



Article A Hybrid Global/Reactive Algorithm for Collision-Free UAV Navigation in 3D Environments with Steady and Moving Obstacles

Satish C. Verma, Siyuan Li and Andrey V. Savkin * 💿

School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney 2052, Australia; satishchandra.verma@student.unsw.edu.au (S.C.V); siyuan.li4@student.unsw.edu.au (S.L.)
* Correspondence: a.savkin@unsw.edu.au

Abstract: This paper introduces a practical navigation approach for nonholonomic Unmanned Aerial Vehicles (UAVs) in 3D environment settings with numerous stationary and dynamic obstacles. To achieve the intended outcome, Dynamic Programming (DP) is combined with a reactive control algorithm. The DP allows the UAVs to navigate among known static barriers and obstacles. Additionally, the reactive controller uses data from the onboard sensor to avoid unforeseen obstacles. The proposed strategy is illustrated through computer simulation results. In simulations, the UAV successfully navigates around dynamic obstacles while maintaining its route to the target. These results highlight the ability of our proposed approach to ensure safe and efficient UAV navigation in complex and obstacle-laden environments.

Keywords: UAV path planning; navigation of aerial drones; collision-free navigation of aerial drones; obstacle avoidance; steady and moving obstacles; hybrid global/reactive navigation algorithm; unmanned aerial vehicle (UAV); dynamic programming; sliding-mode control



Citation: Verma, S.C.; Li, S.; Savkin, A.V. A Hybrid Global/Reactive Algorithm for Collision-Free UAV Navigation in 3D Environments with Steady and Moving Obstacles. *Drones* **2023**, 7, 675. https://doi.org/10.3390/ drones7110675

Academic Editor: Oleg Yakimenko

Received: 30 September 2023 Revised: 7 November 2023 Accepted: 7 November 2023 Published: 13 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).

1. Introduction

In recent decades, Unmanned Aerial Vehicles (UAVs) have been successfully applied in various sectors to perform a variety of tasks, such as inspection and maintenance of civil structures [1], search and rescue operations during disasters [2], surveillance [3], agricultural tasks [4], environmental monitoring [5], wildlife surveys [6], space exploration [7], and many others [8,9]. One of the main challenges for autonomous operation in these environments is ensuring the safe navigation of UAVs in complex environments without colliding with static and dynamic obstacles.

Designing a collision-free path within a dynamic environment poses a formidable challenge, requiring substantial computational resources. The onboard processing capacity of UAVs is limited by multiple constraints such as battery power, weight, and space. One popular approach is to offload complex tasks with no hard real-time deadlines from an autonomous vehicle's onboard computers to a cloud server [10,11]. Recent advances in wireless communication technology and cloud computing make it possible to offload non-critical complex tasks [12]. The primary benefit of offloading computations is the enhancement of overall efficiency, as it reduces the workload on the onboard computer. However, reliance on cloud computing in dynamic environments can present difficulties such as latency and real-time constraints. In response to these challenges, we endeavour to tackle them by proposing a viable strategy tailored for Unmanned Aerial Vehicles (UAVs) operating in dynamic space.

There are various approaches in the literature to path planning for UAVs [13]. Recent works are classified into two groups: global path planning and reactive (local) path planning. Global path planning typically utilizes a complete map of the 3D space to provide an optimal path, making it more suitable for known environments with static barriers. However, when deployed in dynamic environments, global path planning requires the additional burden of replanning. Major prominent contributions include rapidly exploring random tree (RRT) and its variants [14], the A* algorithm and its variants [15], machine learning techniques [16,17], meta-heuristic strategies [18], and various Dynamic Programming-based algorithms [19]. Each of these approaches exhibits distinct strengths and weaknesses. To illustrate, within the A* algorithm family, the selection of appropriate heuristics plays an important role in the outcomes. These heuristic approaches excel in providing satisfactory solutions for UAV path planning with single-objective objectives, and have proven suitable for static environments characterized by diverse constraints. However, they do not consistently reach optimal solutions and tend to generate suboptimal results when tasked with UAV path planning involving multiple objectives, frequently falling into local optima [15]. Within the RRT algorithm family, the optimality of the resulting trajectory is dependent upon the scale of the randomly sampled points. They yield favourable outcomes, offering efficient path optimization, rapid solution generation, and time efficiency when applied to static environments featuring straightforward obstacles. Nevertheless, they are susceptible to shortcomings with respect to local minima, and provide no assurance of optimality in their results due to the imposition of constraints [14]. Machine learning techniques can produce optimal results encompassing path length, smoothness, time efficiency, and collision avoidance. These intelligent approaches are particularly wellsuited for dynamic environments prone to sudden changes. Nevertheless, they come with a drawback in that they demand substantial environmental datasets in order to furnish optimal solutions, particularly in the context of supervised learning. Furthermore, they entail extensive training sessions before achieving an accurate final model [16]. Meta-heuristic strategies deliver commendable outcomes for UAV path planning involving multi-objective objectives. They exhibit reasonable execution times and offer intelligent solutions that are relatively straightforward to implement. Nonetheless, they are less suitable for real-time planning, as they require meticulous control parameter tuning and substantial resource allocations. Additionally, they lack a robust theoretical converging property [18]. Dynamic Programming can produce optimal results, although it can be computationally expensive. Cloud computing can provide the excess computational power required to achieve optimal outcomes. However, Dynamic Programming can be challenging for real-time applications due to the latency involved in wireless communications. Consequently, using local path planning with onboard sensors can be a potential solution to this problem [19].

Reactive or local path planning techniques for Unmanned Aerial Vehicles (UAVs) typically operate without a comprehensive environmental map for navigation. One prevalent category of local path planning involves 'sense and avoid algorithms', sometimes referred to as 'reactive algorithms'. These algorithms use the data acquired from onboard sensors to evade encountered obstacles. In terms of computational complexity, they are generally more cost-efficient than the alternatives. Consequently, reactive strategies facilitate fast real-time responsiveness for UAVs in dynamic environments.

Conventional reactive algorithms for two-dimensional (2D) environments encompass dynamic windows, velocity obstacles, collision cones, and other similar approaches [13]. Another set of methods for obstacle avoidance stems from biologically-inspired concepts [13]. In a prior research work [20], a range of Sliding Mode-based Controllers (SMCs) were deployed. SMCs are favoured for their ease of implementation and reduced computational resource requirements. Furthermore, their inherent robustness in the face of disturbances and model uncertainties makes them a good choice [21,22]. Consequently, reactive algorithms are preferred for local path planning, enabling UAVs to avoid obstacles in dynamic environmental settings.

In this research paper, we introduce an innovative hybrid global/reactive algorithm designed for Unmanned Aerial Vehicle (UAV) path planning within unfamiliar and dynamic settings. Combining global path planning and reactive control approaches in hybrid strategies can overcome the aforementioned drawbacks and lead to more efficient navigation. We combine a dynamic programming-based global path planning algorithm with a Sliding Mode Control (SMC)-based reactive (local) algorithm. The dynamic programming-based global path planner ensures that the UAV navigation algorithm operates in a time-optimal manner, whereas the SMC-based reactive path planning makes the proposed strategy able to operate in unknown dynamic 3D spaces. Moreover, using SMC-based navigation grants the proposed path planning algorithm robustness against uncertainties in the environment, including both unknown motion of moving obstacles and unmodelled dynamics of the UAV. The trajectory may become sub-optimal in real-time obstacle scenarios, a trade-off that is vital for obstacle avoidance. UAV efficiency is improved by offloading complex computations to remote systems. It reduces onboard computational demands, which benefits UAVs with limited resources such as battery power, memory space, and computational power. Additionally, the utilization of Sliding Mode Control (SMC) ensures swift UAV response. In summary, our approach balances optimal path planning with computational ease and agility to collectively enhance UAV efficiency. Unlike many other papers in the field, we consider non-planar 3D environments as well as both steady and dynamic (moving) obstacles.

This paper is divided into five major parts. Section 2 is about related work, discussing a significant body of literature on various hybrid approaches used by different researchers for path planning. Section 3 describes the proposed system and presents the problem statement. Section 4 provides a concise overview of the solution and introduces the technique that addresses the specified problem. Section 5 presents the results of computer simulations. Finally, Section 6 provides concluding remarks.

2. Related Work

In the general literature, hybrid approaches encompass combinations of different pre-existing path planning strategies. In this section, we discuss recent developments in hybrid algorithms. Quite often, two-dimensional work is augmented to include threedimensional space. Several notable contributions have been made in two-dimensional path planning using hybrid algorithms. Xu et al. combined Artificial Potential Field with a Genetic Algorithm for path planning and collision avoidance [23]. Hank et al. focused on generating near-time-optimal trajectories during the preprocessing phase. Their approach combines optimal trajectories with a reactive module comprising three fuzzy logic-based submodules [24]. Yi Zhu et al. proposed a DH-Bug (Distance Histogram Bug) reactive algorithm, which was fused with the A* algorithm while ensuring robust convergence [25]. Jose et al. utilized the A* algorithm to find waypoints. To find collision-free paths, they used a probabilistic roadmap guided by a potential field map function [26]. Taha et al. leveraged the RRT-Connect algorithm for path planning and introduced a sliding modebased controller for obstacle avoidance in cluttered environments with simple geometrical obstacles [27]. Pablo et al. implemented their work on an Ackerman-based model car, employing Dijkstra's algorithm as a global planner and enhancing it with a Time Elastic Band mechanism to avoid obstacles [28]. In another study by Jian et al., an iterative optimization method was utilized in the global planning stage, while a sampling-evaluation strategy was adopted in the local planning stage [29]. Woojin et al. integrated the Dynamic Window Approach with pre-generated waypoints for mobile robots with constrained velocity and acceleration [30]. Louis et al. proposed the Parallel Elliptic Limit-Cycle (PELC) approach for global planning along with its variant based on multi-criteria optimization to form a reactive component for obstacle avoidance [31].

Next, we discuss recent advancements in hybrid algorithm approaches for the 3D path planning landscape. Wzorek et al. incorporated a probabilistic path planning algorithm with a reactive Optimal Reciprocal Collision Avoidance (ORCA) controller. Rapidly Exploring Random Tree (RRT) was used for global path planning, while the ORCA controller was used to avoid collisions with obstacles [32]. Kwangjin et al. used RRT for path planning and its variant with linear Model Predictive Control for obstacle avoidance and path-following [33]. Lin et al. showed that cluster-based UAV navigation path planning is possible through a combination of modified Artificial Potential Methods and evolutionary

algorithms [34]. Matthias et al. designed real-time planning solutions for micro-aerial vehicles. Their proposed solution combined the A* algorithm for global path planning with a grid-based method for obstacle avoidance [35]. Vincent et al. provided path-planning solutions for fixed-wing UAVs by fusing Particle Swarm Optimization with a Genetic Algorithm [36]. Xiobing et al. developed a hybrid approach based on the Grey Wolf Optimizer (GWO) and Differential Evolution (HGWODE) algorithms to provide an optimised solution for UAV path planning [37]. Chengzhi et al. [38] introduced a novel approach called RLGWO, which combines Grey Wolf Optimization (GWO) with Reinforcement Learning (RL), to address the challenging UAV path planning problem. The performance of RLGWO was evaluated through simulations in three distinct case studies within a 3D space in the presence of eight stationary obstacles. In a study by Soheila et al. [39], a novel approach named TLBO* was introduced based on the fusion of Teaching-Learning-based Optimization (TLBO) and Genetic Algorithm techniques Research work by Qu et al. [40] combined three distinct algorithms: the Dijkstra Algorithm, Genetic Algorithm (GA), and Artificial Potential Field (APF). The Dijkstra algorithm was employed to determine the shortest path, while the APF algorithm identified a feasible path by guiding the UAV towards smoother regions while avoiding potential threats. A GA algorithm was then applied to further optimize the path generated by the other two algorithms. Veronica et al. employed a hybrid approach to tackle multiple objectives, which they achieved by including PDDL for task planning, Bayesian Networks (BN) for feasibility evaluation, and the K-Nearest Neighbors (KNN) algorithm for path planning and obstacle avoidance selection [41].

Mobile robot path planning in two dimensions is known to be an NP-hard problem [42], and the complexity of 3D path planning is further escalated by the extra dimension. Increased complexity arises from the expansion of the search space. This has significant implications, particularly for small UAVs, as these must swiftly respond to new information despite their limited computational capabilities. If the environment of the UAVs is discretized, the volume of points to be taken into account experiences a substantial surge in 3D space. Furthermore, when seeking better accuracy, if more points are generated more densely then the complexity escalates even further. The significance of devising efficient mechanisms becomes even more pronounced when tackling path planning challenges with dynamic objects in 3D space. In the majority of current research, the complete computational processing is carried out onboard. Moreover, most existing research has only considered static obstacles. In contrast, in our research we suggest a method for enhancing UAV efficiency through cloud computing and wireless technology. In order to offload computing effort around optimal path computation, the UAV wirelessly connects to cloud computing infrastructure using the Software as a Service (SaaS) model.

3. System Description and Investigated Problem

We examine a non-holonomic Unmanned Aerial Vehicle (UAV) navigating within a confined 3D space denoted as S, where $S \subset \mathbb{R}^3$. In the Cartesian coordinate system at τ time, $P_{uav}(\tau) := [x_u(\tau), y_u(\tau), z_u(\tau)]$ describes the UAV's location within the environment. The UAV's motion is expressed as $\dot{P}_{uav}(\tau)$, and is characterized as follows [20].

$$\begin{aligned} \dot{x}_{u} &= v_{uav}(\tau) sin(\theta_{\alpha}(\tau)) cos(\theta_{\beta}(\tau)) \\ \dot{y}_{u} &= v_{uav}(\tau) cos(\theta_{\alpha}(\tau)) cos(\theta_{\beta}(\tau)) \\ \dot{z}_{u} &= v_{uav}(\tau) sin(\theta_{\beta}(\tau)) \end{aligned}$$
(1)

The orientation angles of the UAV, namely, the yaw (θ_{α}) and pitch (θ_{β}) angles, are measured in the anticlockwise direction. The UAV's rolling motion is not included in the calculation. While the roll angle is essential for certain types of aircraft (e.g., fixed-wing aircraft), it may not be as crucial for nonholonomic UAVs, such as quadcopters, which primarily rely on variations in thrust and rotor speeds to control their movement. We assume that the UAV's roll angle remains within acceptable limits to preserve altitude stability. As incorporating roll control would introduce added complexity, it was avoided.

$$\begin{aligned} P_{uav}(\tau) &= v_{uav}(\tau) Q_d(\tau), \\ \dot{Q}_d(\tau) &= c_{uav}(\tau), \end{aligned} \tag{2}$$

where $Q_d(\tau)$ and $c_{uav}(\tau)$ are three-dimensional vectors ($Q_d(\tau) \in \mathbb{R}^3$ and $c_{uav}(\tau) \in \mathbb{R}^3$) with $||Q_d(\tau)|| = 1$ for all the values for time τ ($\tau > 0$). Setting the norm of Q_d to 1 guarantees that the UAV's overall speed remains constant while allowing for variations in the direction of motion. Additionally, the following conditions must be met for all the values of τ :

$$||c_{uav}(\tau)|| \le C_{uav},$$
$$(Q_d(\tau), c_{uav}(\tau)) = 0,$$

where C_{uav} is a constant provided by the user. The notation ||.|| signifies the Euclidean norm, (*, *) represents the scalar dot product, and the minimum turning radius, denoted as R_s , is determined as follows.

$$R_s = \frac{v_{uav}}{C_{uav}}$$

The environment S encompasses both discrete known obstacles and unknown obstacles, denoted as $\mathcal{H} = \{\mathcal{H}_1, \ldots, \mathcal{H}_i, ..., \mathcal{H}_n\}$. We denote ' K_s ', where ($K_s \leq n$) known static obstacles can take on various shapes, including convex and non-convex. In the case of unknown obstacles \mathcal{H}_i , it is assumed that each obstacle can be contained within a finite sphere of radius A_i . The unknown obstacles can either be stationary or in motion; in the latter case, they are characterized by velocities $S_h = \{s_1, \ldots, s_i, \ldots, s_n\}$.

The UAV is equipped with an onboard sensor system capable of detecting obstacles H_i when they enter its sensing range A_d . Furthermore, it can compute the shortest Euclidean distance d_i between itself and these obstacles.

The primary objective is to ensure collision-free navigation for a non-holonomic UAV commencing from an initial position P_{start} and finishing at a fixed target position P_{goal} . This navigation should be achieved within an optimal time while avoiding collisions with both static and dynamic obstacles and while maintaining a minimum safe distance D_{safe} from the obstacles. This objective holds true as long as $\{P_{start}, P_{goal}\}$ are within the set \mathcal{N} , defined as $\mathcal{N} := S \setminus \mathcal{H}$.

Assumption 1. The velocity of dynamic obstacles is assumed to be constant and slower than that of the UAV, i.e., $S_h < V_a < V_{max}$, where V_a is a given positive constant and V_{max} is the maximum speed attainable by the UAV.

Assumption 2. For any dynamic obstacle \mathcal{H}_i , the UAV possesses knowledge about its velocity \mathbf{s}_i and the radius of its sphere \mathbf{A}_i .

The Livox AVIA solid-state lidar system is a good fit for putting Assumption 2 into practice. It has multiple benefits specific to aerial applications, such as its light weight and adaptable size for small-scale UAVs. Thanks to its low false alarm rate, it has an extended object detection range that makes identifying obstacles during fast flight more feasible. Moreover, it is capable of up to 2500 Hz data output, enabling prompt identification of moving objects. Even in stationary operation, its non-repetitive scanning technology provides improved resolution. Incorporating the Livox AVIA enhances the practicality and precision of the proposed navigation strategy, ensuring accurate determination of obstacle locations and speed vectors in real-world UAV missions [43].

4. Global/Reactive UAV Path Planning Algorithm

The planning methodology, as illustrated in Figure 1, consists of three fundamental constituents: global path planning, reactive control, and transition guidelines. The sequence commences with the UAV transmitting its initial and destination positions to the cloud computing facility through a wireless communication channel. Subsequently, the remote computer computes the optimal route and relays the trajectory waypoints to the UAV. The UAV adheres to these waypoints with the assistance of a trajectory-tracking algorithm. If the UAV is unable to locate the global path after an obstacle avoidance manoeuvre, it will stop the motion and safely land. Alternatively, the UAV may attempt to rejoin the original global path if it remains feasible. The choice of action in such situations is based on factors such as the proximity to the original path, the presence of additional obstacles, and real-time computational capabilities.



Figure 1. Structure of the Path Planning Algorithm.

During the trajectory tracking phase, if obstacles are detected by the UAV's onboard sensors, it switches to the reactive controller. The reactive controller is responsible for executing obstacle avoidance manoeuvres. When an obstacle has been successfully avoided, the UAV reverts to the global planning algorithm and resumes following the trajectory waypoints.

It is important to note that this method assumes accurate localization of the UAV within the environment. The global planner operates on the remote computing facility, while the reactive controller is executed by the UAV's onboard computing facility.

Global Path Planning. The first step is to calculate the shortest collision-free path between the starting point P_{start} and the destination point P_{goal} , with static barriers while satisfying condition (2). In this paper, the Dynamic Programming algorithm (see, e.g., [44,45]) is utilised to achieve the objective. Dynamic programming leverages the principle of optimality to efficiently find optimal solutions. In our case, it finds the optimal path by breaking the space down into a grid. Each grid cell represents a state, and dynamic programming is employed to find the optimal path by recursively updating the cost-to-come values for each state while adhering to the principles of optimality and constraint. This method efficiently finds the best path by considering intermediate points and backtracking from the goal state. It is particularly effective for grid-based environments, although it may encounter scalability issues in high-dimensional spaces. Its curse of dimensionality is overcome through the use of cloud computing facilities. In order to efficiently solve dynamic programming problems in real-time, high-performance computing resources are often necessary. This typically involves dedicated servers, cloud computing instances, or clusters equipped with powerful processors and substantial memory. Services offered by cloud providers such as Amazon Web Services, Microsoft Azure, and Google Cloud are commonly utilized to access scalable and high-performance computing resources. The choice of

hardware specifications should align with the specific computational requirements of the application's real-time constraints. The UAV should complete the journey within a finite time limit of T_g while operating inside environment S.

In Figure 2, the shaded regions represent barriers within the environment S. The space S undergoes discretization into a grid consisting of N nodes. Provided that there are no obstacles impeding their connection, every node establishes connections with its neighboring nodes. Two specific nodes, denoted as P_{start} and P_{goal}, are introduced into this grid. For any given point with coordinates (x_k, y_k, z_k) , we designate P_k as the nearest node. The cost of traversing from node P_k to its neighbouring node is computed through the Euclidean distance formula. The function $F_k(x_u(\tau), y_u(\tau), z_u(\tau))$ serves as a vital model for evaluating the proximity of a UAV to the closest point of a static obstacle k. In this context, the variable k spans from 1 to K_s and the function $F_k(\cdot)$ adheres to the constraint $0 \leq F_k(x_u, y_u, z_u) \leq 1$. A value of 0 for $F_k(x_u, y_u, z_u)$ indicates a significant distance between the UAV and a static obstacle *j*. Conversely, $0 \le F_k(x_u, y_u, z_u) \le 1$ signifies that the static obstacle is within the UAV's proximity, with the value of $F_k(x_u, y_u, z_u)$ indicating the degree of closeness. It is crucial to note that the function $F_k(x_u, y_u, z_u)$ is inherently a monotonically decreasing function, transitioning from a value of 1 to 0; this particular characteristic reflects the diminishing closeness of obstacles to the UAV as the value of $F_k(x_u, y_u, z_u)$ decreases. In environments characterized by dynamic obstacles, it is evident that maintaining a safe distance from static obstacles is of paramount importance when seeking to avoid potential collisions. To illustrate this we can define $F_k(x_u, y_u, z_u)$ for a static spherical obstacle as follows. Let the center of a given obstacle k be represented by $P_k = (x_c, y_c, z_c)$. In this scenario, the 3D Euclidean distance between two points in 3D space is expressed as $E_d(.,.)$. Moreover, we define two constants: $0 < r_1 < r_2$. The function $F_k(x_u, y_u, z_u)$ is characterized as follows: (1) $F_k(x_u, y_u, z_u) = 1$ when $E_d(P_k, (x_u, y_u, z_u)) \le r_1$ and (2) $F_k(x_u, y_u, z_u) = 0$ when $E_d(P_k, (x_u, y_u, z_u)) \ge r_2$. For the intermediate range where $r_1 \leq E_d(P_k, (x_u, y_u, z_u)) \leq r_2$, we can express $F_k(x_u, y_u, z_u)$ as any monotonically decreasing function that transitions from 1 to 0. The specific form of this function depends on the spatial relationship between the point (x_u, y_u, z_u) and the given static obstacle k. The optimal trajectory $T(\tau)$ of the UAV adheres to the condition that $T(0) = P_{\text{start}}$ at $\tau = 0$ seconds and that $T(t_f) = P_{goal}$ at $\tau = t_f$ seconds, where $0 < t_f \leq T_g$.



Figure 2. Three-dimensional environment map.

The cost function for evaluating the trajectory cost of the UAV is as follows:

$$\sum_{k=1}^{K_{\rm s}} \int_0^{t_f} F_k(x_u(\tau), y_u(\tau), z_u(\tau)) d\tau \tag{3}$$

The aim is to minimize a given function as specified in (3). We can consider a given positive integer, denoted as M, and define σ as $\sigma := \frac{T_g}{M}$. For all $a = \{0, 1, ..., m, ..., M\}$, we introduce sets $\mathcal{P}(a) \subset S$ as follows. The set $\mathcal{P}(M)$ exclusively comprises the target point P_{goal} , expressed as $\mathcal{P}(M) := P_{goal}$. For all a < M, the set $\mathcal{P}(a)$ encompasses all points $\hat{E} \in S$ that have a solution for (2) over the interval $[0, \sigma]$, where $v_{uav}(\tau) = V_{max}$ and $c_{uav}(\tau)$ remain constant for all $\tau \in [0, \sigma]$. Additionally, the set must satisfy the conditions $P_{uav}(0) = \hat{E}$, $P_{uav}(\sigma) \in \mathcal{P}(m+1)$, and $P_{uav}(\tau) \in S$ for all $\tau \in [0, \sigma]$.

The function $R(a, \hat{E})$ is defined on $\mathcal{P}(a)$ for a = 0, 1, ..., M as follows:

$$R(M, \mathsf{P}_{\mathsf{goal}}) := 0, \quad R(a, \hat{E}) :=$$

$$\min\left[R(a+1, P_{uav}(\sigma)) + \sum_{k=1}^{K_s} \int_0^{\sigma} F_k(x_u(\tau), y_u(\tau), z_u(\tau)) dt\right]$$
(4)

Here, the minimum value is computed across all potential solutions. Additionally, we define $\hat{G}(a, \hat{E})$ as the vector for which the minimum in (4) (with $P_{uav}(0) = \hat{E}$) is obtained with $c_{uav}(\tau) \equiv \hat{G}(a, \hat{E})$. We additionally introduce the set \mathcal{J} , which consists of indexes $a \in \{0, 1, \ldots, M-1\}$ such that $P_{start} \in \mathcal{P}(a)$. If \mathcal{P} is not empty, we designate $a_0 \in \mathcal{J}$ as the index at which the minimum in $\min_{a \in \mathcal{J}} R(a, P_{start})$ is achieved, i.e., for all $a \in \mathcal{J}$, $a = a_0$.

Now, the UAV's trajectory is constructed with an initial condition $P_{uav}(0) = P_{start}$, and we utilize the following t_f and control inputs:

$$t_{f} := \frac{(M - a_{0})\mathbf{T}_{g}}{\mathbf{N}} = (M - a_{0})\sigma$$

$$v_{uav}(\tau) = V_{max} \quad \forall \tau \in [0, t_{f}],$$

$$c_{uav}(\tau) = \hat{G}(a, P_{uav}((a - a_{0})\sigma))$$
(5)

for all $\tau \in [(a - a_0)\sigma, (a - a_0 + 1)\sigma)$ and all $a = a_0, a_0 + 1, \dots, M - 1$.

The set of waypoint data, denoted as $\mathcal{B} := \{B_1, B_2, ..., B_R\}$, is generated and stored. The number of waypoints W_r is user-defined and specifies the resolution of the trajectory data. When the UAV receives the dataset \mathcal{B} , it begins to move. The summary of global navigation algorithm pseudo code is given by Algorithm 1. Global path planning is considered a non-critical task carried out through remote computers. Waypoints are wirelessly transferred to the UAVs; in the event that the UAV is unable to receive these waypoints, it will not commence its journey.

Algorithm 1 Global Navigation Algorithm Pseudocode

Discretize the environment into a grid with N nodes.

Define the starting point P_{start} and the destination point P_{goal} .

Calculate the cost of traversing between nodes using the Euclidean distance formula.

Define the cost function to evaluate the trajectory cost of the UAV.

Introduce sets $\mathcal{P}(a)$ representing points with solutions for the specified condition over a defined interval.

Define function $R(a, \hat{E})$ for cost evaluation and trajectory optimization across $\mathcal{P}(a)$.

Construct the UAV's trajectory with initial conditions, duration, and control inputs. Generate a set of waypoint data \mathcal{B} with user-defined resolution.

The UAV follows the trajectory and adjusts based on received waypoint data.

Reactive (Local) Path Planning. The proposed reactive (local) path planner utilizes a Sliding Mode Control (SMC)-based reactive controller. SMC is based on the principles of the Lyapunov theory. It involves a "sliding surface" in a system's state space. When the system's state crosses this surface, the control strategy swiftly adjusts to maintain the system on this surface. This methodology ensures the desired system behaviour even when uncertainties and disturbances are present. The robustness of SMC in handling these challenges is a result of its reliance on the actual system state rather than a precise mathematical model.

As the UAV approaches a moving obstacle, it seamlessly transitions from its optimal path to an obstacle-avoidance mode to avert collisions. The algorithm for avoiding collisions with obstacles is detailed below:

$$c_{uav}(\tau) = C_{uav} \mathcal{K}(\dot{p}_d(\tau), L_i^{(h)}(\tau))$$

$$v_{uav}(\tau) = ||L_i^{(h)}(\tau)||$$
(6)

$$c_{uav}(\tau) = 0; \quad v_{uav}(\tau) = V_{max} \tag{7}$$

where the function \mathcal{K} is expressed as

$$\mathcal{K} = \begin{cases} 0 & K(\gamma, \xi) = 0\\ ||K(\gamma, \xi)||^{-1}K(\gamma, \xi) & K(\gamma, \xi) \neq 0 \end{cases}$$

where K is defined as

$$K(\gamma,\xi) := \xi - (\gamma,\xi)\gamma$$

In the above equation, γ and ξ are unit vectors. The expression $L_i^{(h)}(\tau)$ is described as

$$L_i^{(j)}(\tau) := (V_{max} - \mathbf{s}_i)(\cos(\phi_i^j(\tau)), \sin(\phi_i^j(\tau))), \tag{8}$$

where j = 1, 2, while $\phi_i^j(\tau)$ is described as

$$\phi_i^1(\tau) := \psi_i^{(1)}(\tau) + \delta_0, \tag{9}$$

$$\phi_i^2(\tau) := \psi_i^{(2)}(\tau) + \delta_0, \tag{10}$$

where δ_0 is the constant avoidance angle maintained by the UAV (illustrated in Figure 3) and the angles $\phi_i^{(1)}(\tau)$ and $\phi_i^{(2)}(\tau)$ are obtained through measurements by the onboard sensors of the UAV. The expanded cone is formed by the two tangent lines extending from the UAV's present location to the obstacle, as shown in Figure 3.



Figure 3. Enlarged vision cone from UAV.

Switching Rule. The "Switching Rule" dictates when the UAV should transition between optimal path following and obstacle avoidance and vice versa. When an obstacle $\mathcal{H}i$ is detected within the sensor's sensing range A_d , the switching action is initiated. Let the closest waypoint reached by the UAV be denoted as B_s . Assuming that the UAV can estimate the size of the dynamic obstacle, let B_{s+z} represent the pseudo-target for the obstacle avoidance algorithm, where $s < s + z < W_r$. We define a positive constant D_{safe} as the safe distance between the obstacle and the UAV. The distance to the nearest obstacle measured by the UAV's onboard sensor is denoted as d_i . The switching modes are as follows:

SR1: when $(d_i < D_{safe})$ and $(\dot{d}_i < 0)$, this indicates that the UAV is approaching the object. In this scenario, the UAV transitions from following the path from the global planner to the one from the reactive planner using the control method (6). The pseudo-target of the reactive control is set to $G_0 := B_{s+z}$. The UAV attempts to reach a waypoint B_{s+z} which is positioned *z* units ahead of B_s .

SR2: the UAV switches from (6) to (7) when it is aligned toward G_o and when $(d_i \le 1.1w_i)$ [20]. The parameter w_i is defined as

$$w_i := \frac{\mathbf{A}_i}{\cos(\delta_0)} - \mathbf{A}_i,\tag{11}$$

where A_i is the radius of the sphere enclosing the obstacle.

SR3: the UAV transitions from (7) to the global path tracking algorithm when it approaches sufficiently close to the interim target G_o while satisfying the condition $|G_o - P_{uav}(\tau)| < \Delta$, with Δ being a small positive constant value defined by the user.

5. Computer Simulation Results

In this section, we present results obtained through simulations using the proposed navigation strategy. All of our simulations used MATLAB software. The 3D environment map depicted in Figure 2 was accessible to the global path planning algorithm using dynamic programming. The values of the parameters used in the simulations are as follows: $\mathbb{N} = 500$, M = 100, $T_g = 200$, $V_{max} = 0.5 \text{ m/s}$, $C_d = 2 \text{ rad/s}$, $R_s = 0.25 \text{ m}$, $P_{start} = [5,5,5]$, $P_{goal} = [47,39.1,11.4]$, $R_i = 0.8 \text{ m}$, $\delta_0 = 30^\circ$, $D_{safe} = 0.2 \text{ m}$. The values of z and Δ were determined through a trial and error method. The function $F_k(x_u(\tau), y_u(\tau), z_u(\tau))$ in Section 4 was defined as $F_k(x_u(\tau), y_u(\tau), z_u(\tau)) := E_d(P_k(x_u(\tau), y_u(\tau), z_u(\tau))) \times (\frac{1}{r_1 - r_2} - \frac{r_2}{r_1 - r_2})$. Both the UAV's starting position and its target location were known for the purposes of cloud computing. It should be pointed out that sliding-mode type algorithms similar in spirit to the algorithm in this paper have been successfully implemented with real UAVs in [20,46], where solutions to the real problems typical for such systems such as determining the locations of obstacles and their speed vectors were provided, as well as a real description of the actually implementation.

In this section, we describe the following two scenarios.

Scenario 1. In this simple scenario, the environment is devoid of dynamic obstacles. The UAV initiates the navigation process by transmitting its start and target points to a remote computer via wireless connection. The remote computer calculates the optimal route and returns trajectory waypoints. The UAV adeptly follows these waypoints, as illustrated in Figures 4–6. Throughout this journey, the UAV demonstrates its ability to navigate collision-free around pre-known static obstacles. RRT* is known for its effectiveness in path planning for holonomic vehicles. Thus, we compared our strategy with the RRT*-type algorithm from [47]. As our UAV was non-holonomic, we employed spline curves to accommodate the constraints in order to make a fair comparison possible. Figure 7 represents the variation of the cost function for the proposed algorithm and the RRT* algorithm. Figure 7 clearly demonstrates that our algorithm consistently maintains a lower cost function value throughout the UAV's trajectory, highlighting its superiority in path planning and collision avoidance. Additionally, it is worth noting that our approach outperforms the RRT* algorithm in several other aspects. Our proposed algorithm significantly reduces the time required to reach the target, taking around 112 s compared to approximately 118 s with

RRT*. A shorter distance is covered by the UAV when using the proposed algorithm, at approximately 55 m in contrast to roughly 59 m for RRT*. The total value of the cost function at the end of the trajectory is substantially lower for the proposed algorithm, measuring approximately 2015, while RRT* yields a higher value of approximately 2446. It is worth mentioning that the cost function was calculated at regular intervals of 0.125 s throughout the UAV's journey. These additional metrics emphasize the superior performance of our proposed algorithm.



Figure 4. Case I: complete trajectory.



Figure 5. Case I: trajectory top view.



Figure 6. Case II: trajectory side view.



Figure 7. Case I: comparison of cost function.

Scenario 2. In this scenario, multiple dynamic obstacles are introduced that intersect with the optimal path. When the UAV's sensors detect the first obstacle, the switching rules are activated, as illustrated in Figure 8. A zoomed-in view is shown in Figure 9. After UAV successfully avoiding an obstacle, the UAV resumes tracking the path (see Figure 10). As the UAV continues to follow the waypoints, another moving obstacle approaches towards the

UAV, attempting to intercept it, as shown in Figure 10. The UAV successfully manoeuvres around the obstacle and continues moving toward the target. The entire trajectory is illustrated in Figures 11 and 12. The trajectories in both scenarios exhibit similarity, with the exception of occasional deviations from the optimal path by the UAV that were necessary to avoid obstacles in Scenario 2. These results affirm the practicality and adaptability of our navigation strategy for UAVs in dynamic and obstacle-laden environments. Furthermore, they underscore the robustness of our proposed strategy against real-time changes in the environment, highlighting its potential in a wide range of applications.



Figure 8. Case II: trajectory view I.



Figure 9. Case II: trajectory view I zoom-in.



Figure 10. Case II: trajectory view II.



Figure 11. Case II: complete trajectory view III.



Figure 12. Case II: complete trajectory side view.

6. Conclusions

We have introduced an innovative approach that seamlessly integrates a global path planning system and a robust obstacle avoidance strategy. The proposed navigation approach is intended for non-holonomic Unmanned Aerial Vehicles (UAVs) operating within complex 3D environments. Our solution addresses the intricate challenges posed by both static and dynamic obstacles. In particular, we have paid special consideration to the non-holonomic constraints that are often inherent to UAVs. Our method leverages the power of cloud computing coupled with efficient wireless communication to provide UAVs with optimal and collision-free trajectories. Seeking both safety and precision, we have employed a sliding-mode-based reactive control algorithm to ensure that the UAV is able to make real-time adjustments when confronted by obstacles. Through extensive computer simulations, we have demonstrated the feasibility of our proposed approach and showcased its effectiveness in overcoming the intricate hurdles posed by 3D environments. The effectiveness of the developed navigation method was additionally confirmed by comparison with another state-of-art method. In conclusion, our research paves the way for further advancements in UAV navigation in complex 3D settings. We have successfully amalgamated global path planning and obstacle avoidance while acknowledging nonholonomic constraints. Nevertheless, there are certain limitations to our simulation-based approach. Our future work will focus on bridging the gap between simulations and real-world implementation.

7. Future Work

Our research presents a stepping stone for future investigations to solve the challenging problem of non-holonomic Unmanned Aerial Vehicle (UAV) navigation in 3D environments with multiple dynamic obstacles. The next step is to move from simulations to real-world implementation. The focus should be on experimental tests and validation in complex and obstacle-rich environments by leveraging advanced sensor technologies and specialized hardware. Anticipating the movements of dynamic obstacles through predictive algorithms and integrating machine learning techniques for enhanced path planning and obstacle avoidance represent key priorities. Further research might explore multi-UAV collaboration, energy-efficient navigation, and human–UAV interaction, and could address evolving regulatory requirements as well. Scalability and adaptability to UAVs of various sizes and capabilities are among the potential directions for future research with the aim of advancing the field of UAV navigation and safety.

Author Contributions: Conceptualization, S.C.V. and A.V.S.; methodology, S.C.V. and A.V.S.; software, S.C.V. and S.L.; validation, S.C.V., A.V.S. and S.L.; formal analysis, A.V.S.; writing—original draft preparation, S.C.V. and A.V.S.; writing—review and editing, S.C.V.; visualization, S.C.V. and S.L.; supervision, A.V.S.; project administration, A.V.S.; funding acquisition, A.V.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work received funding from the Australian Government via grant AUSMURIB000001, associated with ONR MURI grant N00014-19-1-2571.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Tan, Y.; Li, G.; Cai, R.; Ma, J.; Wang, M. Mapping and modelling defect data from UAV captured images to BIM for building external wall inspection. *Autom. Constr.* 2022, 139, 104284. [CrossRef]
- Klamt, T.; Rodriguez, D.; Schwarz, M.; Lenz, C.; Pavlichenko, D.; Droeschel, D.; Behnke, S. Supervised Autonomous Locomotion and Manipulation for Disaster Response with a Centaur-Like Robot. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–8. [CrossRef]
- 3. Mishra, B.; Garg, D.; Narang, P.; Mishra, V. Drone-surveillance for search and rescue in natural disaster; *Comput. Commun.* 2020, 156, 1–10. [CrossRef]
- 4. Mammarella, M.; Comba, L.; Biglia, A.; Dabbene, F.; Gay, P. Cooperation of unmanned systems for agricultural applications: A theoretical framework. *Biosyst. Eng.* **2022**, 223, 61–80. [CrossRef]
- Asadzadeh, S.; de Oliveira, W.J.; de Souza Filho, C.R. UAV-based remote sensing for the petroleum industry and environmental monitoring: State-of-the-art and perspectives. J. Pet. Sci. Eng. 2022, 208, 109633. [CrossRef]
- 6. Huang, H; Savkin, A.; Huang, C. Autonomous Navigation and Deployment of UAVs for Communication, Surveillance and Delivery; Wiley: Hoboken, NJ, USA, 2022.
- Hassanalian, M.; Rice, D.; Abdelkefi, A. Evolution of space drones for planetary exploration: A review. *Prog. Aerosp. Sci.* 2018, 97, 61–105. [CrossRef]
- 8. Rhiat, Ahmed; Chalal, Lamine; Saadane, Allal. A Smart Warehouse Using Robots and Drone to Optimize Inventory Management. In Proceedings of the Future Technologies Conference (FTC) 2021, Virtual, 28–29 October 2021; Volume 1, 2022; pp. 475–483.
- Cristiani, Davide; Bottonelli, Filippo; Trotta, Angelo; Di Felice, Marco. Inventory Management through Mini-Drones: Architecture and Proof-of-Concept Implementation. In Proceedings of the 2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), 2020, Cork, Ireland, 31 August–3 September 2020; pp. 317–322. [CrossRef]
- Verma, S.C.; Savkin, A.V. A Hybrid Global/Reactive Algorithm for Collision-Free Navigation of Mobile Ground Robots in Environments with Steady and Moving Obstacles. In Proceedings of the 2022 IEEE International Conference on Robotics and Biomimetics (ROBIO), Jinghong, China, 5–9 December 2022; pp. 1321–1326. [CrossRef]
- Verma, S.C. Safe Navigation of a Non-Holonomic Robot with Low Computational-power in a 2D Dynamic Environment. In Proceedings of the 2022 Australian & New Zealand Control Conference (ANZCC), Gold Coast, Australia, 24–25 November 2022; pp. 13–17. [CrossRef]
- 12. Hunziker, D.; Gajamohan, M.; Waibel, M.; D'Andrea, R. Rapyuta: The RoboEarth Cloud Engine. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 438–444. [CrossRef]
- 13. Hoy, M.; Matveev, A.S.; Savkin, A.V. Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey. *Robotica* **2015**, *33*, 463–497. [CrossRef]
- Véras, L.G.D.O.; Medeiros, F.L.L.; Guimaráes, L.N.F. Systematic Literature Review of Sampling Process in Rapidly-Exploring Random Trees. *IEEE Access* 2019, 7, 50933–50953. [CrossRef]
- Rios, L.H.O.; Chaimowicz, L. A Survey and Classification of A* Based Best-First Heuristic Search Algorithms. In 20th Brazilian Symposium on Artificial Intelligence, Sao Bernardo do Campo, Brazil, 23–28 October 2010, Proceedings; da Rocha Costa, A.C., Vicari, R.M., Tonidandel, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 253–262.

- 16. Xiao, X.; Liu, B.; Warnell, G.; Stone, P. Motion planning and control for mobile robot navigation using machine learning: A survey. *Auton. Robot.* **2022**, *46*, 569–597. [CrossRef]
- 17. Chowdhury, R.; Subramani, D. Optimal Path Planning of Autonomous Marine Vehicles in Stochastic Dynamic Ocean Flows Using a GPU-Accelerated Algorithm. *IEEE J. Ocean. Eng.* **2022**, *47*, 864–879. [CrossRef]
- Saadi, A.A.; Soukane, A.; Meraihi, Y.; Gabis, A.B.; Mirjalili, S.; Ramdane-Cherif, A. UAV Path Planning Using Optimization Approaches: A Survey. Arch. Comput. Methods Eng. 2022, 29, 4233–4284. [CrossRef]
- Cheng, C.T.; Fallahi, K.; Leung, H.; Tse, C.K. A Genetic Algorithm-Inspired UUV Path Planner Based on Dynamic Programming. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 2012, 42, 1128–1134. [CrossRef]
- Wang, C.; Savkin, A.V.; Garratt, M. A strategy for safe 3D navigation of non-holonomic robots among moving obstacles. *Robotica* 2018, 36, 275–297. [CrossRef]
- Gracia, L.; Garelli, F.; Sala, A. Reactive Sliding-Mode Algorithm for Collision Avoidance in Robotic Systems. *IEEE Trans. Control Syst. Technol.* 2013, 21, 2391–2399. [CrossRef]
- Wiig, M.S.; Pettersen, K.Y.; Krogstad, T.R. A 3D Reactive Collision Avoidance Algorithm for Nonholonomic Vehicles. In Proceedings of the 2018 IEEE Conference on Control Technology and Applications (CCTA), Copenhagen, Denmark, 21–24 August 2018; pp. 67–74. [CrossRef]
- Xu, X.; Xie, J.; Xie, K. Path Planning and Obstacle-Avoidance for Soccer Robot Based on Artificial Potential Field and Genetic Algorithm. In Proceedings of the 2006 6th World Congress on Intelligent Control and Automation, Dalian, China, 21–23 June 2006; IEEE: Piscataway, NJ, USA, 2006. [CrossRef]
- Hank, M.; Haddad, M. A hybrid approach for autonomous navigation of mobile robots in partially-known environments. *Robot. Auton. Syst.* 2016, 86, 113–127. [CrossRef]
- 25. Zhu, Y.; Zhang, T.; Song, J.; Li, X. A new hybrid navigation algorithm for mobile robots in environments with incomplete knowledge. *Knowl. Based Syst.* **2012**, *27*, 302–313. [CrossRef]
- Sanchez-Lopez, J.L.; Pestana, J.; Campoy, P. A robust real-time path planner for the collision-free navigation of multirotor aerial robots in dynamic environments. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; IEEE: Piscataway, NJ, USA, 2017. [CrossRef]
- Elmokadem, T. A Hybrid Strategy for Safe Navigation of Mobile Robots in Partially-Known Dynamic Environments. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; IEEE: Piscataway, NJ, USA, 2018. [CrossRef]
- 28. Marin-Plaza, P.; Hussein, A.; Martin, D.; de la Escalera, A. Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles. *J. Adv. Transp.* **2018**, 2018, 1–10. [CrossRef]
- Wang, Y.; Li, W.; Jiang, R. A Novel Hybrid Algorithm Based on Improved Particle Swarm Optimization Algorithm and Genetic Algorithm for Multi-UAV Path Planning with Time Windows. In Proceedings of the 2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 16-18 December 2022; IEEE: Piscataway, NJ, USA, 2022. [CrossRef]
- Chung, W.; Kim, S.; Choi, M.; Choi, J.; Kim, H.; Moon, C.B.; Song, J.B. Safe Navigation of a Mobile Robot Considering Visibility of Environment. *IEEE Trans. Ind. Electron.* 2009, 56, 3941–3950. [CrossRef]
- Adouane, L. Reactive versus cognitive vehicle navigation based on optimal local and global PELC*. *Robot. Auton. Syst.* 2017, 88, 51–70. [CrossRef]
- Wzorek, M.; Berger, C.; Doherty, P. A Framework for Safe Navigation of Unmanned Aerial Vehicles in Unknown Environments. In Proceedings of the 2017 25th International Conference on Systems Engineering (ICSEng), Las Vegas, NV, USA, 22–24 August 2017; IEEE: Piscataway, NJ, USA, 2017. [CrossRef]
- 33. Yang, K.; Gan, S.K.; Sukkarieh, S. An Efficient Path Planning and Control Algorithm for RUAV's in Unknown and Cluttered Environments. *J. Intell. Robot. Syst.* 2009, *57*, 101–122. [CrossRef]
- 34. Lin, C.L.; Li, Y.H.; Aouf, N. Potential-Field-Based Evolutionary Route Planner for the Control of Multiple Unmanned Aerial Vehicles. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* 2010, 224, 1229–1242. [CrossRef]
- Nieuwenhuisen, M.; Behnke, S. Layered Mission and Path Planning for MAV Navigation with Partial Environment Knowledge. In *Intelligent Autonomous Systems* 13; Springer: Berlin/Heidelberg, Germany, 2015; pp. 307–319. [CrossRef]
- 36. Roberge, V.; Tarbouchi, M.; Labonte, G. Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. *IEEE Trans. Ind. Inform.* **2013**, *9*, 132–141. [CrossRef]
- 37. Yu, X.; Jiang, N.; Wang, X.; Li, M. A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning. *Expert Syst. Appl.* **2023**, *215*, 119327. [CrossRef]
- 38. Qu, C.; Gai, W.; Zhong, M.; Zhang, J. A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning. *Appl. Soft Comput.* 2020, *89*, 106099. [CrossRef]
- Ghambari, S.; Idoumghar, L.; Jourdan, L.; Lepagnot, J. An Improved TLBO Algorithm for Solving UAV Path Planning Problem. In Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 6–9 December 2019; IEEE: Piscataway, NJ, USA, 2019. [CrossRef]
- Wang, Z.; Li, Y.; Li, W. An approximation path planning algorithm for fixed-wing UAVs in stationary obstacle environment. In Proceedings of the Proceedings of the 33rd Chinese Control Conference, Nanjing, China, 28–30 July 2014; IEEE: Piscataway, NJ, USA, 2014. [CrossRef]

- 41. Vannini, V.; de Moura Souza, G.; Toledo, C.F.M. Harpia: A hybrid system for agricultural UAV missions. *Smart Agric. Technol.* **2023**, *4*, 100191. [CrossRef]
- 42. Erickson, L.; LaValle, S. A Simple, but NP-Hard, Motion Planning Problem. *Proc. AAAI Conf. Artif. Intell.* 2013, 27, 1388–1393. [CrossRef]
- Nam, Dinh Van; Kim, Gon-Woo. Solid-State LiDAR based-SLAM: A Concise Review and Application; In Proceedings of the 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju Island, South Korea, 17–20 January 2021; pp. 302–305. [CrossRef]
- 44. Savkin, A.V.; Evans, R.J. Hybrid Dynamical Systems: Controller and Sensor Switching Problems; Birkhauser: Boston, MA, USA, 2002.
- 45. Goodwin, G.C.; Graebe, S.F.; Salgado, M.E. Control System Design; Prentice Hall: Kent, OH, USA, 2001.
- 46. Elmokadem, T.; Savkin, A.V. A method for autonomous collision-free navigation of a quadrotor UAV in unknown tunnel-like environments. *Robotica* 2022, 40, 835–861. [CrossRef]
- 47. Mechali, O.; Xu, L.; Wei, M.; Benkhaddra, I.; Guo, F.; Senouci, A. A Rectified RRT* with Efficient Obstacles Avoidance Method for UAV in 3D Environment. In Proceedings of the 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Suzhou, China, 29 July–2 August 2019; pp. 480–485. [CrossRef]

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.