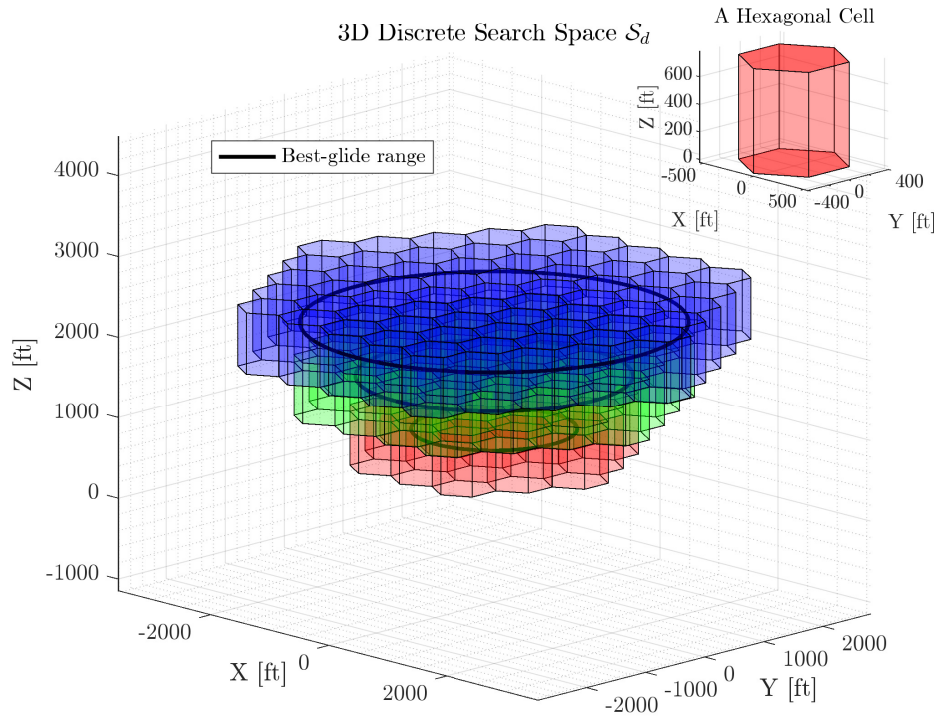


Figure 9. An illustration of a discrete search space in 3D.

Table 1 presents upper bounds on the search space size for an engine-out Cessna 182, considering various altitude differences and segment lengths. As shown, increasing altitude difference expands the search space, while the longer segment lengths reduce the number of discrete cells.

Table 1. Upper bounds of discrete search space size for the Cessna 182 experiencing loss of thrust.

$h_0 - h_N$ [ft]	n_s for Different Circumradii in ft		
	500	1000	1500
500	2.88×10^5	7.39×10^4	2.85×10^3
2500	16.75×10^6	4.50×10^6	2.01×10^6
5000	88.44×10^6	29.35×10^6	14.44×10^6

It should be noted that this structured grid is not directly used for search expansions. The search algorithm expands states based on actions defined in the previous section, and the resulting child states do not necessarily align with the hexagonal cell centers. Instead, each expanded state is assigned as the centroid of a newly generated hexagonal cell at an arbitrary position. Once a cell is explored, its corresponding centroid is stored in a closed list to track visited states and prevent redundant expansions using a point-in-polygon algorithm.

5.2. Contingency Landing Planner

This section describes a contingency landing planning framework consisting of three sequential stages. The first stage constructs an initial set of feasible landing paths using a 3D Dubins path planner formulated with Equation (9) to provide a geometric solution and holding pattern identification

governed by equations (60) - (68) to ensure safe altitude management. This first stage also performs gradient-guided search based on standard tree search [56,57] but with a key distinction: it detects repeated discrete states using a geometrical point-in-polygon algorithm rather than state hashing.

In the second stage, Algorithm 1 implements handling of different altitude cases by conditionally utilizing the first stage algorithms. Contingent on the initial emergency altitude, a direct search-based path \mathcal{P}_s is computed, followed by a Dubins connection to the landing site. If the aircraft has sufficient altitude, a holding pattern is first executed to dissipate altitude before transitioning to the final approach descent via successive search- and Dubins-based trajectories.

Algorithm 1 Altitude-dependent Path Planning

Require: $\mathbb{A}, \mathcal{A}, g(s), s_0, s_N, s_t, v_w, \chi_w$
Ensure: \mathcal{P}

```

if  $h_0 \leq h_c$  then
   $\mathcal{P}_s(s_0, s'_N) \leftarrow \mathcal{T}(\mathbb{A}, \mathcal{A}, g(s), s_0, s_N, v_w, \chi_w)$ 
   $\mathcal{P}_D(s'_N, s_t) \leftarrow \arg \min_{\mathcal{P} \in \mathcal{D}(s'_N, s_t)} \mathcal{L}(\mathcal{P})$ 
   $\mathcal{P} = \mathcal{P}_s \cup \mathcal{P}_D$ 
else
   $s_h^* \leftarrow \arg \max_{s_h \in \mathcal{H}} U$ 
   $s_{h \leftarrow} \leftarrow \text{Eq. (65)}$ 
   $\mathcal{P}_D(s_0, s_{h \leftarrow}) \leftarrow \arg \min_{\mathcal{P} \in \mathcal{D}(s_0, s_{h \leftarrow})} \mathcal{L}(\mathcal{P})$ 
  if  $h_{h \leftarrow} > h_c$  then
     $s_{h \rightarrow} \leftarrow \text{Eq. (66)-(68)}$ 
     $\mathcal{P}_D(s'_N, s_t) \leftarrow \arg \min_{\mathcal{P} \in \mathcal{D}(s'_N, s_t)} \mathcal{L}(\mathcal{P})$ 
     $\mathcal{P}_s(s_{h \rightarrow}, s'_N) \leftarrow \mathcal{T}(\mathbb{A}, \mathcal{A}, g(s), s_{h \rightarrow}, s_N, v_w, \chi_w)$ 
     $\mathcal{P}_D(s'_N, s_t) \leftarrow \arg \min_{\mathcal{P} \in \mathcal{D}(s'_N, s_t)} \mathcal{L}(\mathcal{P})$ 
     $\mathcal{P} \leftarrow \mathcal{P}_D(s_0, s_{h \leftarrow}) \cup \mathcal{P}_s(s_{h \rightarrow}, s'_N) \cup \mathcal{P}_D(s'_N, s_t)$ 
  else
     $s_c \leftarrow \arg \min_{s_c \in \mathcal{P}_D(s_0, s_{h \leftarrow})} |h - h_c|$ 
     $\mathcal{P}_s(s_c, s'_N) \leftarrow \mathcal{T}(\mathbb{A}, \mathcal{A}, g(s), s_c, s_N, v_w, \chi_w)$ 
     $\mathcal{P}_D(s'_N, s_t) \leftarrow \arg \min_{\mathcal{P} \in \mathcal{D}(s'_N, s_t)} \mathcal{L}(\mathcal{P})$ 
     $\mathcal{P} \leftarrow \mathcal{P}_D(s_0, s_c) \cup \mathcal{P}_s(s_c, s'_N) \cup \mathcal{P}_D(s'_N, s_t)$ 
  end if
end if
return  $\mathcal{P}$ 

```

The final stage executes the contingency landing planner in Algorithm 2 which runs two planners in parallel: (1) the altitude-dependent search-based path planner to define the risk-aware path, (2) a Dubins-based shortest path solver \mathcal{P}_D . Since Dubins is a geometrical solution, it enables real-time response with onboard computing resources for execution as a fallback solution in case the search-based path cannot be computed in time limit $T_{max} \in \mathbb{R}_{>0}$. As a result, this framework guarantees an emergency landing solution can be computed within a prescribed time limit. As described below, the search-based planner typically identifies a solution in time that improves safety by our cost metrics relative to Dubins.

6. Use Cases and Algorithm Benchmarking

This section presents use cases demonstrating the capabilities of the contingency landing planner under varying altitude conditions and steady wind. Descriptive statistics are also provided on the performance of the proposed method in different initial emergency states over a uniform grid. A populated residential area, Farmingdale, Long Island, NY, is chosen to simulate emergency landing scenarios. Therefore, all use cases and tests share the same goal state: a 1 NM final approach fix at 467 ft mean sea level (MSL) altitude to Runway 14 at Republic Airport (KFRG) in Long Island, New York. Table 2 lists user defined parameters for path planning.

Algorithm 2 Contingency Landing Path Planner**Require:** $\mathbb{A}, \mathcal{A}, g(s), s_0, s_N, s_t, v_w, \chi_w, T_{\max}$ **Ensure:** \mathcal{P} or fallback path P_D

```

1: Initialize runtime  $t \leftarrow 0$ 
2: Initialize path set  $\mathcal{P} \leftarrow \emptyset$ 
3: Initialize status flag  $\text{flag} \leftarrow \text{False}$ 
4: while  $t \leq T_{\max}$  and  $\neg \text{flag}$  do
5:   Parallel Execution:
6:     Run Algorithm 1 on Processor-1
7:     Solve (8) and (9) for  $P_D(s_0, s_N)$  on Processor-2
8:   if  $\mathcal{P} \neq \emptyset$  then
9:      $\text{flag} \leftarrow \text{True}$ 
10:  end if
11:   $t \leftarrow t + \Delta t$ 
12: end while
13: if  $\text{flag}$  then
14:   return  $\mathcal{P}$  ▷ Return the minimum-risk path
15: else
16:   return  $P_D(s_0, s_N)$  ▷ Fallback to default path
17: end if

```

Table 2. Parameters for altitude-dependent path planning.

Parameter	Value	Unit
R	1500	ft
h_c	5000	ft
\mathcal{X}	$\{-30, -15, 0, 15, 30\}$	$^\circ$
\underline{h}, m	2000, 0.1	ft, -
\underline{r}	0.5	NM
R_i, R_o	1500, 3000	ft, ft
$\underline{h}_{h \rightarrow}$	1000	ft
h_{buffer}	1500	ft
$\Delta\psi$	5	$^\circ$
Weight coefficients as a function of g_χ		
Condition on g_χ	Cost Weights	
$g_\chi \geq 0.5$	$(w_1, w_2, w_3, w_4) = (0.2, 0.3, 0.2, 0.3)$	
$10^{-4} < g_\chi < 0.5$	$(w_1, w_2, w_3, w_4) = (0.2, 0.3, 0.1, 0.4)$	
$g_\chi \leq 10^{-4}$	$(w_1, w_2, w_3, w_4) = (0.2, 0.4, 0.0, 0.4)$	

Weight coefficients are used to compute $g(s)$ (23) as a function of $g_\chi(s)$. A state with a high g_χ requires a final approach turn greater than 90° . To mitigate this, state expansion toward regions where the final course change remains below 90° is prioritized. As g_χ decreases, w_3 is gradually reduced, thereby shifting the cost emphasis toward overflown population and distance to the goal.

6.1. Altitude-Dependent Path Planning

The impact of initial emergency altitude on contingency landing solutions is illustrated in Figure 10, where an emergency scenario is simulated at a randomly selected location within the aircraft footprint, heading to the East, at altitudes of 2500 ft, 6000 ft, and 10000 ft MSL.

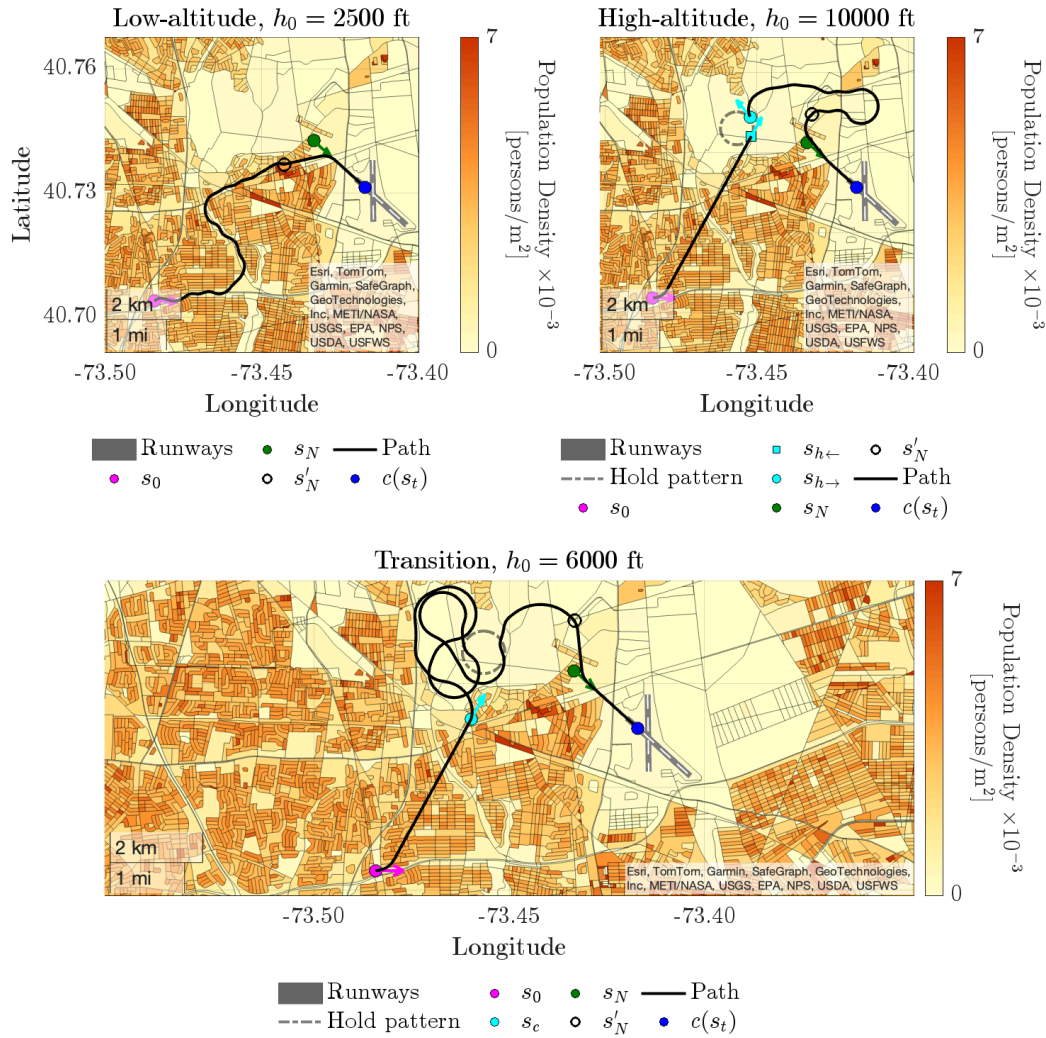


Figure 10. Contingency landing solutions for varying initial altitudes.

At a low emergency onset altitude of 2500 ft MSL, Algorithm 1 directly computes the search-based path $\mathcal{P}_s(s_0, s'_N)$ and a 3D Dubins path for the final approach to Runway 14 at KFRG, as the initial altitude h_0 is below the crossover altitude h_c . The trajectory effectively avoids residential areas by following roads, leveraging available altitude for risk mitigation. This path is computed with as few as 80 explored states, dramatically below the upper bound of the search space size of 1.2×10^6 for this case.

At a high emergency altitude of 10000 ft MSL, altitude dissipation is necessary before approach. A holding pattern is established over a sparsely populated region while maintaining a 0.5 NM no-fly zone around Runway 14 to prevent conflicts with the airport traffic pattern. A 3D Dubins path is then generated from s_0 to $s_{h\leftarrow}$ without considering population risk, as exposure risk at high altitude is minimal due to $h_0 > h_c$. The distressed aircraft descends from $h_{h\leftarrow} = 8637$ ft to $h_{h\rightarrow} = 2500$ ft over 8.2 full left-hand holding turns. Despite this descent, excess altitude remains as the aircraft approaches s_N . The gradient-guided search subsequently determines a descent path to s'_N that avoids densely populated areas, exploring a total of 72 states. Once the search-based path is generated, a 3D Dubins path is planned from s'_N to s_t .

At an initial emergency altitude of 6000 ft, a holding pattern is also identified. However, unlike the 10000 ft case, the landing trajectory deviates from the initial Dubins path upon reaching the crossover altitude h_c at s_c , as the hold inbound cannot be reached above this altitude. From the crossover state s_c , gradient-guided search is initialized. Similar to the example in Figure 4, $g_{d,1}(s)$ leads the minimum cost states away from s_N due to the aircraft's higher altitude relative to the remaining horizontal

traversal, causing state expansion away from s_N . This outward expansion is further constrained by the densely populated area in the Northwest of the map, guiding the landing trajectory along the boundary separating low- and high-density regions. As altitude decreases, $g_{d,1}(s)$ shifts the minimum-cost states closer to s_N , gradually directing state expansion toward s_N . Compared to the low- and high-altitude cases, the landing path found by the gradient-guided search is significantly longer due to the higher crossover altitude; however, $\mathcal{P}_s(s_c, s'_N)$ is computed with a total of 110 explored states whereas $n_s = 12.3 \times 10^6$. As with the previous cases, the final segment is completed with a 3D Dubins path to s_t . The complete 3D contingency landing solution, along with the explored states for the transition case, is shown in Figure 11. The low number of state expansions demonstrates the efficiency of the presented search-based landing planner.

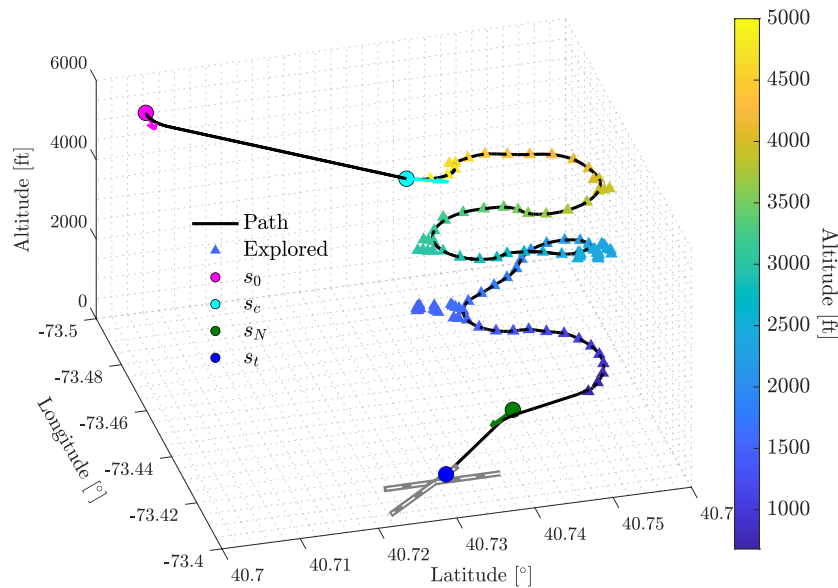


Figure 11. The 3D contingency landing trajectory and explored state coordinates for the transition case, initially at 6000 ft MSL.

6.2. Contingency Landing Planning Under Steady Wind

Recalling equations (5) and (14), a feasible action depends on wind conditions. A nonzero wind vector alters both the direction and magnitude of the ground speed vector, which in turn affects the required flight path angle to maintain a given airspeed. For path planning, the flight path angle for each segment is selected from a dataset derived from equations (1)-(4), taking into account wind speed and relative direction.

To illustrate the effect of wind on contingency landing planning, two trajectories are compared in Figure 12.

In this emergency scenario, the aircraft begins at 2500 ft MSL with a 10 knot South wind ($v_w = 10$ kts, $\chi_w = 180^\circ$). In windless conditions, the planned trajectory spans 5.41 NM at a constant optimal flight path angle of 4.4° . However, with a partial tailwind, the feasible flight path angle range becomes shallower with an average optimal flight path angle of 4.18° , leading to an extended trajectory of 5.53 NM. Steady wind is particularly challenging under headwind conditions which requires a steeper descent. This landing site would not be reachable in case of 10 knots North wind because the feasible flight path angle range would too steep, resulting in greater altitude loss per unit distance traveled.

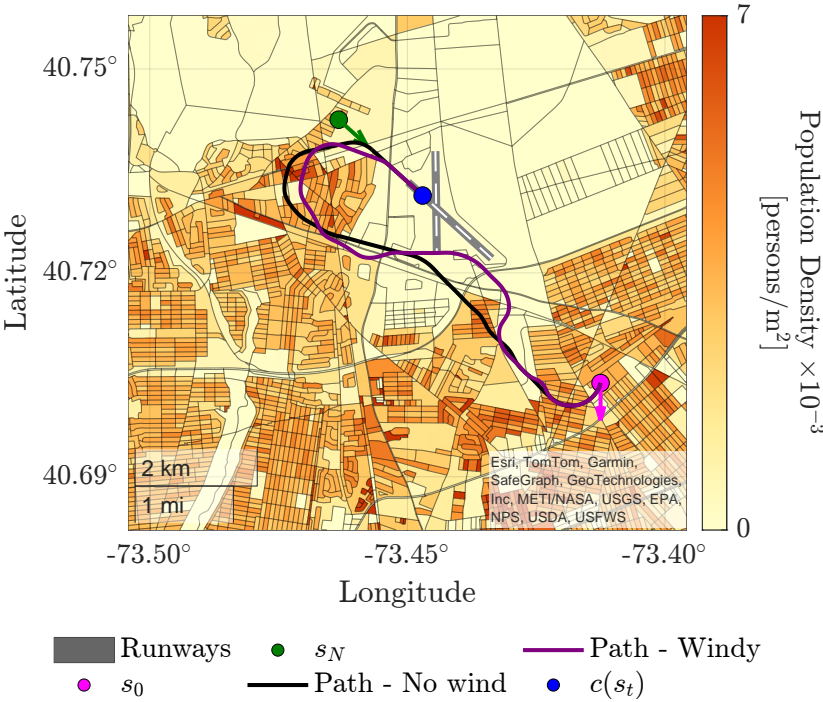


Figure 12. The impact of tailwind in the contingency landing solution.

6.3. Algorithm Benchmarking on a Uniform Grid

A benchmark study is conducted to evaluate the developed algorithm’s performance in terms of overflowed population risk, optimal flight path angle, and computational overhead. Flight time comparison is refrained as it conflicts with the optimal descent. For contingency landing solutions, the implemented framework is compared against 3D and S-turn Dubins paths. To determine the best Dubins-based solution for low-altitude cases, all possible Dubins path types are evaluated and the one that minimizes overflowed population risk is selected. Specifically, for lower altitudes, four types of 3D Dubins paths are generated. If additional altitude dissipation is required, S-turn Dubins paths, extending in both directions, are computed and the one with minimum risk is chosen for comparison. Test scenarios are distributed across a uniform grid with nine coordinates, four altitude levels, and four cardinal directions, totaling 144 unique cases. All initial states are ensured to be reachable from the goal. Algorithm 1 is compared against the minimum-risk Dubins solutions, with results summarized in Table 3, including descriptive statistics on overflowed population risk and flight path angle deviation from the optimal.

Table 3. Risk and gliding constraint comparison of the gradient-guided search and the minimum-risk 3D Dubins solutions.

h_0 [ft]	Method	Population Risks					γ Deviation [°]		
		Min.	Max.	Median	μ	σ	Max.	μ	σ
2500	Search	0.0006	0.2157	0.0221	0.0418	0.0451	0.30	-0.06	0.09
	Dubins	0.0028	0.2048	0.0620	0.0667	0.0575	0.27	-0.02	0.06
5000	Search	0.0021	0.1183	0.0085	0.0206	0.0260	0.23	-0.07	0.04
	Dubins	0.0132	0.1987	0.0704	0.0843	0.0574	0.00	0.00	0.00
6000	Search	0.0010	0.0136	0.0038	0.0046	0.0035	0.29	-0.09	0.08
	Dubins	0.0022	0.1317	0.0267	0.0358	0.0373	0.00	0.00	0.00
10000	Search	0.0014	0.0101	0.0045	0.0047	0.0023	0.27	-0.05	0.07
	Dubins	0.0021	0.0388	0.0068	0.0106	0.0090	0.00	0.00	0.00

The search-based method consistently outperforms the Dubins-based approach in minimizing overflowed population risk across altitude levels. This is expected as the construction of Dubins paths does not offer explicit risk mitigation.

Meanwhile, the Dubins-based paths adhere strictly to their predefined optimal flight path angles, with deviations close to zero in mid- and high altitude cases. This is because S-turn Dubins paths can precisely achieve the desired altitude loss in case altitude dissipation is needed. In contrast, the search-based method exhibits slight deviations. However, it ensures that these deviations remain within flight envelope and can be further reduced by increasing the discrete search resolution with shorter segment lengths or tighter constraints.

The framework, along with the benchmarking Dubins path solvers, is programmed in C, with the exception of R*-Tree whose source codes are written in C++ [58]. Table 4 provides a detailed runtime analysis benchmarking search-based path planning against the minimum-risk Dubins solutions. For higher altitude scenarios, the total elapsed time for optimal holding point identification and the subsequent Dubins path solution to the hold pattern, labeled as hold planning, are also tabulated. Despite being slower than Dubins solutions as would be anticipated, the developed contingency landing planner exhibits strong potential for real-time implementation. In addition, the separate computation times $g_p(s)$ (44) are provided to highlight its significant contribution to the total computational overhead. As is evident, a high percentage of the total runtime is attributed to the overflown population risk estimation despite an efficient R*-Tree implementation. This overhead could be reduced, albeit with a trade-off in accuracy due to interpolation, by reducing the integration time step or employing data projection methods onto uniform grids.

Table 4. Runtime comparison of the gradient-guided search and the minimum-risk 3D Dubins solutions in C-language.

h_0 [ft]	Method	Total Runtime [ms]					Risk (g_p) Computation Runtime [ms]				
		Min.	Max.	Median	μ	σ	Min.	Max.	Median	μ	σ
2500	Search	27.0	764.5	92.2	135.1	128.8	26.0	671.6	89.6	129.3	115.2
	Dubins	8.2	121.4	67.9	63.5	28.6	8.1	68.3	34.3	35.7	18.1
5000	Search	133.0	6897.0	441.4	723.1	1140.3	130.3	4963.5	432.4	646.3	820.0
	Dubins	99.7	221.2	167.7	162.0	31.6	89.2	210.9	158.9	153.1	31.1
6000	Hold Planning	963.9	1120.8	987.7	1003.4	41.0	24.0	66.4	49.2	48.0	10.4
	Search	76.5	1947.3	180.5	314.5	358.7	74.8	1893.8	175.1	305.9	349.1
	Dubins	59.9	226.2	94.8	114.0	47.6	42.9	218.8	82.0	100.6	51.8
10000	Hold Planning	976.5	1127.2	1033.8	1034.9	31.9	24.5	69.1	50.2	49.2	11.2
	Search	37.8	510.8	175.3	197.1	116.1	36.8	495.5	168.7	190.2	112.3
	Dubins	46.0	105.9	73.0	75.6	16.0	15.2	81.8	54.7	53.7	15.4

All experiments are conducted using a single core of Apple M2 chip (3.49 GHz).

Table 5 presents the solution availability of Algorithm 2 across 144 test cases for different solver time limits. For initial emergency altitudes of 2500 ft and 5000 ft MSL, the planner provides search-based minimum-risk landing solutions in 67 out of 72 scenarios within the solver time limit of $T_{max} = 1$ second. At higher altitudes, Algorithm 2 predominantly outputs shortest Dubins solutions due to the hold planning cost, as shown in Table 4. When the solver time limit is increased to $T_{max} = 2$ seconds, search-based landing solutions are found in 141 out of 144 cases. Only one scenario requires a fallback Dubins solution when $T_{max} = 3$ seconds is used.

Table 5. Algorithm 2 solution availability for different time limits.

T_{max} [s]	h_0 [ft]				Number of solutions	
	2500	5000	6000	10000	Search-based (\mathcal{P})	Fallback Dubins (\mathcal{P}_D)
1	36	31	0	1	68	76
2	36	35	34	36	141	3
3	36	35	36	36	143	1

7. Discussion

This paper presented a contingency landing planning framework for an engine-out Cessna 182 model, serving as a surrogate for a distressed AAM aircraft in cruise configuration. However, the proposed approach is adaptable to various emergency scenarios, including but not limited to control

surface failures and structural damage by defining the set of feasible actions based on the aircraft's degraded performance envelope.

To estimate overflown population risk, publicly available population census data is utilized. To improve risk quantification, an interquartile rule for outlier detection was applied. These values were then adjusted to the third quartile population density, ensuring a more representative risk assessment. Alternative population models might be created and updated with real-time data, e.g., from cellular network call data.

In general, the number of explored states in the presented use cases remains low. However, it is observed that when the initial emergency state is located in a densely populated area with a high initial course angle cost, state expansion increases as the algorithm searches for a low-cost trajectory, leading to longer solution times. Additionally, a greater altitude difference between the initial and goal states requires a longer landing path, further increasing state expansion. The gradient-guided search utilizes course angle changes to navigate through narrow low-risk passages. This is not a concern for aircraft with attitude control authority and modern autopilot systems. However, these solutions can only reasonably be shared with air traffic control and other nearby air traffic via datalink to assure coordinated response to separation and collision avoidance.

8. Conclusion

This paper presented a real-time emergency landing planner that combines a novel gradient-guided discrete search with 3D Dubins paths to generate safe and risk-aware landing trajectories for fixed-wing aircraft. To further minimize high-altitude risk, a minimum-risk holding pattern identification algorithm was also introduced. A series of 144 Long Island, New York case studies demonstrated that the presented method significantly reduces risk compared to standard minimum-risk Dubins path solvers. While computationally more demanding, the search-based planner achieved real-time performance, generating minimum-risk landing trajectories within three seconds in 143 out of 144 cases, with a fallback Dubins path returned only once. Future work must consider other air traffic in contingency flight planning solutions to minimize risk associated with disrupting or losing separation with other air traffic. Such solutions will require awareness of commercial and Advanced Air Mobility airspace corridors and reserved geofence volumes for small uncrewed aircraft system traffic. To address this, airspace and air traffic avoidance risks will be captured in contingency landing planning by introducing new cost terms within the gradient-guided search framework.

Author Contributions: Conceptualization, H.E.T. and E.A.; methodology, H.E.T. and E.A.; software, H.E.T.; validation, E.A.; formal analysis, H.E.T. and E.A.; investigation, H.E.T. and E.A.; resources, E.A.; data curation, H.E.T. and E.A.; writing—original draft preparation, H.E.T. and E.A.; writing—review and editing, E.A.; visualization, H.E.T. and E.A.; supervision, E.A.; project administration, E.A.; funding acquisition, E.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Langley Research Center grant number 80LARC23DA003.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Bacchini, A.; Cestino, E.; Van Magill, B.; Verstraete, D. Impact of lift propeller drag on the performance of eVTOL lift+ cruise aircraft. *Aerospace Science and Technology* **2021**, *109*, 106429.
2. Mathur, A.; Atkins, E. Multi-Mode Flight Simulation and Energy-Aware Coverage Path Planning for a Lift+Cruise QuadPlane. *Drones* **2025**, *9*. <https://doi.org/10.3390/drones9040287>.

3. Castagno, J.; Atkins, E. Roof Shape Classification from LiDAR and Satellite Image Data Fusion Using Supervised Learning. *Sensors* **2018**, *18*. <https://doi.org/10.3390/s18113960>.
4. Kim, J.; Atkins, E. Airspace Geofencing and Flight Planning for Low-Altitude, Urban, Small Unmanned Aircraft Systems. *Applied Sciences* **2022**, *12*. <https://doi.org/10.3390/app12020576>.
5. Russell, L.; Goubran, R.; Kwamena, F. Emerging Urban Challenge: RPAS/UAVs in Cities. In Proceedings of the 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), 2019, pp. 546–553. <https://doi.org/10.1109/DCOSS.2019.00103>.
6. Dubins, L.E. On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *American Journal of Mathematics* **1957**, *79*, 497–516.
7. Yomchinda, T.; Horn, J.F.; Langelaan, J.W. Modified Dubins parameterization for aircraft emergency trajectory planning. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* **2017**, *231*, 374–393.
8. Chitsaz, H.; LaValle, S.M. Time-optimal paths for a Dubins airplane. In Proceedings of the IEEE CDC, 2007, pp. 2379–2384. <https://doi.org/10.1109/CDC.2007.4434966>.
9. Ma, D.; Hao, S.; Ma, W.; Zheng, H.; Xu, X. An optimal control-based path planning method for unmanned surface vehicles in complex environments. *Ocean Engineering* **2022**, *245*, 110532. <https://doi.org/10.1016/j.oceaneng.2022.110532>.
10. Bergman, K.; Ljungqvist, O.; Axehill, D. Improved Path Planning by Tightly Combining Lattice-Based Path Planning and Optimal Control. *IEEE Transactions on Intelligent Vehicles* **2021**, *6*, 57–66. <https://doi.org/10.1109/TIV.2020.2991951>.
11. Liu, J.; Han, W.; Liu, C.; Peng, H. A New Method for the Optimal Control Problem of Path Planning for Unmanned Ground Systems. *IEEE Access* **2018**, *6*, 33251–33260. <https://doi.org/10.1109/ACCESS.2018.2846769>.
12. Zhang, K.; Sprinkle, J.; Sanfelice, R.G. A hybrid model predictive controller for path planning and path following. In Proceedings of the Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems, New York, NY, USA, 2015; p. 139–148. <https://doi.org/10.1145/2735960.2735966>.
13. Shen, C.; Shi, Y.; Buckham, B. Model predictive control for an AUV with dynamic path planning. In Proceedings of the 2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), 2015, pp. 475–480. <https://doi.org/10.1109/SICE.2015.7285374>.
14. Akametalu, A.K.; Tomlin, C.J.; Chen, M. Reachability-Based Forced Landing System. *Journal of Guidance, Control, and Dynamics* **2018**, *41*, 2529–2542. <https://doi.org/10.2514/1.g003490>.
15. Bertsekas, D.P. *Dynamic Programming and Optimal Control, Volume 1, Third Edition*; Athena Scientific, Belmont, 2005.
16. Di Donato, P.F.A.; Atkins, E.M. Evaluating Risk to People and Property for Aircraft Emergency Landing Planning. *AIAA Journal of Aerospace Information Systems* **2017**, *14*, 259–278. <https://doi.org/10.2514/1.J010513>.
17. Tekaslan, H.E.; Atkins, E.M. Vehicle-to-Vehicle Approach to Assured Aircraft Emergency Road Landings. *Journal of Guidance, Control, and Dynamics* **2025**, *48*, 1800–1817. <https://doi.org/10.2514/1.G008803>.
18. Howlett, J.K.; McLain, T.W.; Goodrich, M.A. Learning Real-Time A* Path Planner for Unmanned Air Vehicle Target Sensing. *Journal of Aerospace Computing, Information, and Communication* **2006**, *3*, 108–122. <https://doi.org/10.2514/1.16623>.
19. Qian, Y.; Sheng, K.; Ma, C.; Li, J.; Ding, M.; Hassan, M. Path Planning for the Dynamic UAV-Aided Wireless Systems Using Monte Carlo Tree Search. *IEEE Transactions on Vehicular Technology* **2022**, *71*, 6716–6721. <https://doi.org/10.1109/TVT.2022.3160746>.
20. Chour, K.; Pradeep, P.; Munishkin, A.A.; Kalyanam, K.M. Aerial Vehicle Routing and Scheduling for UAS Traffic Management: A Hybrid Monte Carlo Tree Search Approach. In Proceedings of the 2023 IEEE/AIAA Conference on Digital Avionics Systems, 2023, pp. 1–9. <https://doi.org/10.1109/DASC58513.2023.10311314>.
21. Guo, Y.; Liu, X.; Jia, Q.; Liu, X.; Zhang, W. HPO-RRT*: a sampling-based algorithm for UAV real-time path planning in a dynamic environment. *Complex & Intelligent Systems* **2023**, *9*, 7133–7153. <https://doi.org/10.1007/s40747-023-01115-2>.
22. Xu, D.; Qian, H.; Zhang, S. An Improved RRT*-Based Real-Time Path Planning Algorithm for UAV. In Proceedings of the IEEE International Conference on High Performance Computing & Communications, 2021, pp. 883–888. <https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00141>.
23. Kothari, M.; Postlethwaite, I.; Gu, D.W. Multi-UAV path planning in obstacle rich environments using Rapidly-exploring Random Trees. In Proceedings of the IEEE Conference on Decision and Control, 2009, pp. 3069–3074. <https://doi.org/10.1109/CDC.2009.5400108>.

24. Sláma, J.; Herynek, J.; Faigl, J. Risk-Aware Emergency Landing Planning for Gliding Aircraft Model in Urban Environments. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2023, pp. 4820–4826. <https://doi.org/10.1109/IROS55552.2023.10341622>.
25. Meuleau, N.; Plaunt, C.; Smith, D.; Smith, T. A POMDP for Optimal Motion Planning with Uncertain Dynamics. In Proceedings of the ICAPS-10: POMDP Practitioners Workshop. Citeseer, 2010.
26. Sharma, P.; Kraske, B.; Kim, J.; Laouar, Z.; Sunberg, Z.; Atkins, E. Risk-Aware Markov Decision Process Contingency Management Autonomy for Uncrewed Aircraft Systems. *AIAA Journal of Aerospace Information Systems* **2024**, *21*, 234–248.
27. Strube, M.; Sanner, R.; Atkins, E. Dynamic Flight Guidance Recalibration After Actuator Failure. In Proceedings of the AIAA 1st Intelligent Systems Technical Conference, 2004, p. 6255.
28. Atkins, E.M.; Portillo, I.A.; Strube, M.J. Emergency Flight Planning Applied to Total Loss of Thrust. *AIAA Journal of Aircraft* **2006**, *43*, 1205–1216.
29. Castagno, J.; Atkins, E. Map-based planning for small unmanned aircraft rooftop landing. In *Handbook of Reinforcement Learning and Control*; Springer, 2021; pp. 613–646.
30. Tekaslan, H.E.; Atkins, E.M. Gradient Guided Search for Aircraft Contingency Landing Planning. In Proceedings of the IEEE International Conference on Robotics and Automation, 2025.
31. Daniel, K.; Nash, A.; Koenig, S.; Felner, A. Theta*: Any-Angle Path Planning on Grids. *Journal of Artificial Intelligence Research* **2010**, *39*, 533–579.
32. Beard, R.W.; McLain, T.W. *Small Unmanned Aircraft: Theory and Practice*; Princeton University Press, 2012.
33. Raymer, D. *Aircraft Design: A Conceptual Approach, Fifth Edition*; American Institute of Aeronautics and Astronautics, Inc., 2012.
34. Donato, P. Toward Autonomous Aircraft Emergency Landing Planning. PhD thesis, University of Michigan, Ann Arbor, 2017.
35. Napolitano, M.R. *Aircraft Dynamics: From Modeling to Simulation*; John Wiley, 2011.
36. Coombes, M.; Chen, W.H.; Render, P. Reachability Analysis of Landing Sites for Forced Landing of a UAS. *Journal of Intelligent & Robotic Systems* **2013**, *73*, 635–653. <https://doi.org/10.1007/s10846-013-9920-9>.
37. Matthew Coombes, W.H.C.; Render, P. Landing Site Reachability in a Forced Landing of Unmanned Aircraft in Wind. *AIAA Journal of Aircraft* **2017**, *54*, 1415–1427. <https://doi.org/10.2514/1.C033856>.
38. Arslantaş, Y.E.; Oehlschlägel, T.; Sagliano, M. Safe landing area determination for a Moon lander by reachability analysis. *Acta Astronautica* **2016**, *128*, 607–615. <https://doi.org/https://doi.org/10.1016/j.actaastro.2016.08.013>.
39. Kirchner, M.R.; Ball, E.; Hoffler, J.; Gaublumme, D. Reachability as a Unifying Framework for Computing Helicopter Safe Operating Conditions and Autonomous Emergency Landing. *IFAC-PapersOnLine* **2020**, *53*, 9282–9287. <https://doi.org/https://doi.org/10.1016/j.ifacol.2020.12.2381>.
40. Chen, M.; Tomlin, C.J. Hamilton–Jacobi Reachability: Some Recent Theoretical Advances and Applications in Unmanned Airspace Management. *Annual Review of Control, Robotics, and Autonomous Systems* **2018**, *1*, 333–358. <https://doi.org/https://doi.org/10.1146/annurev-control-060117-104941>.
41. Di Donato, P.F.A.; Atkins, E.M. Optimizing Steady Turns for Gliding Trajectories. *AIAA Journal of Guidance, Control, and Dynamics* **2016**, *39*, 2627–2637. <https://doi.org/10.2514/1.G000319>.
42. Shanmugavel, M.; Tsourdos, A.; White, B.; Zbikowski, R. 3D Dubins Sets Based Coordinated Path Planning for Swarm of UAVs. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, 2006.
43. Shanmugavel, M.; Tsourdos, A.; White, B.; Zbikowski, R. Cooperative path planning of multiple UAVs using Dubins paths with clothoid arcs. *Control Engineering Practice* **2010**, *18*, 1084–1092.
44. Fallast, A.; Messnarz, B. Automated trajectory generation and airport selection for an emergency landing procedure of a CS23 aircraft. *CEAS Aeronautical Journal* **2017**, *8*, 481–492. <https://doi.org/10.1007/s13272-017-0252-5>.
45. Di Donato, P.F.A.; Balachandran, S.; McDonough, K.; Atkins, E.; Kolmanovsky, I. Envelope-Aware Flight Management for Loss of Control Prevention Given Rudder Jam. *AIAA Journal of Guidance, Control, and Dynamics* **2017**, *40*, 1027–1041. <https://doi.org/10.2514/1.G000252>.
46. Bacon, B.; Gregory, I. General Equations of Motion for a Damaged Asymmetric Aircraft. In Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit, 2012. <https://doi.org/doi:10.2514/6.2007-6306>.
47. Lampton, A.; Valasek, J. Prediction of Icing Effects on the Lateral/Directional Stability and Control of Light Airplanes. In Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit, 2012. <https://doi.org/10.2514/6.2006-6834>.

48. Imagery, N.; (NIMA), M.A. Department of Defense World Geodetic System 1984, Its Definition and Relationships With Local Geodetic Systems. Technical Report TR8350.2, National Imagery and Mapping Agency, Bethesda, MD, 2000.
49. U.S. Census Bureau. RACE. U.S. Census Bureau. Accessed on 6 November 2023.
50. Rana, I.K. *An Introduction to Measure and Integration, Second Edition*; American Mathematical Society, 2002.
51. Chae, S.B. *Lebesgue Integration, Second Edition*; Springer, 1995.
52. Beckmann, N.; Kriegel, H.P.; Schneider, R.; Seeger, B. The R*-tree: an efficient and robust access method for points and rectangles. *SIGMOD Rec.* **1990**, *19*. <https://doi.org/10.1145/93605.98741>.
53. Johnson, S.G. The NLOpt nonlinear-optimization package. <https://github.com/stevengj/nlopt>, 2007.
54. Bandi, S.; Thalmann, D. Space discretization for efficient human navigation. *Computer Graphics Forum* **1998**, *17*, 195–206.
55. Henrich, D.; Wurll, C.; Worn, H. Online path planning with optimal C-space discretization. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, 1998, Vol. 3, pp. 1479–1484 vol.3. <https://doi.org/10.1109/IROS.1998.724803>.
56. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 3 ed.; Prentice Hall Series, Pearson, 2010.
57. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* **1968**, *4*, 100–107. <https://doi.org/10.1109/TSSC.1968.300136>.
58. Libspatialindex Contributors. libspatialindex: A General Framework for Spatial Indexing, 2025. "Accessed: Feb 2025".

Short Biography of Authors



H. Emre Tekaslan is a Ph.D. candidate at the Kevin T. Crofton Aerospace and Ocean Engineering Department at Virginia Tech. He holds a B.S. degree in Aeronautical Engineering and an M.S. degree in Aeronautical and Astronautical Engineering from Istanbul Technical University (2019, 2022). During his M.S. studies, he pursued research in multidisciplinary low-boom supersonic aircraft design optimization and scientific machine learning. In addition to his research activities, he volunteered in a NATO Science and Technology Organization Applied Vehicle Technology Panel. H. Emre's current research area focuses on aircraft contingency landing management. In recognition of his work, he has been awarded the Boeing Scholarship and the Pratt Fellowship. He is a reviewer for the AIAA Journal of Aerospace Information Systems and AIAA Journal of Aircraft. He is also a student glider pilot.



Ella M. Atkins is Fred D. Durham Professor and Head of the Kevin T. Crofton Aerospace and Ocean Engineering Department at Virginia Tech. She was previously a Professor in the University of Michigan's Aerospace Engineering and Robotics Departments. Dr. Atkins holds B.S. and M.S. degrees in Aeronautics and Astronautics from MIT (1988, 1990) and M.S. and Ph.D. degrees in Computer Science and Engineering from the University of Michigan (1995, 1999). She is an AIAA Fellow and private pilot. She has authored over 250 refereed journal and conference papers. Dr. Atkins pursues research in AI-enabled autonomy and control to support resilience and contingency management in crewed and uncrewed Aerospace applications with focus on Advanced Air Mobility and Uncrewed Aircraft Systems (UAS). She is Editor-in-Chief of the AIAA Journal of Aerospace Information Systems (JAIS).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.