

## Article

# A Fusion Approach for UAV Onboard Flight Trajectory Management and Decision Making Based on the Combination of Enhanced A\* Algorithm and Quadratic Programming

Shuguang Sun <sup>1,2</sup>, Haolin Wang <sup>1,\*</sup>, Yanzhi Xu <sup>1</sup>, Tianguang Wang <sup>1</sup>, Ruihua Liu <sup>1</sup> and Wantong Chen <sup>1</sup> 

<sup>1</sup> College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China; sgsun@cauc.edu.cn (S.S.); 2021022151@cauc.edu.cn (Y.X.); 2022022207@cauc.edu.cn (T.W.); rhliu@cauc.edu.cn (R.L.); wtchen@cauc.edu.cn (W.C.)

<sup>2</sup> Laboratory of Technology and Equipment of Tianjin Urban Air Transportation System, Civil Aviation University of China, Tianjin 300300, China

\* Correspondence: 2021021019@cauc.edu.cn



**Citation:** Sun, S.; Wang, H.; Xu, Y.; Wang, T.; Liu, R.; Chen, W. A Fusion Approach for UAV Onboard Flight Trajectory Management and Decision Making Based on the Combination of Enhanced A\* Algorithm and Quadratic Programming. *Drones* **2024**, *8*, 254. <https://doi.org/10.3390/drones8060254>

Academic Editor: Andrey V. Savkin

Received: 4 May 2024

Revised: 29 May 2024

Accepted: 4 June 2024

Published: 8 June 2024

**Correction Statement:** This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.



**Copyright:** © 2024 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/>).

**Abstract:** The rapid advancement of unmanned aerial vehicle (UAV) technologies has led to an increasing demand for UAV operations in low-altitude, high-density, and complex airspace such as mountains or urban areas. In order to handle complex scenarios and ensure flight safety for UAVs with different flight missions beyond visual line of sight in such environments, a fusion framework of onboard autonomous flight trajectory management and decision-making system using global strategical path planning and local tactical trajectory optimization combination is proposed in this paper. The global strategical path planning is implemented by an enhanced A\* algorithm under the multi-constraint of UAV positioning uncertainty and obstacle density to improve the safety and cost-effectiveness. The local tactical trajectory optimization is realized using quadratic programming to ensure smoothness, kinematic feasibility, and obstacle avoidance of the planned trajectory in dynamic environments. Receding-horizon control is used to ensure the flight path and trajectory planning efficiently and seamlessly. To assess the performance of the system, a terrain database and a navigation system are employed for environment and navigation performance simulation. The experimental results confirm that the fusion approach can realize better safety and cost-effectiveness through path planning with kino-dynamic feasible trajectory optimization.

**Keywords:** unmanned aerial vehicle; trajectory management; decision making; enhanced A\* algorithm; quadratic programming

## 1. Introduction

There has been a dramatic increase in the adoption of unmanned aerial vehicles (UAVs) for transportation, cargo/package logistics, precision agriculture, remote sensing, search and rescue, and video surveillance in recent years, which requires transformative and disruptive airborne technology to support a low-altitude airspace ecosystem designed to transport people and things to locations which are not traditionally served by the current modes of air transportation, including mountainous rural areas and the more challenging and complex urban environments [1]. Different from current commercial aircrafts, which fly by following the specified waypoints and routes under mature operation regulations, UAVs have to cope with a new flight scenario under more complex airspace and unexpected non-cooperative dynamic obstacles, leading to the demand for UAVs to manage feasible and flexible flight trajectory planning and real-time decision making simultaneously for mission fulfilling and obstacle avoidance. Autonomy for flight trajectory management and decision making would be an essential solution for UAV operation, especially in case of a loss of the command and control (C2) link between the pilot's control station and the UAV, dynamic obstacle encounter, and some other unexpectedly emergent events.

NASA Langley Research Center introduced design guidelines for automated flight path management in 2021 [2]. According to NASA, several functions will need to be performed by advanced onboard flight path management automation systems with access to broad information regarding the vehicle and the airspace environment and having a purposeful design, which manages various aspects of an aircraft's flight path to ensure flight safety in a complex traffic environment and conformance to operational constraints [2]. The autonomy of the onboard system would reduce the dependency on communication links and enable greater scalability. An onboard autonomous flight trajectory management and decision-making (AFTM and DM) algorithm according to the UAV mission requirements, airspace environment, vehicle performance, and characteristics is presented and simulated in this paper to support UAV self-flight.

There are several challenging aspects that UAV onboard autonomous flight trajectory management and decision making have to face:

- Highly complex, dynamic, and dense airspace. Low-altitude airspace in which UAVs fly includes known static obstacles and unexpected dynamic obstacles such as birds, weather, balloons, etc. demanding flexible trajectory management based on these factors and other stakeholder activity in the airspace volume;
- Multiple safety and feasibility restrictions and constraints. Not only the airspace environment and the dynamic and kinematic characteristics of a certain UAV type are essential concerns, but the performance degradation of onboard navigation systems in the case of the Global Positioning System (GPS) being denied or masked should be considered to ensure the safety and feasibility of the planned trajectory;
- Trajectory planning and optimization algorithms would consume large onboard calculation resources and may have poor real-time performance at a large airspace scale with high complexity, while onboard autonomous flight trajectory management needs to adjust the trajectory in time to ensure the spatiotemporal coordination of UAV operation.

The presented AFTM and DM will provide the dynamic flight trajectory planning function under the aforementioned restrictions and constraints, while meeting the following objectives: feasible for mission completion, deconflicted from hazards, coordinated with other traffic, flexible to accommodate future disturbances, and optimized to meet business objectives, which means the UAV AFTM and DM is a complex multi-objective optimization and decision-making problem under multisource heterogeneity constraints, where the selected algorithms affect the efficiency and effectiveness of trajectory management directly.

Recent studies on UAV trajectory management have been approached from two main directions. One is flight path planning performing at a global level. Some conventional and intelligent algorithms are used for global optimal path seeking according to selected criteria (e.g., energy consumption, flight time, or operational costs minimization) and the imposed UAV dynamic and kinematic characteristics constraints [3–19]. In this case, the main focus is economical path seeking. The other approach concerns the optimization of a local dynamic flight trajectory within the local sensing range of onboard sensors, while performing dynamic obstacle avoidance by using the localization of UAV positions and sensing information of the dynamic environment with an optimization strategy or certain resolution rules. In this case, the smoothness of the trajectory and successfully collision avoidance are two key objectives, which possibly lead to trajectory deviation from the UAV's original planned path [20–26].

Planning algorithms used for global flight path planning can be categorized as follows: the random sampling-based method, graph searching-based method, and evolutionary method. Each of them has its own strengths and shortcomings. The Rapidly-exploring Random Tree (RRT) algorithm, a typical sampling-based path planning method, prioritizes completeness, efficiency, and search speed in a complex environment, though it may not yield optimal path results. Karaman et al. [3] extended RRT and proposed an asymptotically optimal sampling-based method, i.e., RRT\*. Gammell et al. [4] used an informed strategy to enhance the optimality of the RRT algorithm. Alam et al. [5] proposed an energy-efficient

motion planning approach using a flight cost-based RRT (FC-RRT\*) algorithm to optimize the trajectory flight cost compared with the conventional RRT and RRT\* algorithms as well as the kinematic solutions. Chen et al. [6] established a double-tree to enhance the efficiency of the RRT algorithm. Gao et al. [7] presented an improved DB-RRT\* algorithm coupling with the sliding mode guidance based on angle constraints with finite-time convergence to control the UAV flying to the destination autonomously. Besides, since the sampling-based algorithm is not relying on a well-established map, Gao et al. [8] presented a framework using a safe-region RRT\* algorithm for online generating safe and dynamically feasible trajectories directly on the point cloud, which could constrain the position and kino-dynamics of the trajectory entirely within the flight corridors and given physical limits. The A\* algorithm is one of the most popular graph-searching methods, using discretized spatial nodes to calculate the cost value of each traversed node to obtain the feasible path, while a heuristic function is used to search for approximate optimal solutions based on empirical rules under acceptable computational costs to find solutions [9,10]. Moreover, there are some variants of A\*, such as Anytime Repairing A\* (ARA) [11], Jump Point Search (JPS) [12], and Theta\* [13]. Most research efforts have focused on the trade balance between optimality and computational efficiency. Motivated by the purpose of improving efficiency, Szczerba et al. [14] introduced the sparse A\* search method, which reduces the expansion of redundant neighbor nodes based on constraint conditions, and Harabor, et al. [12] introduced the JPS algorithm, which utilizes a novel neighbor expansion rule that expands key nodes adjacent to obstacles leading to fast computations and safety risks. Liu et al. [15] developed a three-dimensional JPS algorithm for path search in a complex environment and ensured safety by building up safe flight corridors. Evolutionary algorithms like the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) converge to optimal results after multiple iterations, and usually excel at handling complex challenges with various constraints and environment uncertainties [16–18]. Learning-based algorithms can be used to cope with global path planning as well. Park et al. [19] used reinforcement learning to cope with the multi-agent UAV transportation management problem. The main limitation of learning-based algorithms is the lack of comprehensibility of the algorithm's logics. The sampling-based algorithm and evolutionary algorithms share similar limitations, since both of them rely on randomness and the path planning results may not be optimal but can asymptotically approach optimality after a large number of iterations. Graph-searching algorithms, especially the A\* algorithm, ensures optimality in a well-established voxel map when the heuristic function is admissible.

Local dynamic flight trajectory optimization generates and assesses time-dependent trajectories within local areas under the support of onboard sensors. The lattice-based method, akin to the graph-based approach, samples various discrete motion states to construct a lattice graph, using graph-search or other techniques for motion planning, as employed by Pivtoraiko and Kelly [20]. The advantage of lattice-based planning is the motion feasibility, since the state space and control space of a certain type of UAV is generated based on its motion model. Mueller et al. [21] generated various feasible motion primitives between two nodes and selected the one with the lowest cost as the planned trajectory. Liu et al. [22] sampled a quadrotor's control space and created a lattice graph for local trajectory planning. Nevertheless, lattice-based methods require the sampling of several discrete points in the UAV's motion state space, which would consume a huge scale of computation resources to generate a several-step lattice graph, and might not cover all possible states, and thus resulting in a suboptimal trajectory. Mellinger and Kumar [23] specified the differential flatness of multi-rotor UAVs and modeled the motion planning as a numerical optimization problem of polynomial trajectories, generating minimum-snap trajectories by solving a convex optimization problem. Liu et al. [15] generated convex safe flight corridors (SFCs) to model a UAV's local free-space, and they used the SFCs as the constraints to enhance the safety of Mellinger and Kumar's work [23]. Wang et al. [24] summarized and proposed a geometric modeling method for various types of motion and environment constraints for UAV trajectory optimization. Local trajectory optimization can

be realized by forming a convex optimization problem, and this method can be enhanced through the justification of optimization objectives and constraint conditions. Shen et al. [25] used optimization-based trajectory planning on problems of UAV landing on moving carriers. Blasi et al. [26] used clothoid curves to represent smooth trajectories and used a similar optimization-based planning algorithm to generate trajectories.

For flight trajectory planning methods of UAVs operated in complex areas, Ippolito et al. [27] conducted the modeling and simulation of multi-copter UAV motion in urban environments. Baculi and Ippolito [28] identified key risks associated with UAV operation in low-altitude, high-density urban environments characterized by dynamic and uncertain conditions. These risks potentially pose hazards to UAVs and include factors like degraded global navigation satellite system (GNSS) or radio frequency (RF) signals, atmospheric uncertainty, the presence of other aircrafts, and dynamic objects on the ground. Refs. [29,30] focus on the applications and collision avoidance of UAVs in urban or low-altitude environments, but they did not cover all concerns related to UAV operation in complex environments. Shen's thesis [31] introduced a complete micro-UAV structure and implementation, including an onboard navigation system and perception sensors, planning system, control system, and actuators. Refs. [32,33] provide established efficient motion planning frameworks for a micro-UAV in a small-scale environment. Studies like [27,28] contributed to identify characteristics of the UAV's operation environment, while [31] provided an actual implementation method of a complete autonomous UAV system. Refs. [29,30] contributed to deal with collision avoidance of dynamic obstacles while [32,33] focused on fast and efficient planning and control in a static environment.

From the comprehensive analysis can be concluded as that most of the global flight path planning algorithms are computationally expensive and time-consuming, which contrasts with the limited computational power available on UAVs, not to mention real-time planning, and the existence of dynamic obstacles and navigation degradation would make things worse.

The proposed AFTM and DM approach takes into account both global and local requirements and constraints for the UAV flight trajectory planning using fusion optimization. The fusion optimization approach combining an enhanced A\* algorithm and the quadratic programming (QP) is proposed in this paper to solve the aforementioned problems and helps the implementation of onboard AFTM and DM. The key features and benefits of the proposed approach are as follows:

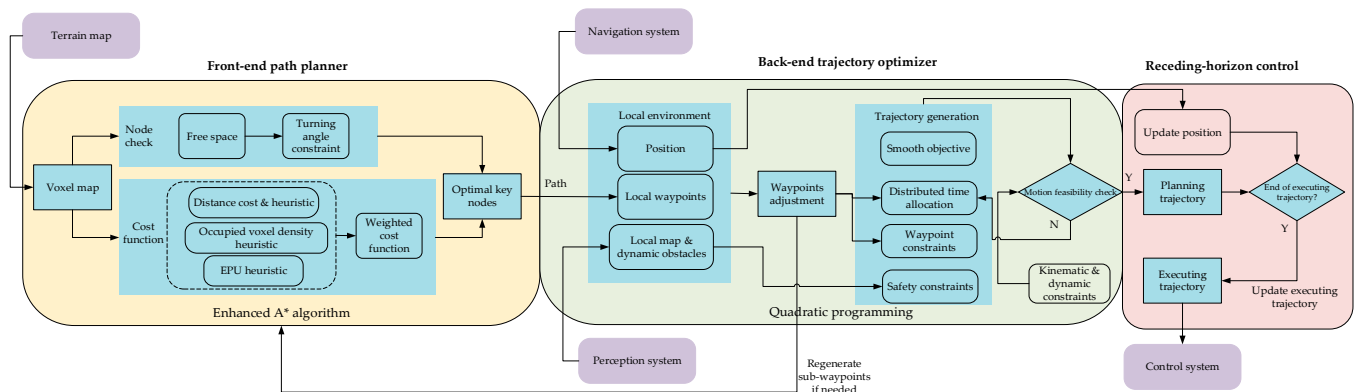
1. The proposed optimization framework divides the problem of large-scale real-time optimization in limited time into two steps with a distributed time allocation. An enhanced A\* algorithm is used to generate a global optimal flight path intermittently with respect to a self-defined cost function, while quadratic programming is used to smooth the short-term trajectory suitable for UAV kinematic characteristics and dynamic constraints in real time. The fusion approach splits the multiple constraints into the A\* algorithm and quadratic programming based on their respective features, reducing the computational complexity and resource consumption, and enhancing the system's real-time performance;
2. Navigation performance is introduced as a soft constraint to guarantee a safety margin and the feasibility of the planned trajectory. To discourage the planned trajectory from approaching the boundary of its constraint set too closely, a navigation system estimated position uncertainty (EPU) heuristic cost is employed in the A\* algorithm to inflate the UAV voxels to guarantee safety margins. This can prevent the UAV from traversing narrow passages, while avoiding navigation performance degradation caused by GPS masking as much as possible;
3. Quadratic programming based on the UAV's current position and local dynamic environment information is used for real-time trajectory optimization under the constraints of the UAV's maneuver performance and dynamic obstacles within its detection range, with reduced computation consumption and improved trajectory optimization response efficiency.

The rest of this paper is organized as follows. Section 2 describes the overall fusion framework, including the methodology of global strategical flight path planning using the enhanced A\* algorithm, and local tactical dynamic trajectory optimization using quadratic programming. Section 3 presents the simulation results of the proposed fusion approach, and verification of the feasibility. Finally, Section 4 concludes the paper, with a brief summary and suggestions for future research work.

## 2. Methods

### 2.1. Fusing Approach of Flight Trajectory Management

According to the characteristics, the objectives of the flight trajectory management and decision making are classified into two groups in our approach: long-term related aspects, such as safety and economy, and short-term related aspects, such as flight feasibility and dynamic obstacle collision avoidance. This classification provides the opportunity to reach the management goals in different steps with a combination of suitable methods. Figure 1 presents the proposed AFTM and DM framework of the fusion approach. The “front-end path planner” executes the strategical path planning on the global level employing an enhanced A\* algorithm, while the “back-end trajectory optimizer” executes the local tactical trajectory management using quadratic programming. “Receding-horizon control” is used to balance the computation time of trajectory management with the real-time execution of the planned trajectory.



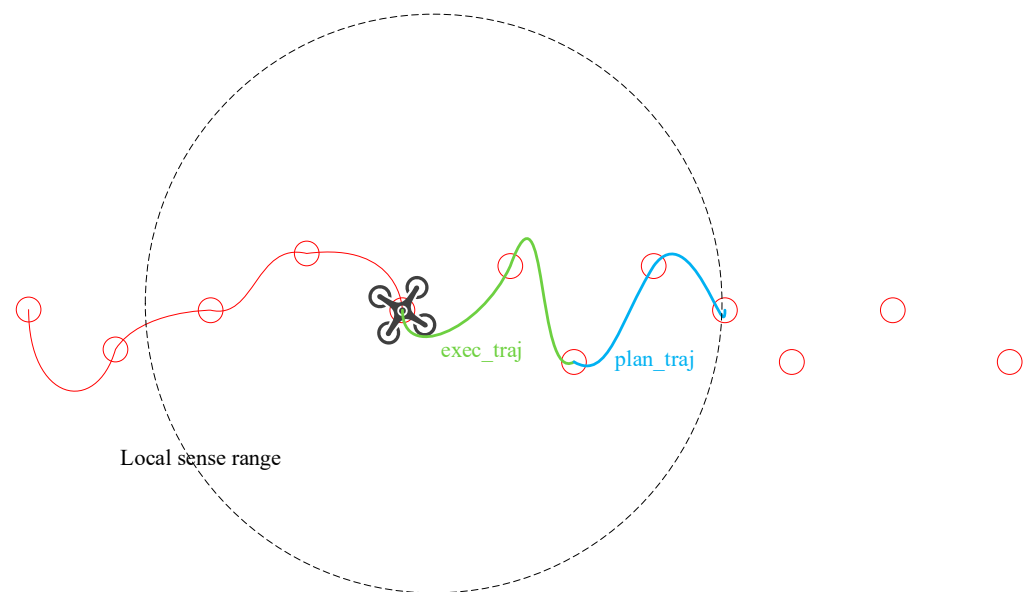
**Figure 1.** Framework of the fusion approach of the front-end path planner and the back-end trajectory optimizer.

The front-end path planner is at a strategical view and used for finding the safety-critical and economical flight path. The planned flight path is prescribed with a sequence of waypoints that connects the vehicle’s start location to the goal set, named the global path. Considering that the A\* algorithm has a fast search speed, high efficiency, and capability to ensure finding the shortest path that meets specific conditions, the A\* algorithm can introduce different information into the heuristic functions according to the application scenarios, making the search direction more accurate and reducing the computation time. Meanwhile, the enhanced A\* algorithm is chosen for global flight path searching. Here, the UAV’s angle change constraint is used to improve the searching efficiency while the EPU cost and obstacle density cost are introduced to ensure that the vehicle’s flight path remains sufficiently clear of preflight-known hazards (static obstacles such as terrain and buildings).

As the front-end path planner does not account for the UAV’s dynamics, path following may result in a dynamically infeasible problem. To overcome this limitation, the back-end trajectory optimizer is tasked with computing a set of dynamically feasible time-parameterized splines that interpolates the waypoints generated by the front-end path planner. The back-end trajectory optimizer is at a tactical view, used for local trajectory fly feasibility refinement and dynamic obstacle collision avoidance trajectory designing based on the vehicle maneuver performance and the dynamically changing airspace environment around the UAV’s current positions. It generates local trajectory sub-waypoints based on

global path information, onboard navigation systems, and environment perception sensors. The back-end trajectory optimizer also implements a safety check and adjustment for local sub-waypoints before trajectory generation and will refine the local sub-waypoints through the front-end path planner if the safety check is not passed. Quadratic programming is adopted to generate a time-continuous, smooth, and dynamically feasible polynomial trajectory with the ability of obstacle collision avoidance, which realizes the UAV real-time tactical decision making.

The local trajectory optimization is implemented within the onboard sensors' detection range and managed as two separate parts, current executing trajectory and the following planning trajectory (i.e., predicted trajectory). The predicted trajectory generated by quadratic programming is stored as the planning trajectory after the feasibility and safety check, which occurred each time the local map is updated. The planning trajectory will be regenerated if it fails the feasibility and safety check. Once the UAV reaches the endpoint of the executing trajectory, the executing trajectory will be updated by the stored planning trajectory and a new prediction period starts at the same time for the next planning trajectory. This method is similar to the concept of receding-horizon control [34,35] and helps the UAV manage its flight trajectories efficiently and seamlessly. As presented in Figure 2, the current executing trajectory is marked as "exec\_traj" in green, while the predicted planning trajectory is marked as "plan\_traj" in blue. The red line in this figure represents the flight trajectory, while the red circles represent the waypoints of the global path. Receding-horizon control maintains a balance between the execution time of motion and the computation time of planning and adjustment, providing an efficient scheme of local trajectory management.



**Figure 2.** Receding-horizon control.

## 2.2. Strategical Path Planner

The front-end strategical path planner employed here is based on an enhanced A\* algorithm with consideration of a UAV's kinematic constraints and safety distance (contingent upon onboard navigation performance), which plans an optimal path consisting of a set of key waypoints using the static map. A three-dimensional terrain map is processed to a 3-D occupation voxel map in which each voxel is  $30\text{ m} \times 30\text{ m} \times 30\text{ m}$  and is used for flight path planning at a global view.

Algorithm 1 represents the overall details of the enhanced A\* algorithm used in this paper.

**Algorithm 1: The Enhanced A\* Algorithm**


---

Input: Map,  $\mathbf{p}_{init}$ ,  $\mathbf{p}_{goal}$ , NAV,  $\mathbf{W}$   
Result: A path from  $\mathbf{p}_{init}$  to  $\mathbf{p}_{goal}$   
Initialization: Initialize the priority queue with  $\mathbf{p}_{init}$  and assign  $g(\mathbf{p}_{init}) = 0$  and  $g(n) = \infty$  for all other nodes in Map.  
Repeat:  
    Remove the node “n” with the lowest  $f(n) = g(n) + h(n)$  from the priority queue;  
    If the node “n” =  $\mathbf{p}_{goal}$ , return TRUE; break;  
    Mark node “n” as expanded;  
    For all unexpanded neighbor nodes “m” of node n:  
        If node check (m, n) then  
            If  $g(m) = \infty$   
                Push node “m” into the queue;  
                 $g(m) = g(n) + C_{nm}$ ;  
                Calculate heuristic distance cost  $h_1$ ;  
                Calculate obstacle density cost  $h_2(m, \text{Map})$ ;  
                Calculate navigation performance cost  $h_3(m, g(m), \text{NAV}, \text{Map})$ ;  
                 $h(m) = \mathbf{W} \cdot [h_1 \quad h_2 \quad h_3]$ ;  
            If  $g(m) > g(n) + C_{nm}$  then  
                 $g(m) = g(n) + C_{nm}$ ;  
                Calculate navigation performance cost  $h_3(m, g(m), \text{NAV}, \text{Map})$ ;  
                 $h(m) = \mathbf{W} \cdot [h_1 \quad h_2 \quad h_3]$ ;  
        end  
    end  
end

---

**2.2.1. Enhancement on Node Check of A\* Algorithm**

During each step of the A\* search, each neighbor of the current node is assessed and stored in a priority queue based on an extension rule and an evaluation function. Usually, there would be a large number of redundant nodes stored in the queue without any enhancement of the node check, affecting the searching speed. Frequent and sharp turns are not suitable maneuvers for UAV flight, so the extension rule has been enhanced by introducing constraints on angle change in order to generate a smooth and control-friendly flight path and to improve the computation efficiency by decreasing the number of expanded nodes. The angle change constraints include the horizontal and vertical directions normalized based on the kinematic characteristics of the UAV and the terrain map’s resolution. During each extension step, nodes that fail to meet the angle change constraints are disregarded. This refinement aims to produce path planning results that are faster and more readily adaptable for the local trajectory planner. For a specific node in the 3-D occupancy-grid map, the horizontal angle change can be calculated as follows:

$$\Delta angle_{horizontal} = \cos^{-1} \left( \frac{a_1 \cdot a_2}{|a_1| |a_2|} \right), \quad (1)$$

where

- $a_1 = [x(n) - x(n-1), y(n) - y(n-1)]$ ;
- $a_2 = [x(m) - x(n), y(m) - y(n)]$ .

The angle change in the vertical direction can be calculated as follows:

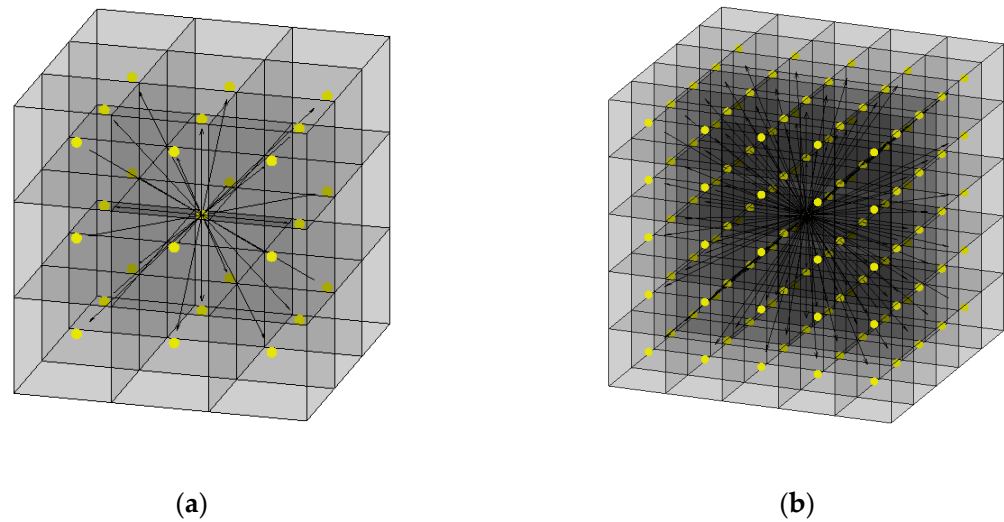
$$\Delta angle_{vertical} = v\_angle(m) - v\_angle(n), \quad (2)$$

where

- $v\_angle(m) = \tan^{-1} \left( \frac{z(m) - z(n)}{\sqrt{[x(m) - x(n)]^2 + [y(m) - y(n)]^2}} \right)$ ;
- $v\_angle(n) = \tan^{-1} \left( \frac{z(n) - z(n-1)}{\sqrt{[x(n) - x(n-1)]^2 + [y(n) - y(n-1)]^2}} \right)$ .

In both Equations (1) and (2), the coordinates of the current node  $n$  are  $x(n)$ ,  $y(n)$ , and  $z(n)$ , while the coordinates of its parent node are  $x(n-1)$ ,  $y(n-1)$ , and  $z(n-1)$ , and the coordinates of the neighbor node  $m$  which is currently being checked and extended are  $x(m)$ ,  $y(m)$ , and  $z(m)$ .

The conventional extension rule for the 3-D A\* algorithm typically examines neighboring nodes in a  $3 \times 3 \times 3$  field, as depicted in Figure 3a. However, when incorporating angle change limitations, the available motion options within this framework may prove insufficient for the A\* search. To address this issue, as illustrated in Figure 3b, the searching range is extended from  $3 \times 3 \times 3$  to  $5 \times 5 \times 5$ , effectively expanding the angle choices to align with the enhancement strategy.



**Figure 3.** Conventional extension range and the expanded range used in this paper. (a) Conventional extension range of the 3-D A\* algorithm; (b) expanded extension range of the 3-D A\* algorithm.

### 2.2.2. Enhancement of Cost Function of A\* Algorithm

While introducing angle change limitations improved the A\* algorithm's performance and alignment with UAV kinematic characteristics, it remained insufficient for the UAV's safe operation in complex airspace. Safety and positioning uncertainty are critical concerns that cannot be overlooked. To enhance security, this paper incorporates obstacle density and positioning uncertainty costs within the heuristic function.

The cost function of the conventional A\* algorithm for node  $n$  is represented as  $f(n) = g(n) + h(n)$ , which comprises the distance  $g(n)$  from the start node to node  $n$  and the heuristic function  $h(n)$ . The distance  $g(n)$  can be expressed as follows:

$$g(n) = g(n-1) + \sqrt{[x(n) - x(n-1)]^2 + [y(n) - y(n-1)]^2 + [z(n) - z(n-1)]^2}, \quad (3)$$

where  $(n-1)$  represents the parent node of node  $n$ .

In this paper,  $g(n)$  is also used to estimate the time by dividing the UAV's speed and determine the satellites' information in order to estimate the navigation performance of the onboard navigation system. The heuristic function  $h(n)$  of the conventional A\* algorithm represents the estimated distance between node  $n$  and the target node. The most common heuristic functions are the Manhattan distance, the Euclidean distance, and the diagonal distance. In many cases, the Manhattan distance is the longest, and it is often not admissible. The Euclidean distance is the shortest and is generally the most admissible. The diagonal distance is usually shorter than the Manhattan distance but longer than the Euclidean distance, encouraging the algorithm to favor diagonal extension, and it is typically admissible. When compared to the Euclidean distance, the diagonal heuristic is supposed to expand fewer redundant nodes and find the shortest path faster. When compared to the Manhattan distance, the diagonal heuristic usually results in shorter paths.

Consequently, the diagonal distance is selected as the heuristic distance for this paper. The calculation of the 3-D diagonal distance is as follows:

$$h_1(n) = dx + dy + dz + (\sqrt{3} - 3)\min(dx, dy, dz) + (\sqrt{2} - 2)[\text{mid}(dx, dy, dz) - \min(dx, dy, dz)], \quad (4)$$

where  $dx = |x(\text{target}) - x(n)|$ ,  $dy = |y(\text{target}) - y(n)|$ , and  $dz = |z(\text{target}) - z(n)|$ .  $\min(dx, dy, dz)$  represents the minimum value of  $(dx, dy, dz)$ .  $\text{mid}(dx, dy, dz)$  represents the second minimum value of  $(dx, dy, dz)$ .

Most of the UAV onboard navigation systems are GPS or GPS/low-cost INS integrated ones, whose positioning performance relies on the numbers and geometry of visible satellites. A sufficient safety margin guarantee in the flight path can not only improve the UAV terrain avoidance but also prevent the UAV from traversing narrow passages, where GPS satellites would be masked and degrade the navigation performance. Moreover, a large safety margin gives the back-end trajectory optimizer more room for trajectory optimization and does not disrupt the computational efficiency of the proposed quadratic programming problem, but an unreasonably large safety margin may make path planning impossible or the UAV's frequent tactical behavior face coasting obstacles. To discourage the UAV's planned path from approaching the boundary of its constraint requirement set too close or unreasonably far, the obstacle density cost within an estimated local sensing range and onboard navigation system performance EPU are employed to introduce some soft constraint. The obstacle density cost is represented as Equation (5).

$$h_2(n) = \text{num}_{\text{occupied\_voxel}}(n, \text{map}_{\text{local}}^*), \quad (5)$$

According to the ICAO Performance-Based Navigation (PBN) manual [36], airspace planners use the required navigation performance (RNP-x) to determine route spacing and separation minima, where x refers to the lateral navigation accuracy which is expected to be achieved at least 95 percent of the flight time by the population of aircraft operating within the airspace, route, or procedure; the aircraft are permitted to operate within the airspace, route, or procedure when the EPU value of their onboard navigation systems is much lower than x, which means, even though there is no RNP specification defined for UAV low-altitude operation, the EPU is suitable for UAV positioning performance and flight path safety margin determination. A typical approach to guarantee the safety margin is to introduce offsets of user-defined size or inflate the occupied voxels. The EPU cost is implemented as a soft constraint in this paper and defined as Equation (7).

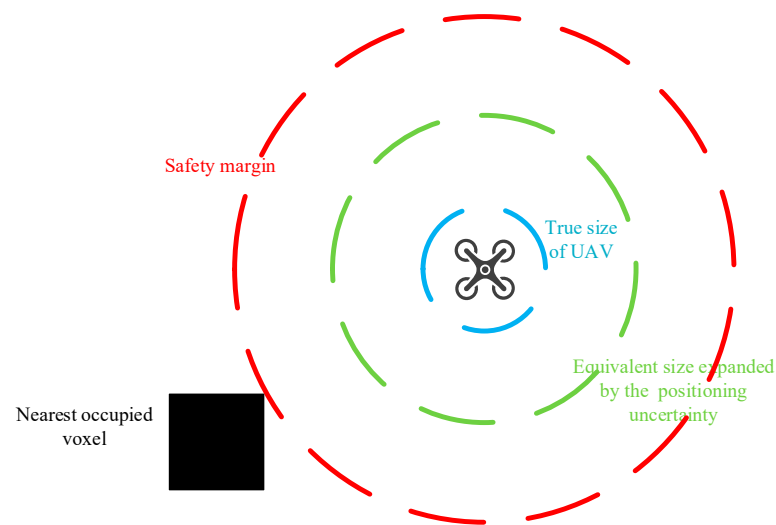
$$EPU = 2\sqrt{\sigma_{\text{east}}^2 + \sigma_{\text{north}}^2 + \sigma_{\text{up}}^2}. \quad (6)$$

$$h_3(n) = \frac{r_{\text{UAV}} + EPU}{\text{dis}_{\text{nearest\_obs}}(n, \text{map}_{\text{local}}^*)}. \quad (7)$$

In Equation (6),  $\sigma_{\text{east}}$ ,  $\sigma_{\text{north}}$ , and  $\sigma_{\text{up}}$  denote the estimated deviation of the positioning error in three directions. In Equation (7),  $r_{\text{UAV}}$  denotes the true radius of the size of the UAV, and  $EPU$  is the estimated EPU at node  $n$  calculated by Equation (6), while  $\text{dis}_{\text{nearest\_obs}}(n, \text{map}_{\text{local}}^*)$  is the distance between node  $n$  and its nearest occupied voxel, named as the safety margin in Figure 4. The navigation performance cost,  $h_3(n)$ , is the ratio of the UAV's equivalent radius and the distance margin for the safety requirements. This cost helps to ensure the path maintaining a safe distance from obstacles, accounting for positioning uncertainty with the UAV's radius of size.

The heuristic function used in the enhancement of the A\* algorithm is a weighted sum of  $h_1(n)$ ,  $h_2(n)$ , and  $h_3(n)$ , represented as Equation (8), where  $w_1 + w_2 + w_3 = 1$ .

$$h(n) = w_1 \times h_1(n) + w_2 \times h_2(n) + w_3 \times h_3(n). \quad (8)$$



**Figure 4.** Equivalent size and safety margin of UAV for the AFTM and DM in this paper.

### 2.3. Tactical Trajectory Optimizer

To cope with the real-time dynamic trajectory management and decision making in a dynamic local environment, the objective of the tactical trajectory optimizer is to generate a time-dependent, smooth, and collision avoidance trajectory. This local trajectory should be generated with consideration of both waypoint constraints provided by the global path and the UAV's environmental perception within its sensing range, while also adhering to the UAV's kinematic and dynamic constraints.

The process of local tactical trajectory management contains a local environment update, local waypoints adjustment according to the updated local environment, and trajectory generation and adjustment. The convex optimization method is an efficient scheme to generate the optimized polynomial trajectory under well-defined optimization objectives and constraints, and the optimization problem for trajectory management can be modeled as a quadratic programming problem for the parameter optimization of the polynomial trajectories. The benefit of modeling the tactical trajectory management into a convex optimization model is that the output of a convex optimization solver is supposed to be both a local and global optimum solution.

#### 2.3.1. Waypoints Adjustment Based on Real-Time Local Environment Update

The local environmental information of the UAV is obtained by the onboard situation perception system [37], building an accurate local map which has a much higher resolution than the global map, based on the terrain and dynamic obstacles detected. The perception system uses cooperative or non-cooperative technologies to detect the dynamic obstacles within a certain range, including their current positions and motion trends. This paper uses the perception system which generates local maps with each voxel as  $3\text{ m} \times 3\text{ m} \times 3\text{ m}$ .

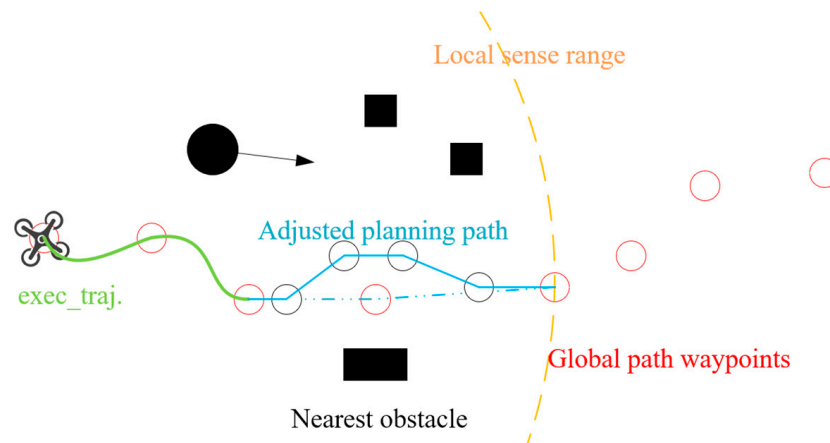
The waypoints adjustment consists of the interpolation of local waypoints, collision check and replan of the local path, and adjustment for waypoint positions according to the local environment update. Since the global path planning is operated in a large-scale map with low resolution, the number of waypoints within the local sensing range may not satisfy the requirements of receding-horizon control, and establishing a constraint condition for implementing the quadratic programming may not be applicable. Therefore, sub-waypoints interpolation is necessary before trajectory generation.

The local map may differ from the global map because of perception uncertainties and previously unknown obstacles, so AFTM and DM must operate the conflict check and collision avoidance. The enhanced A\* algorithm will be used if the local planning path has conflict with local static obstacles. Then, the method for adjustment of the local waypoints is realized by displacing the waypoints away from their nearest obstacles by a distance

relevant to the relative velocity and a predefined minimum time of the closest point of approach (CPA), inspired by the concept of the artificial potential field method [38], and the conflict detection method in ACAS Xu [39]. Figure 5 represents the waypoints interpolation and adjustment for the planning trajectory in the local environment. The planning path is extended, and each waypoint will be displaced from the nearest obstacle if the distance between them is greater than the UAV's current equivalent radius of size. The displacement will be along the direction of  $(\mathbf{pos}_{nearest\_obstacle} - \mathbf{pos}_{UAV})$  and the range will be calculated as Equation (9).

$$\begin{aligned} range &= (r_{UAV} + EPU) - |\mathbf{pos}_{nearest\_obstacle} - \mathbf{pos}_{UAV}| \\ displacement &= range \cdot \frac{\mathbf{pos}_{nearest\_obstacle} - \mathbf{pos}_{UAV}}{|\mathbf{pos}_{nearest\_obstacle} - \mathbf{pos}_{UAV}|} + \mathbf{vel}_{nearest\_obstacle} \cdot \tau \end{aligned} \quad (9)$$

where  $(r_{UAV} + EPU)$  denotes the UAV's current equivalent radius of size and  $\tau$  denotes the predefined minimum time of the closest point of approach. This method does not need to establish a complete potential field or a Euclidean signed distance field (ESDF) map [33] for the local environment while the information of the nearest obstacle is sufficient and will improve the efficiency of collision avoidance.



**Figure 5.** Interpolation and adjustment of local waypoints.

### 2.3.2. Quadratic Programming Implementation

One of the purposes of trajectory management is to generate smooth trajectories, and the smoothness usually means minimizing the rate of change of input of a UAV's dynamic motion system in order to minimize the energy consumption. The motion of a UAV can be represented as a linear control system, and a one-step state change can be expressed as  $\mathbf{X}_{t+1} = \mathbf{A}\mathbf{X}_t + \mathbf{B}\mathbf{u}_t$ . The motion state  $\mathbf{X}$  encompasses its position, velocity, attitudes, and angular velocities, while the control input  $\mathbf{u}$  includes the total thrust and torque provided by the engines. According to Mellinger and Kumar [23], multi-rotor UAVs' motion exhibits differential flatness. All aspects of the motion state and control inputs of a multi-rotor UAV can be expressed as algebraic functions of its flat outputs and their higher-order derivatives. The flat outputs of a multi-rotor UAV are its three-dimensional position and yaw angle. Consequently, the motion planning problem of a UAV can be simplified from a 12-dimensional state and 6-dimensional control input model into a 4-dimensional motion state planning model. This characteristic is also used in the UAV's trajectory tracking [40]. A three-dimensional motion for a UAV exhibiting differential flatness can be simplified and represented as a general model as Equation (10).

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ z_{t+1} \\ \dot{x}_{t+1} \\ \dot{y}_{t+1} \\ \dot{z}_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ z_t \\ \dot{x}_t \\ \dot{y}_t \\ \dot{z}_t \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 & 0 \\ 0 & \frac{1}{2}\Delta t^2 & 0 \\ 0 & 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \begin{bmatrix} \ddot{x}_t \\ \ddot{y}_t \\ \ddot{z}_t \end{bmatrix}. \quad (10)$$

For this model, the UAV’s accelerations in three directions are considered as the control inputs, and the optimization goal for the local trajectory generation is to minimize jerk, which represents the third derivative of position. For multi-rotor UAVs, according to [23], minimizing jerk also implies minimizing angular velocity, resulting in an optimization outcome that is a time-dependent trajectory well suited for onboard sensors, especially cameras.

Minimum jerk is represented as the smoothness cost function  $L = \ddot{\mathbf{x}}^2$ . According to the Euler–Lagrange equation, the optimized results exist if

$$\frac{\partial L}{\partial x} - \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{x}} \right) + \frac{d^2}{dt^2} \left( \frac{\partial L}{\partial \ddot{x}} \right) - \frac{d^3}{dt^3} \left( \frac{\partial L}{\partial \ddot{\mathbf{x}}} \right) + \dots + (-1)^n \frac{d^n}{dt^n} \left( \frac{\partial L}{\partial x^{(n)}} \right) = 0. \quad (11)$$

Substitute the cost function, and then

$$0 - \frac{d^3}{dt^3} \left( \frac{\partial \ddot{\mathbf{x}}^2}{\partial \ddot{\mathbf{x}}} \right) = 0 \rightarrow x^{(6)} = 0. \quad (12)$$

This indicates that a minimum jerk trajectory has a sixth-order derivative of zero, allowing the trajectory to be represented as a fifth-order polynomial:

$$x(t) = p_5t^5 + p_4t^4 + p_3t^3 + p_2t^2 + p_1t + p_0. \quad (13)$$

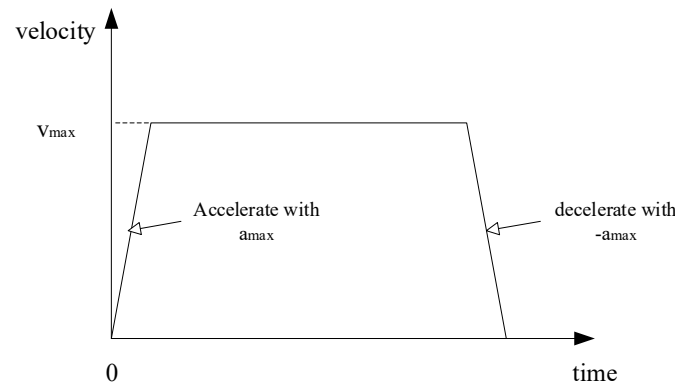
The trajectory planning problem can be solved by calculating the parameters of this polynomial. A polynomial trajectory provides a convenient means of expressing motion planning because it facilitates the derivation of velocity, acceleration, and other motion states through calculations of high-order derivatives of the polynomial [41]. Jerk can be calculated by  $x^{(3)}(t) = \sum_{i \geq 3} i(i-1)(i-2)t^{i-3}p_i$ . Then, the optimization objective is as follows:

$$\begin{aligned} J(T) &= \int_T L(t)dt = \int_T \sum_{i,l \geq 4} i(i-1)(i-2)l(l-1)(l-2)t^{i+l-6}p_i p_l dt \\ &= \sum_{i,l \geq 4} \frac{i(i-1)(i-2)l(l-1)(l-2)}{i+l-5} T^{i+l-5} p_i p_l \\ &= \begin{bmatrix} \vdots \\ p_i \\ \vdots \end{bmatrix}^T \begin{bmatrix} \ddots & & \vdots \\ \dots & \frac{i(i-1)(i-2)l(l-1)(l-2)}{i+l-5} T^{i+l-5} & \dots \\ \ddots & & \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ p_l \\ \vdots \end{bmatrix} \\ &= \mathbf{P}^T \mathbf{Q} \mathbf{P} \end{aligned} \quad (14)$$

The computation of a minimum jerk trajectory between two waypoints is modeled as a quadric programming problem, and the trajectory’s analytical solution can be determined using QP solvers. A trajectory that must pass through multiple waypoints can be viewed as a multi-segment trajectory, with its optimization objective outlined as the following:

$$\mathbf{J}_j(T) = \mathbf{P}_j^T \mathbf{Q}_j \mathbf{P}_j. \quad (15)$$

The initial time allocation for each segment of the trajectory is based on a simple assumption of motion between two waypoints as shown in Figure 6, where the UAV is assumed to perform a motion that it starts moving at 0 velocity and accelerates with the maximum acceleration until the velocity reaches its maximum, keeping the maximum velocity, and then deaccelerates to 0 velocity when it reaches the target waypoint.



**Figure 6.** The assumed motion used for the initial time allocation of the one-segment trajectory between two waypoints.

The standard expression of QP is the following:

$$\begin{aligned} \mathbf{x}^*(t) = \underset{\mathbf{x}(t)}{\operatorname{argmin}} (1/2)\mathbf{x}^T\mathbf{Q}\mathbf{x} \\ \text{subject to } \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \text{ and } \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{aligned} \quad (16)$$

The equation constraints of the QP problem originate from the positions of the waypoints and the continuity of motion states at each waypoint. This can be mathematically expressed as follows:

$$f_j(T_j) = x_j \Rightarrow \begin{bmatrix} \cdots & T_j^i & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ p_{j,i} \\ \vdots \end{bmatrix} = x_j, \quad (17)$$

$$\dot{f}_j(T_j) = \dot{f}_{j+1}(T_j) \Rightarrow \dot{f}_j(T_j) - \dot{f}_{j+1}(T_j) = 0 \Rightarrow \begin{bmatrix} \cdots & iT_j^{l-1} & \cdots & -lT_j^{l-1} & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ p_{j,i} \\ \vdots \\ p_{j+1,l} \\ \vdots \end{bmatrix} = 0, \quad (18)$$

$$\ddot{f}_j(T_j) = \ddot{f}_{j+1}(T_j) \Rightarrow \ddot{f}_j(T_j) - \ddot{f}_{j+1}(T_j) = 0 \Rightarrow \begin{bmatrix} \cdots & i(i-1)T_j^{l-2} & \cdots & -l(l-1)T_j^{l-2} & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ p_{j,i} \\ \vdots \\ p_{j+1,l} \\ \vdots \end{bmatrix} = 0. \quad (19)$$

The matrices  $\mathbf{A}_{eq}$  and  $\mathbf{b}_{eq}$  will be implemented by concluding Equations (17)–(19).

Through the waypoints adjustment process, the discrete waypoints within the local sensing range are supposed to be safe, while the safety of the trajectories between these waypoints still needs to be guaranteed. The inequation constraints in the QP problem will be set up according to the requirements of collision avoidance with the nearest local obstacle.

Since this paper mainly focuses on trajectory management, planning, and optimization, and the perception sensors of the UAV are not the main consideration, it is assumed that the onboard sensors will provide reliable perception information of the local environment. According to Srigrarom et al. [42], trajectories' characteristics of dynamic objects like UAVs and birds can be obtained, tracked, and predicted based on perception data. Fang et al. [43] used a vision-based perception system and a learning-based method to gain the precise and reliable perception information. In this paper, we assume that the fifth-order polynomial trajectory of dynamic obstacles and their sizes can be estimated and obtained by the onboard sensors. The inequation constraints are calculated by linearization of the following:

$$\sqrt{(x_{own}(t) - x_{obs}(t))^2 + (y_{own}(t) - y_{obs}(t))^2 + (z_{own}(t) - z_{obs}(t))^2} \geq radius_{own} + radius_{obs}, \quad (20)$$

where  $(x_{own}, y_{own}, z_{own})$  is the position trajectory of the UAV and  $(x_{obs}, y_{obs}, z_{obs})$  represents the position trajectory of the nearest obstacle. If the obstacle is static, its position  $(x_{obs}, y_{obs}, z_{obs})$  will be described as constant values. The condition is built by time sampling with a 1 s interval.

### 3. Simulation Results

MATLAB is used for algorithm simulation and the performance test. The navigation system was simulated using a GPS almanac file while the satellite signal mask was simulated according to the terrain map and satellite positions, and EPU was given by the simulated navigation system. The UAV is assumed to search an optimized flight path from the starting point (30, 30, 150) to the target point (1500, 1500, 600). The dynamic obstacles were set up as agents with determinate sizes, start positions, and target positions, while their motion trajectories were generated by the combination of a conventional A\* algorithm and closed-form QP without constraints. In the simulation tests, three moving obstacles starting from (1500, 1500, 450), (350, 30, 450), and (30, 1500, 500), separately, and targeting to (30, 30, 500), (350, 1500, 450), and (1500, 30, 450), were simulated. It is assumed that the onboard sensors would maintain reliable perception information and motion prediction of the local environment, including static and dynamic obstacles. The control system of the UAV is assumed to be able to follow and execute the planned trajectory.

#### 3.1. Simulation Test for Front-End Path Planner

##### 3.1.1. Verification of the Effects of Enhancement of Node Check

The effectiveness of the enhancement of the node check by introducing angle change constraints and expanding the one-step extension range is verified by simulation tests of the A\* algorithm with three regular heuristic distances. The angle constraints were set as 60 degrees in the horizontal direction and 45 degrees in the vertical direction. Three sets of experiments with different heuristic functions were operated. The results are concluded in Table 1.

**Table 1.** Simulation results of A\* and A\* + angle change constraints.

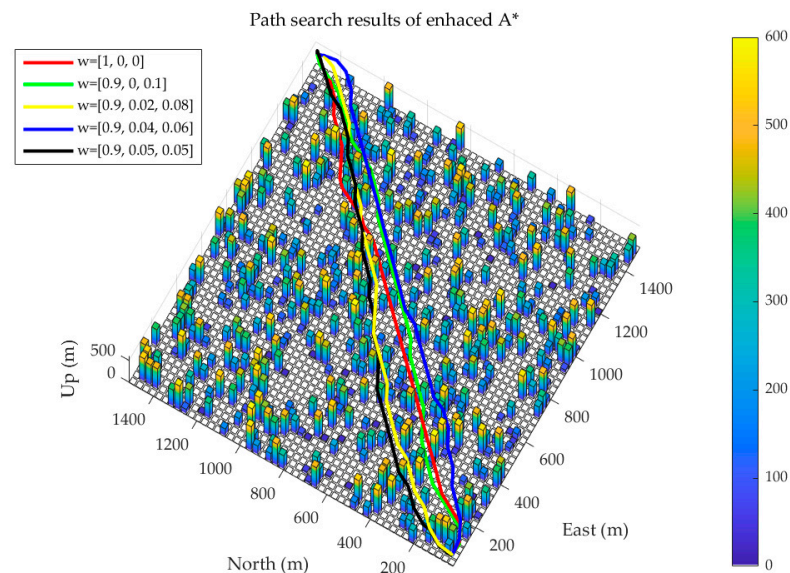
Heuristic Distance	Algorithm	Path Length (m)	Number of Expanded Nodes	Runtime (s)
Manhattan	A*	2237.793	56	0.07
	A* + angle constraints	2210.534	33	0.12
Euclidean	A*	2214.438	6538	42.28
	A* + angle constraints	2149.743	2607	10.65
Diagonal	A*	2214.438	1709	3.08
	A* + angle constraints	2162.328	504	1.10

All of the three types of heuristic functions successfully guided the A\* algorithm to locate the target node. By expanding the one-step extension range and implementing angle change constraints in the extension and node check rule of the A\* algorithm, there was a reduction in the number of expanded nodes. Consequently, the algorithm could

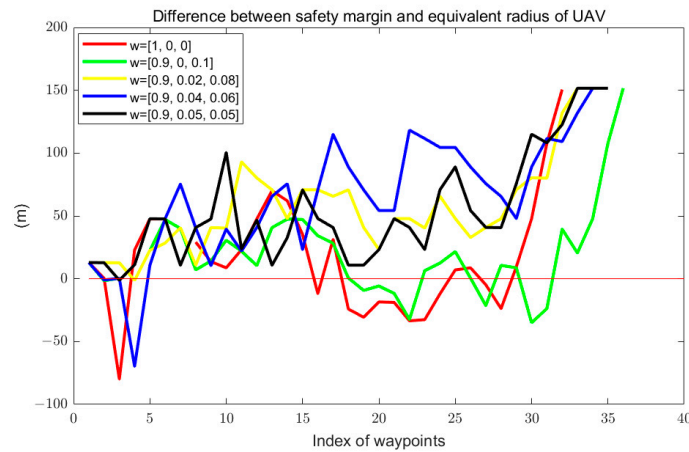
produce paths that were spatially smoother, sometimes with shorter path lengths. For the Manhattan heuristic, the number of expanded nodes and runtime were the smallest, but the path length was the longest. After the improvement, the number of expanded nodes decreased by 41.07%, and the runtime increased by 0.05 s due to the additional computations required for angle limitation checking. In the case of the Euclidean heuristic, the path length was the shortest, but it required expanding the largest number of nodes to reach the target, resulting in the longest runtime. However, after improvement, the number of expanded nodes decreased by 60.12% and the runtime decreased by over 30 s. Before altering the one-step extension range, A\* with the diagonal heuristic had the same optimality as A\* with the Euclidean heuristic, but after the improvement, the optimality decreased slightly. The diagonal heuristic also saw a 70.51% reduction in the number of expanded nodes and a 74.35% reduction in the runtime. These experimental results confirm the effectiveness of introducing angle limitations to the A\* algorithm, with the diagonal distance heuristic function striking a balance between the optimality and the number of expanded nodes. In the subsequent sections of this paper, the heuristic function will be further refined based on the diagonal distance.

### 3.1.2. Simulation Test for the Enhanced A\* Algorithm

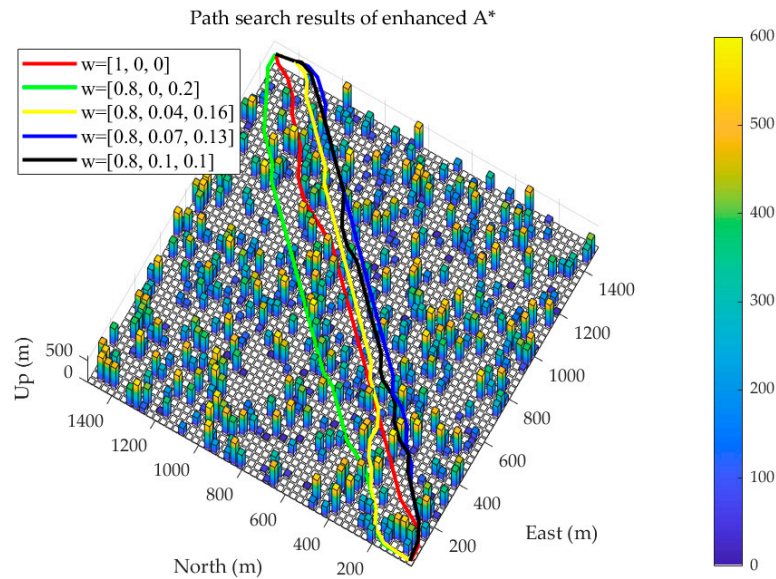
Upon altering the heuristic function as expressed in Equation (7), the safety of the planned paths should be enhanced, since the multiple tests were conducted using different weight ratios to determine the most suitable configuration. One set of simulation results is presented in the following figures, where four weight ratio combinations were tested alongside a diagonal heuristic without safety factors for comparison. The results also provide overviews of the safety evaluation results, equal to the difference between distances from the nearest occupied grid cell and the equivalent radius of the UAV relevant to the estimated navigation uncertainty for each path. Figures 7 and 8 represent the simulation results when  $w_1 = 0.9$ . Figures 9 and 10 represent the simulation results when  $w_1 = 0.8$ , and Figures 11 and 12 represent the simulation results when  $w_1 = 0.7$ . Key test results are summarized in Table 2.



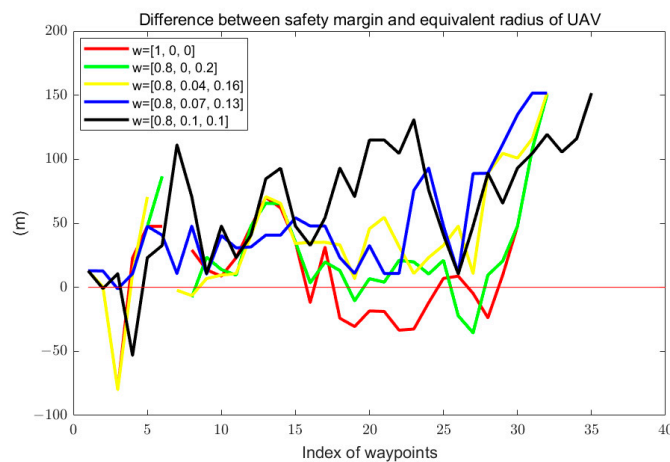
**Figure 7.** Path search results of the enhanced A\* algorithm with  $w_1 = 0.9$ .



**Figure 8.** Safety evaluation results of the paths with  $w_1 = 0.9$ . The safety of the paths is evaluated by the difference between the safety margin (distance to the nearest obstacle) and the equivalent radius of the UAV for each node.



**Figure 9.** Path search results of the enhanced A\* algorithm with  $w_1 = 0.8$ .



**Figure 10.** Safety evaluation results of the paths with  $w_1 = 0.8$ . The safety of the paths is evaluated by the difference between the safety margin (distance to the nearest obstacle) and the equivalent radius of the UAV for each node.

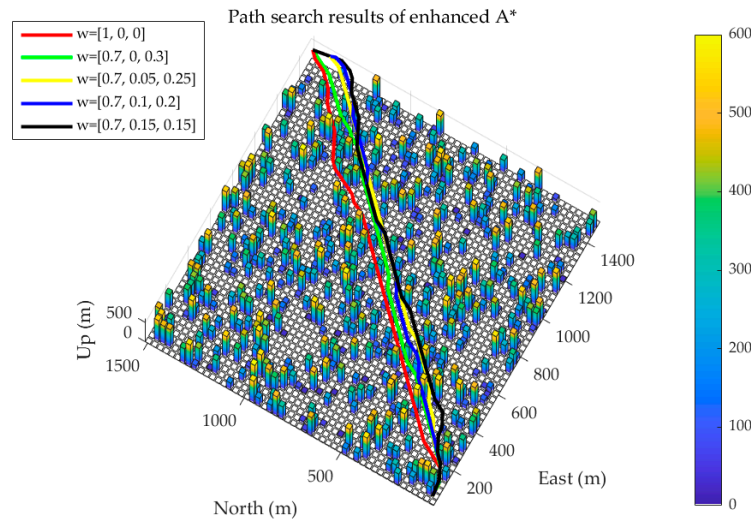


Figure 11. Path search results of the enhanced A\* algorithm with  $w_1 = 0.7$ .

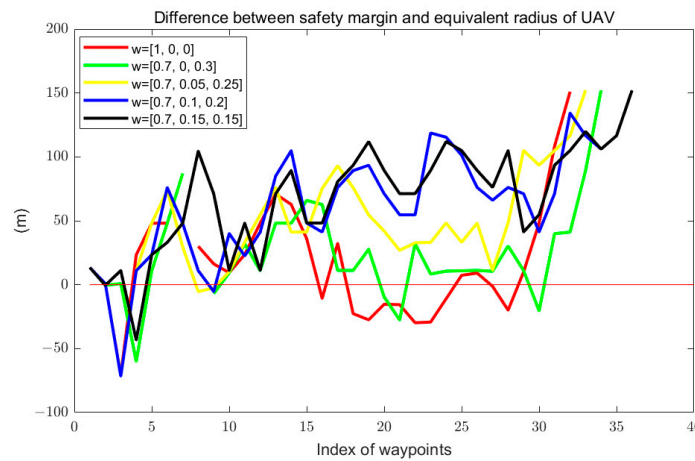


Figure 12. Safety evaluation results of the paths with  $w_1 = 0.7$ . The safety of the paths is evaluated by the difference between the safety margin (distance to the nearest obstacle) and the equivalent radius of the UAV for each node.

Table 2. Key simulation results of the enhanced A\* algorithm.

Weight Ratio			Number of Expanded Nodes	Runtime (s)	Path Length (m)	GPS Denial Rate (%)	Avg. Safety Margin (m)
$w_1$	$w_2$	$w_3$					
1.00	0.00	0.00	504	3.52	2161.995	3.125	15.9263
0.90	0.00	0.10	6079	137.25	2206.698	2.778	19.7855
0.90	0.02	0.08	4998	84.41	2359.587	0	52.7426
0.90	0.04	0.06	4237	53.87	2409.132	0	63.4888
0.90	0.05	0.05	4493	46.91	2403.651	0	57.3843
0.80	0.00	0.20	12,338	457.12	2174.499	3.125	23.1908
0.80	0.04	0.16	8670	144.60	2316.483	6.061	37.8717
0.80	0.07	0.13	6856	100.19	2412.645	0	49.0596
0.80	0.10	0.10	5667	78.22	2474.637	0	65.4700
0.70	0.00	0.30	18,982	1023.76	2388.690	3.125	44.1877
0.70	0.05	0.25	11,019	286.17	2688.516	3.125	44.1543
0.70	0.10	0.20	8413	195.77	2769.345	0	47.5405
0.70	0.15	0.15	5892	113.70	2747.397	0	60.5496

GPS denial is defined as the condition when the number of visible satellites is less than four and the GPS receiver is not able to calculate the positioning results. During

the simulation experiments,  $w_2$  was set not greater than  $w_3$  in order not to overlook the effects of obstacle density from the navigation performance estimates. As in Table 2, lower  $w_1$  and higher  $w_2$  and  $w_3$  indicate the reduced importance of the direction guide to the target position and the heightened importance of safety and navigation performance consideration. This, in turn, results in more expanded nodes, longer runtime, and longer path lengths. Since the navigation performance cost works as a soft constraint in the algorithm, there are some GPS denial conditions in the test scenarios. A higher weight  $w_2$  causes the path to favor areas with low obstacle densities, while  $w_3$  helps ensure navigation performance and safe distances from obstacles. For flight safety, conditions of the equivalent radius of the UAV being much larger than the safety margin, and GPS denial, must be avoided. Taking into account the number of expanded nodes, runtime, safety improvement, and path lengths, a weight combination of 0.9/0.05/0.05 is proved to be a reasonable choice. This combination strikes a balanced approach, improving the average safety margin by 260.31% while increasing the path length by just 11.17% compared to the 1/0/0 weight ratio.

### 3.2. Simulation Test for Back-End Trajectory Optimizer

The simulation of the back-end trajectory optimizer includes the local environment update based on the current position of the UAV, adjustment for planning waypoints, trajectory time allocation, trajectory generation by quadratic programming, safety check, and motion feasibility check for the planning trajectory. The global view of the trajectory and velocity, acceleration, bank angle, and navigation performance data at different times are concluded in Section 3.2.1, and the detailed local view of UAV maneuvers when encountering different situations are concluded in Section 3.2.2.

#### 3.2.1. Trajectory Generated by the Fusing Approach

The objectives of trajectory optimization are to enhance smoothness and safety while satisfying the continuity of motion states and control inputs, as well as kinematic and dynamic motion feasibility. In the simulation tests, the UAV's motion is set with a max bank angle limit of  $35^\circ$ , vertical velocity constraints within  $-6$  to  $8$  m/s, and horizontal speed limits below  $21$  m/s. Based on force analysis, the maximum horizontal acceleration is  $6.8690$  m/s<sup>2</sup>, while the vertical acceleration should not exceed  $2.1658$  m/s<sup>2</sup>. To ensure kinematic and dynamic motion feasibility, the trajectory should satisfy the above limitations.

The following figures represented in Figure 13 present the simulation results of the trajectory sampled every 50 s at a global view, including the global path, executing trajectory, planning trajectory, passed trajectory, and positions of moving obstacles at certain times.

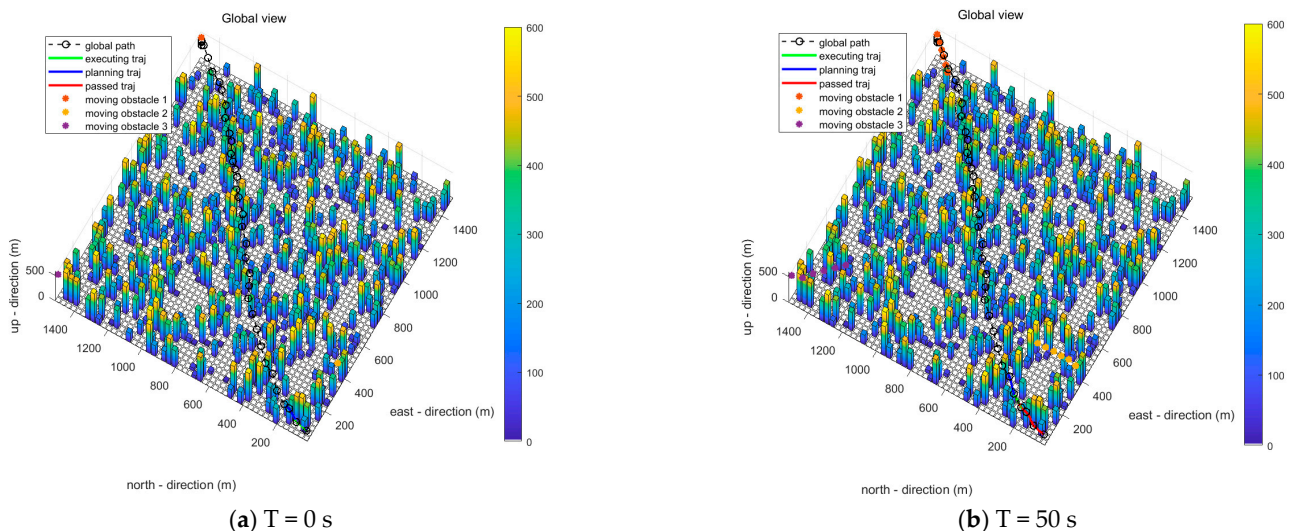
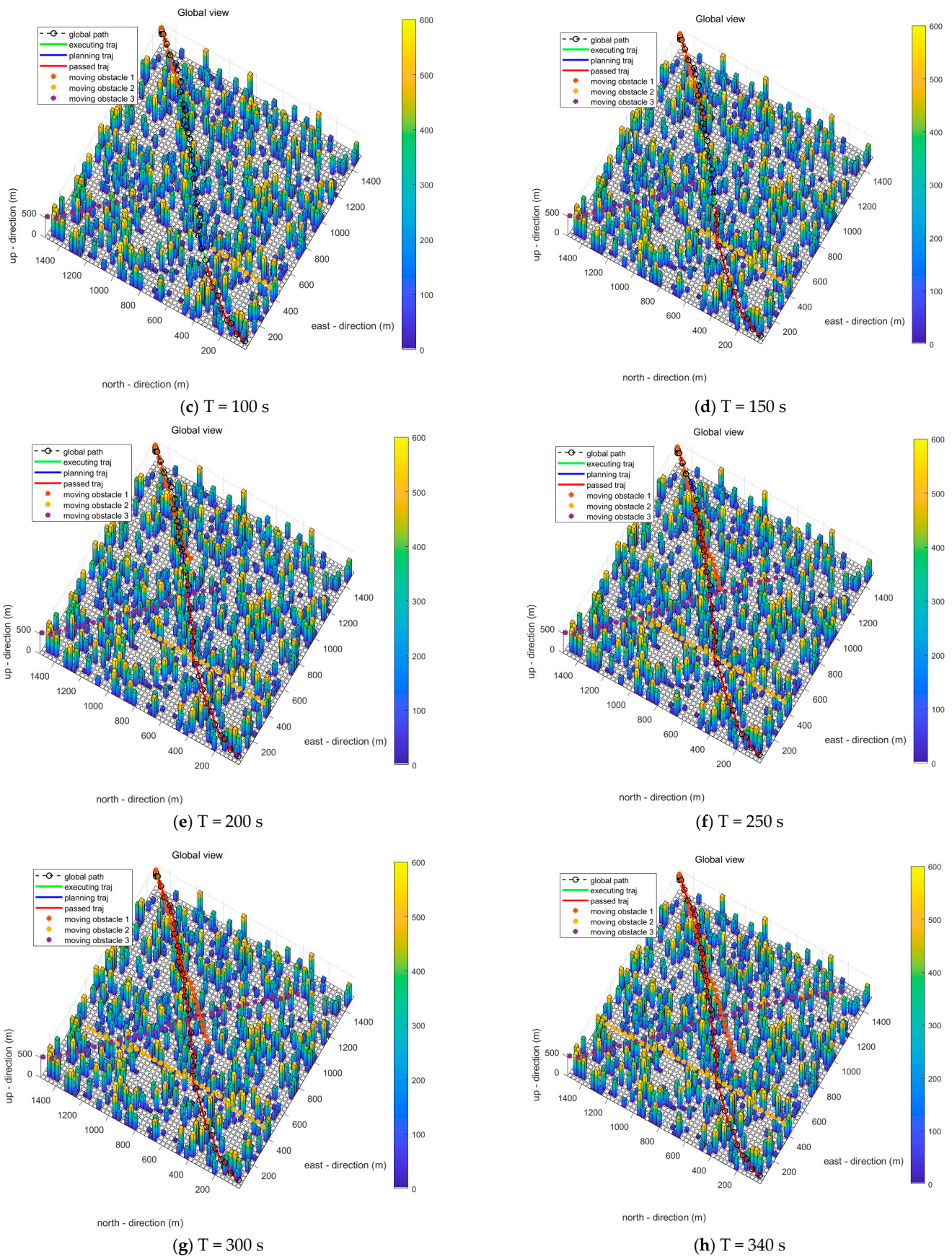


Figure 13. Cont.



**Figure 13.** Global views of trajectory results from the AFTM and DM system from the start time  $T = 0$  s to the end time  $T = 340$  s.

Figure 13 presents a trajectory of the UAV for the simulation scenario using the AFTM and DM system designed in this paper. Figure 13a–h present trajectories of the UAV and moving obstacles at different times. The figures of the global views provide an intuitive scope of the simulation environment, where the UAV encountered safety threats of both static and dynamic obstacles in the local environment. Figures 14–17 conclude the position, velocity, acceleration, and bank angle data of the three-dimensional trajectory over time.

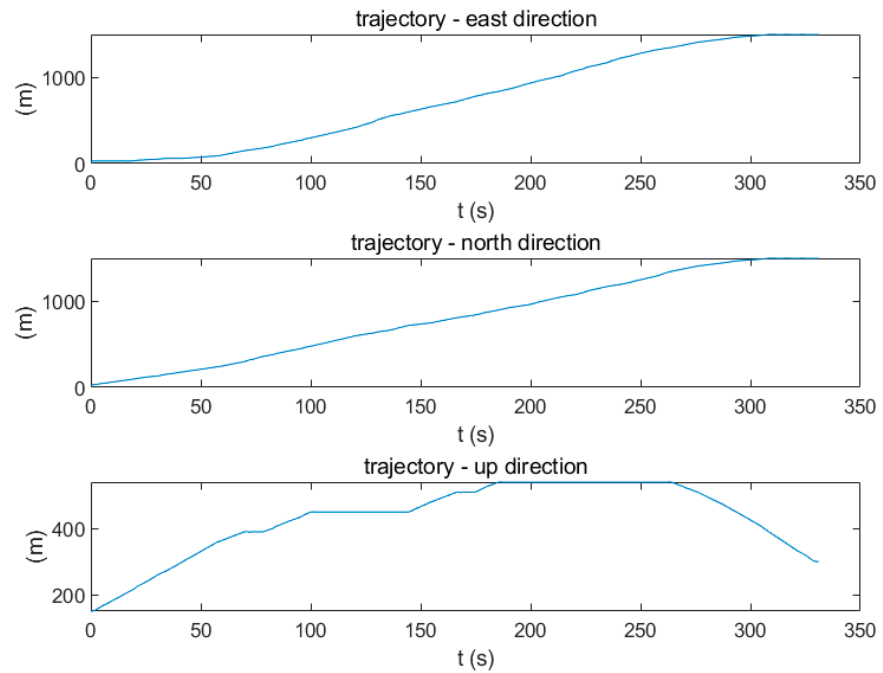


Figure 14. The 3D trajectory vs. time.

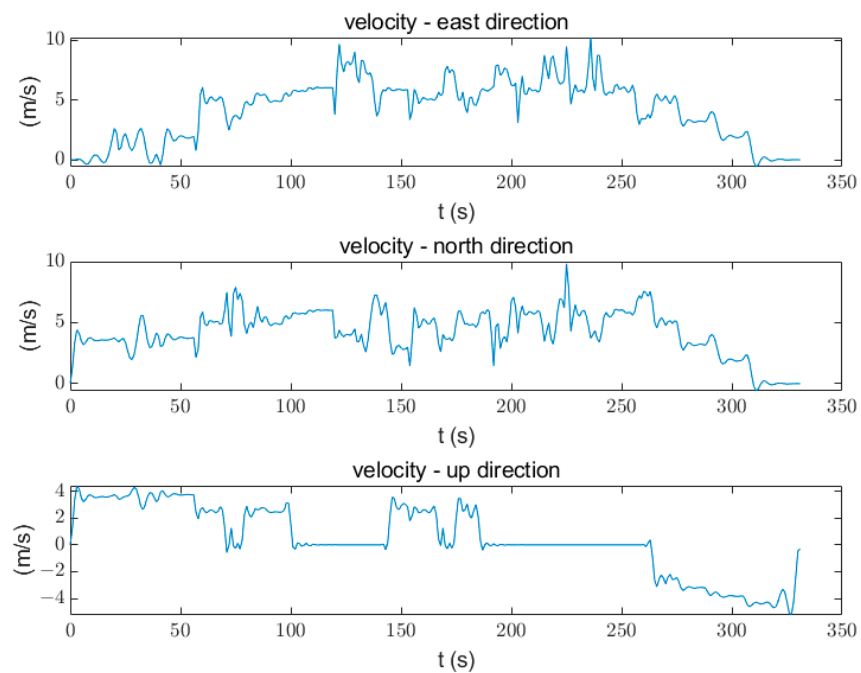
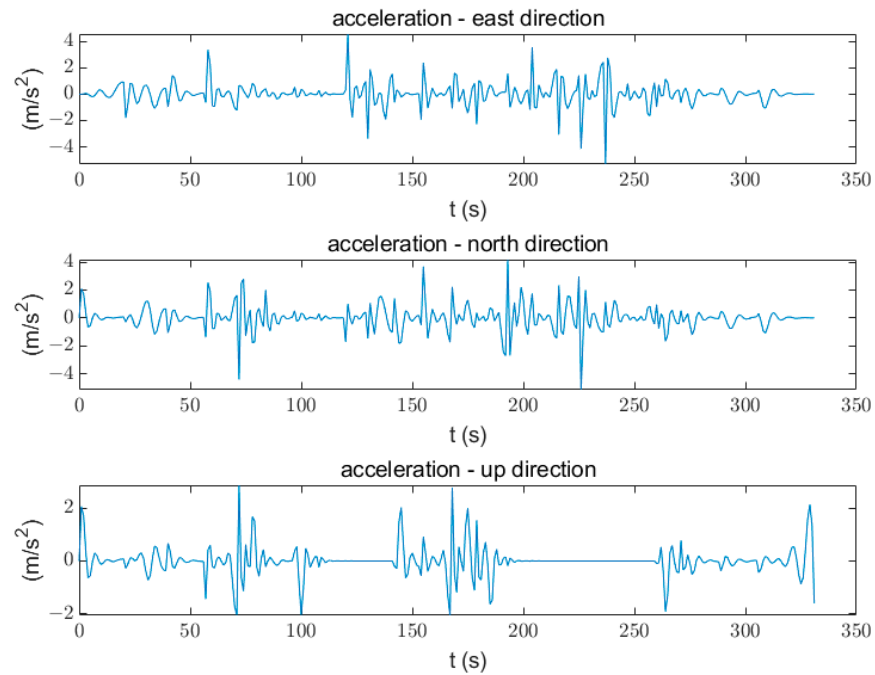
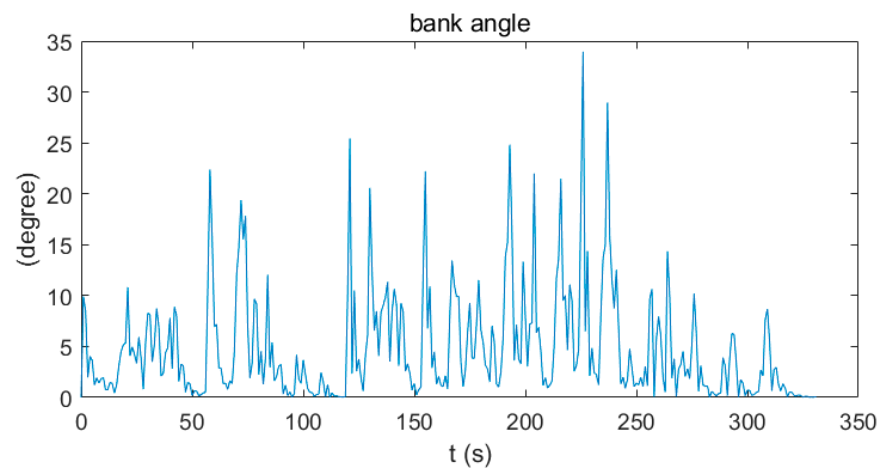


Figure 15. Velocity of the trajectory.



**Figure 16.** Acceleration of the trajectory.



**Figure 17.** Bank angle of the trajectory.

Figures 15–17 present the continuity and smoothness of the velocity, acceleration, and bank angle of the trajectory generated by the AFTM and DM system. The velocity, acceleration, and bank angle during the flight mission are limited within the predefined constraints. The bank angle constraints result in the UAV performing a smooth attitude change and no aggressive maneuvers.

### 3.2.2. Local View of Collision Avoidance Maneuver

This section determines key instances of how the waypoints adjustment and collision avoidance strategy for the local trajectory optimizer works. Figure 18 presents the normal instances when local waypoints do not need to be displaced and the safety requirements of the trajectory can be satisfied when following the global path.

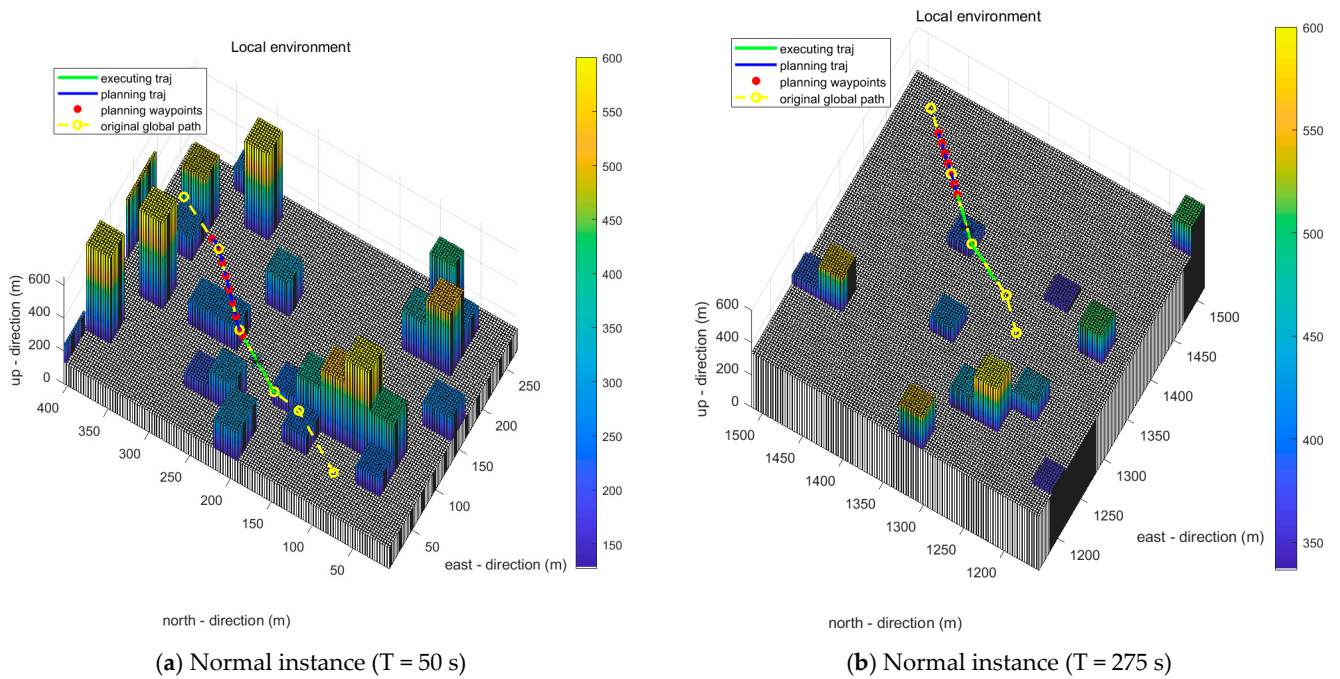


Figure 18. Normal instances of the trajectory following the global path without collision threats.

Figure 19 presents an example of how the waypoints adjustment and collision avoidance strategy for the local trajectory optimizer works when the UAV was planning a trajectory entering a narrow passage at  $T = 65$  s. In the figures, the original global path waypoints did keep safe distances away from the obstacle. However, the waypoints would lead the UAV to places close to the pillar obstacle. The local planning waypoints were adjusted and displaced by safe distances away from the obstacle and, therefore, the local optimized trajectory could be collision-free.

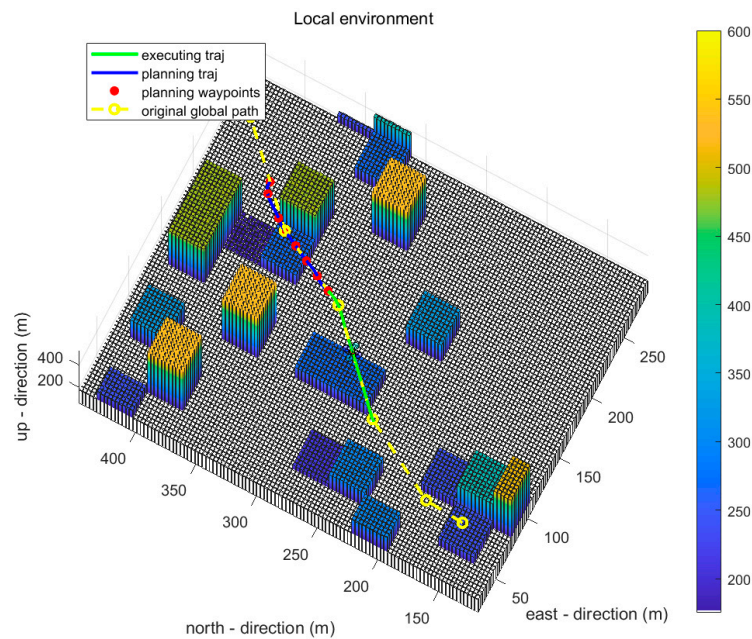
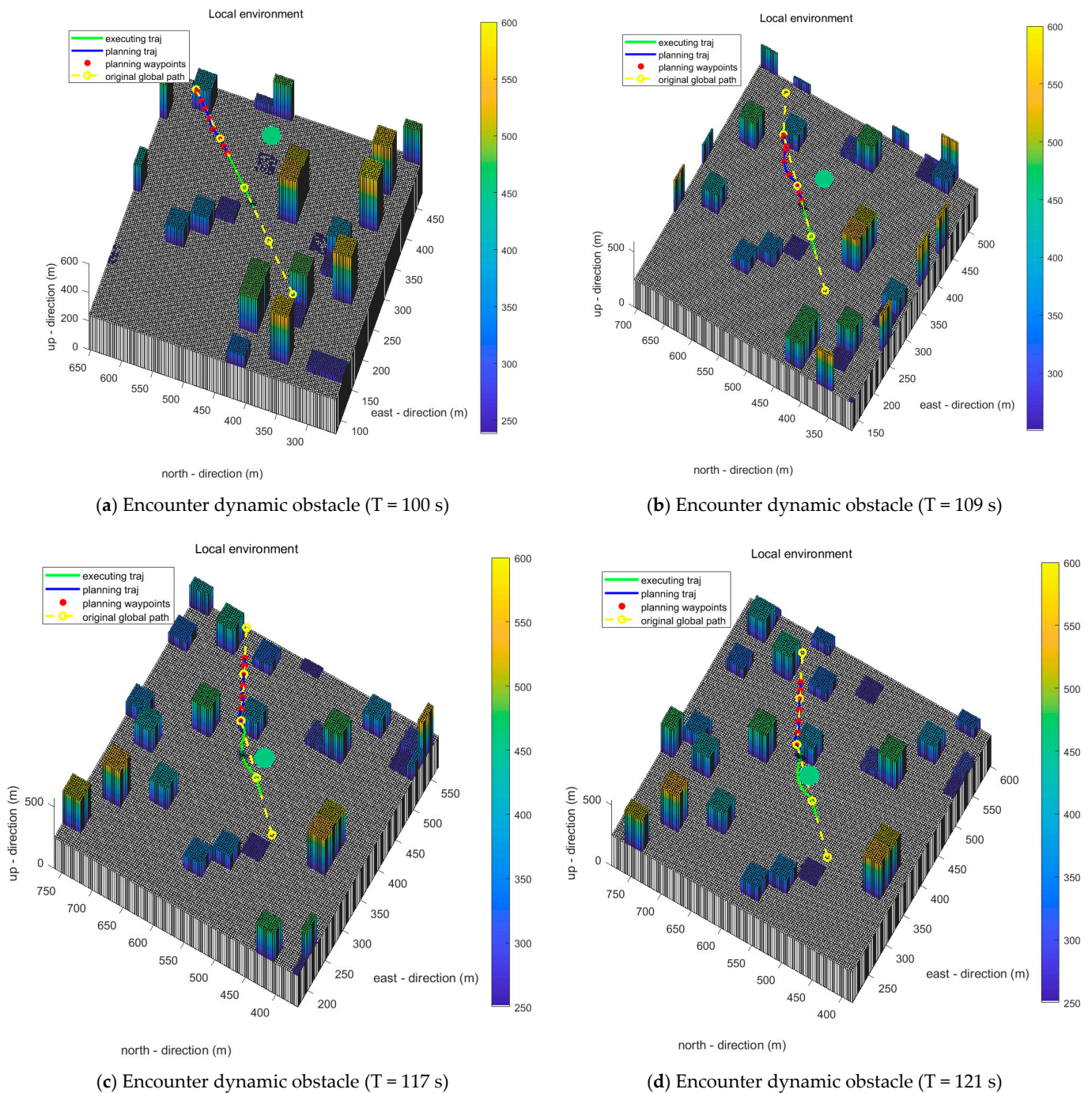


Figure 19. A typical case when the UAV is entering a narrow passage.

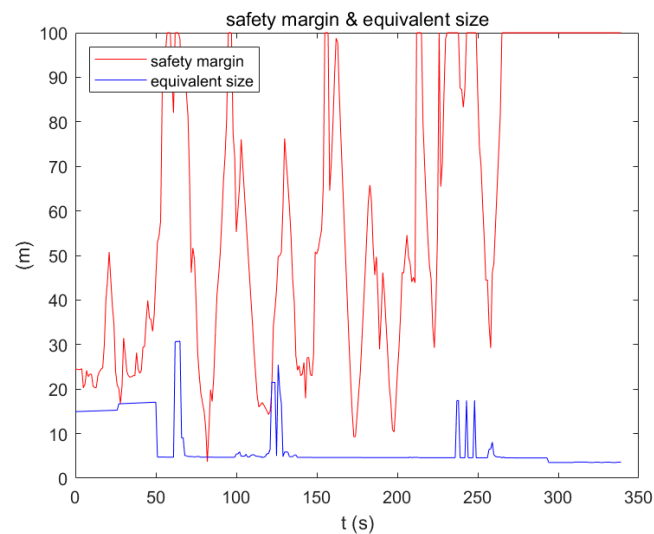
Figure 20 presents the case when the UAV encountered a collision threat with a dynamic obstacle if it followed the originally planned global path. Figure 20a,b present the adjustment of the local planning trajectory according to the collision avoidance demand,

and Figure 20c,d present the execution of the trajectory which did successfully avoid collision with the dynamic obstacle.



**Figure 20.** A typical case when the UAV encountered a dynamic obstacle (presented as a green sphere in the figure).

Figure 21 presents the safety margin and equivalent radius of the UAV of the trajectory, while the safety margin is obtained by the distance between the UAV and the real-time nearest obstacle with consideration of both static and dynamic obstacles. The equivalent size is defined as the real-time equivalent radius of size of the UAV calculated by the sum of its radius expanded by real-time positioning uncertainty. The figure proves that the trajectory generated by the AFTM and DM system in this paper able to help the UAV keep sufficient positioning performances and safe distances from obstacles.



**Figure 21.** The safety margin and equivalent size of the UAV of the trajectory (distances greater than 100 m are ignored).

#### 4. Conclusions

This paper presents a comprehensive UAV AFTM and DM method that encompasses the fusing of static global path planning based on an enhanced A\* algorithm and dynamic local trajectory optimization utilizing local waypoints adjustment and the implementation of quadratic programming. This method is designed to facilitate UAV operations in complex airspace environments where the UAV would encounter high risks of collision and navigation performance decline. The enhanced A\* algorithm considers factors such as UAV angle change constraints, positioning uncertainty, and safety during the global path planning phase, with simulation results affirming its effectiveness. Furthermore, the local trajectory optimization ensures the feasibility and smoothness of the planned trajectories. Simulations of real-time positioning uncertainty and maximum safe distances prove the effectiveness of the collision avoidance. The fusing method based on receding-horizon control successfully balances the execution of the planned trajectory and computation for the adjustment of the planning trajectory.

One of the limitations of this research is the lack of field tests with UAVs, and the experiments were operated as simulations realized by MATLAB. A complete UAV system, including perception system, control system, etc. was not implemented since the research is mainly focusing on the trajectory management and decision-making system. The environmental simulation lacks some consideration. For example, effects of wind were not considered in this research but can be important in practical applications. In addition, the dynamic obstacles were not modeled based on specific motion characteristics of birds, certain types of UAVs, or balloons, etc. Further studies may focus on the enhancement of modeling a windy environment according to [44,45]. Although the effects of wind may be overcome by a well-designed control system, dealing with them during the planning phase, according to [46], will increase the reliability of the UAV system. In addition, the weights determination of the cost function of the enhanced A\* algorithm and time allocation for local trajectories may be further enhanced by using other methods or algorithms. However, the simulation results did generate the positive results of this research. The study on motion planning and trajectory management based on positioning uncertainty and navigation performance is worth the effort. The combination of smooth trajectory generation and collision avoidance realized by quadratic programming is proved to be effective. Future research efforts should prioritize field tests or environmental simulation with the consideration of wind and other details in practical applications, the optimization of trajectory time allocation, and other enhancements associated with positioning performance management with both global strategical path planning and local tactical trajectory optimization.

**Author Contributions:** Conceptualization, S.S. and H.W.; methodology, H.W. and Y.X.; software, H.W. and T.W.; validation, S.S., H.W., R.L. and W.C.; formal analysis, H.W.; investigation, S.S. and H.W.; resources, S.S.; data curation, S.S.; writing—original draft preparation, H.W.; writing—review and editing, S.S. and H.W.; visualization, H.W.; supervision, S.S.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported in part by the State Key Program of the National Natural Science Foundation of China (U2233215).

**Data Availability Statement:** All the data are included in the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. National Academies of Sciences, Engineering, and Medicine. *Advancing Aerial Mobility: A National Blueprint*, 1st ed.; The National Academies Press: Washington, DC, USA, 2020; ISBN 978-0-309-67026-5. [\[CrossRef\]](#)
2. Karr, D.A.; Wing, D.J.; Barney, T.L.; Sharma, V.; Etherington, T.J.; Sturdy, J.L. Initial Design Guidelines for Onboard Automation of Flight Path Management. In Proceedings of the AIAA AVIATION 2021 Forum, Online, 2–6 August 2021.
3. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [\[CrossRef\]](#)
4. Gammell, J.D.; Barfoot, T.D.; Srinivasa, S.S. Informed sampling for asymptotically optimal path planning. *IEEE Trans. Robot.* **2018**, *34*, 966–984. [\[CrossRef\]](#)
5. Alam, M.M.; Nishi, T.; Liu, Z.; Fujiwara, T. A novel sampling-based optimal motion planning algorithm for energy-efficient robotic pick and place. *Energies* **2023**, *16*, 6910. [\[CrossRef\]](#)
6. Chen, L.; Shan, Y.; Tian, W.; Li, B.; Cao, D. A fast and efficient double-tree RRT\*-like sampling-based planner applying on mobile robotic systems. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 2568–2578. [\[CrossRef\]](#)
7. Gao, M.; Yan, T.; Fu, W.; Feng, Z.; Guo, H. Automated flight technology for integral path planning and trajectory tracking of the UAV. *Drones* **2024**, *8*, 9. [\[CrossRef\]](#)
8. Gao, F.; Wu, W.; Gao, W.; Shen, S. Flying on point clouds: Online trajectory generation and autonomous navigation for quadrotors in cluttered environments. *J. Field Robot.* **2019**, *36*, 710–733. [\[CrossRef\]](#)
9. Candra, A.; Budiman, M.A.; Hartanto, K. Dijkstra’s and A-star in finding the shortest path: A tutorial. In Proceedings of the 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA), Medan, Indonesia, 16–17 July 2020.
10. Dechter, R.; Pearl, J. Generalized best-first search strategies and the optimality of A\*. *J. ACM* **1985**, *32*, 505–536. [\[CrossRef\]](#)
11. Likhachev, M.; Gordon, G.J.; Thrun, S. ARA\*: Anytime A\* with provable bounds on sub-optimality. In Proceedings of the Conference and Workshop on Neural Information Processing Systems (NIPS), Vancouver and Whistler, BC, Canada, 8–13 December 2003.
12. Harabor, D.; Grastien, A. Online graph pruning for pathfinding on grid maps. In Proceedings of the AAAI Conference on Artificial Intelligence 2011, San Francisco, CA, USA, 7–11 August 2011.
13. Chen, P.; Jiang, Y.; Dang, Y.; Yu, T.; Liang, R. Real-time efficient trajectory planning for quadrotor based on hard constraints. *J. Intell. Robot. Syst.* **2022**, *105*, 52. [\[CrossRef\]](#)
14. Szczerba, R.J.; Galkowski, P.; Glicktein, I.S.; Ternullo, N. Robust algorithm for real-time route planning. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 869–878. [\[CrossRef\]](#)
15. Liu, S.; Watterson, M.; Mohta, K.; Sun, K.; Bhattacharya, S.; Taylor, C.J.; Kumar, V. Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1688–1695. [\[CrossRef\]](#)
16. Zhang, X.; Duan, H. An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning. *Appl. Soft. Comput.* **2015**, *26*, 270–284. [\[CrossRef\]](#)
17. Cao, Y.; Wei, W.; Bai, Y.; Qiao, H. Multi-base multi-UAV cooperative reconnaissance path planning with genetic algorithm. *Clust. Comput.* **2019**, *22*, 5175–5184. [\[CrossRef\]](#)
18. Ma, Y.; Zhang, H.; Zhang, Y.; Gao, R.; Xu, Z.; Yang, J. Coordinated optimization algorithm combining GA with cluster for multi-UAVs to multi-tasks task assignment and path planning. In Proceedings of the 2019 IEEE 15th International Conference on Control and Automation (ICCA), Edinburgh, UK, 16–19 July 2019.
19. Park, C.; Kim, G.S.; Park, S.; Jung, S.; Kim, J. Multi-agent reinforcement learning for cooperative air transportation services in city-wide autonomous urban air mobility. *IEEE Trans. Intell. Veh.* **2023**, *8*, 4016–4030. [\[CrossRef\]](#)
20. Pivtoraiko, M.; Kelly, A. Generating near minimal spanning control sets for constrained motion planning in discrete state spaces. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005. [\[CrossRef\]](#)
21. Mueller, M.W.; Hehn, M.; D’Andrea, R. A computationally efficient motion primitive for quadcopter trajectory generation. *IEEE Trans. Robot.* **2015**, *31*, 1294–1310. [\[CrossRef\]](#)

22. Liu, S.; Atanasov, N.; Mohta, K.; Kumar, V. Search-based motion planning for quadrotors using linear quadratic minimum time control. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017. [\[CrossRef\]](#)
23. Mellinger, D.; Kumar, V. Minimum snap trajectory generation and control for quadrotors. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011. [\[CrossRef\]](#)
24. Wang, Z.; Zhou, X.; Xu, C.; Gao, F. Geometrically constrained trajectory optimization for multicopters. *IEEE Trans. Robot.* **2022**, *38*, 3259–3278. [\[CrossRef\]](#)
25. Shen, Z.; Zhou, G.; Huang, H.; Huang, C.; Wang, Y.; Wang, F.-Y. Convex optimization-based trajectory planning for quadrotors landing on aerial vehicle carriers. *IEEE Trans. Intell. Veh.* **2023**, *9*, 138–150. [\[CrossRef\]](#)
26. Blasi, L.; D’Amato, E.; Notaro, I.; Raspaolo, G. Clothoid-Based Path Planning for a Formation of Fixed-Wing UAVs. *Electronics* **2023**, *12*, 2204. [\[CrossRef\]](#)
27. Ippolito, C.; Hening, S.; Sankararaman, S.; Stepanyan, V. A modeling, simulation and control framework for small unmanned multicopter platforms in urban environments. In Proceedings of the 2018 AIAA Modeling and Simulation Technologies Conference, Kissimmee, FL, USA, 8–12 January 2018. [\[CrossRef\]](#)
28. Baculi, J.E.; Ippolito, C.A. Onboard decision-making for nominal and contingency sUAS flight. In Proceedings of the AIAA Scitech 2019 Forum, San Diego, CA, USA, 7–11 January 2019. [\[CrossRef\]](#)
29. Luo, X.; Zhang, T.; Xu, W.; Fang, C.; Lu, T.; Zhou, J. Multi-tier 3D trajectory planning for cellular-connected UAVs in complex urban environments. *Symmetry* **2023**, *15*, 1628. [\[CrossRef\]](#)
30. Ma, Z.; Wang, Z.; Ma, A.; Liu, Y.; Niu, Y. A low-altitude obstacle avoidance method for UAVs based on polyhedral flight corridor. *Drones* **2023**, *7*, 588. [\[CrossRef\]](#)
31. Shen, S. Autonomous Navigation in Complex Indoor and Outdoor Environments with Micro Aerial Vehicles. Ph.D. Dissertation, University of Pennsylvania, Philadelphia, PA, USA, 2014.
32. Zhou, B.; Gao, F.; Wang, L.; Liu, C.; Shen, S. Robust and efficient quadrotor trajectory generation for fast autonomous flight. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3529–3536. [\[CrossRef\]](#)
33. Zhou, X.; Wang, Z.; Ye, H.; Xu, C.; Gao, F. EGO-Planner: An ESDF-free gradient-based local planner for quadrotors. *IEEE Robot. Autom. Lett.* **2021**, *6*, 478–485. [\[CrossRef\]](#)
34. Kwon, W.H.; Han, S. Receding horizon schemes for controls, estimations, and optimizations. In Proceedings of the 2007 International Conference on Control, Automation and Systems, Seoul, Republic of Korea, 17–20 October 2017. [\[CrossRef\]](#)
35. Moon, M.Y.; Lee, J.Y.; Park, J.B.; Choi, Y.H. Action-dependent updated terminal cost receding horizon control for discrete-time linear systems. In Proceedings of the 2012 12th International Conference on Control, Automation and Systems, Jeju, Republic of Korea, 17–21 October 2012.
36. ICAO. *Performance-Based Navigation (PBN) Manual*; International Civil Aviation Organization: Montreal, QC, Canada, 2008.
37. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [\[CrossRef\]](#)
38. Zhang, N.; Zhang, Y.; Ma, C.; Wang, B. Path planning of six-DOF serial robots based on improved artificial potential field method. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, China, 5–8 December 2017. [\[CrossRef\]](#)
39. Manfredi, G.; Jestin, Y. An introduction to ACAS Xu and the challenges ahead. In Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, CA, USA, 25–29 September 2016. [\[CrossRef\]](#)
40. Freire, V.; Xu, X. Flatness-based quadcopter trajectory planning and tracking with continuous-time safety guarantees. *IEEE Trans. Control. Syst. Technol.* **2023**, *31*, 2319–2334. [\[CrossRef\]](#)
41. Mellinger, D.; Michael, N.; Kumar, V. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *Int. J. Robot. Res.* **2012**, *31*, 664–674. [\[CrossRef\]](#)
42. Srigrarom, S.; Chew, K.H.; Lee, D.M.D.; Ratsamee, P. Drone versus Bird Flights: Classification by Trajectories Characterization. In Proceedings of the 2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Chiang Mai, Thailand, 23–26 September 2020. [\[CrossRef\]](#)
43. Fang, G.; Wang, X.; Wang, K.; Lee, K.-H.; Ho, J.D.L.; Fu, H.-C.; Fu, D.K.C.; Kwok, K.-W. Vision-Based Online Learning Kinematic Control for Soft Robots Using Local Gaussian Process Regression. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1194–1201. [\[CrossRef\]](#)
44. Phadke, A.; Medrano, F.A.; Chu, T.; Sekharan, C.N.; Starek, M.J. Modeling Wind and Obstacle Disturbances for Effective Performance Observations and Analysis of Resilience in UAV Swarms. *Aerospace* **2024**, *11*, 237. [\[CrossRef\]](#)
45. Wang, B.H.; Wang, D.B.; Ali, Z.A.; Ting, T.B.; Wang, H. An overview of various kinds of wind effects on unmanned aerial vehicle. *Meas. Control.* **2019**, *52*, 731–739. [\[CrossRef\]](#)
46. Jayaweera, H.M.P.C.; Hanoun, S. Path Planning of Unmanned Aerial Vehicles (UAVs) in Windy Environments. *Drones* **2022**, *6*, 101. [\[CrossRef\]](#)

**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.