Quadcopter UAV flight trajectory optimization using improved A* algorithm with adaptive heuristic strategy

Xuan Vinh Bui[1,2], Xuan Minh Dinh[3], Xuan Hai Le[4], Ngoc Thanh Ta[3], Cao Phong Khong[2], Huu Nguyen Thai[1*]

[1]Faculty of Electrical and Electronics Engineering, Vinh University of Technical Education, Vinh, Vietnam
[2]Hanoi University of Mining and Geology, Hanoi, Vietnam
[3]HTI UAS Joint Stock Company, Hanoi, Vietnam
[4]International School, Vietnam National University, Hanoi, Vietnam
[*]Corresponding author E-mail: thainguyenktv@gmail.com
DOI: https://doi.org/10.64032/mca.v30i3.413

Abstract

This study proposes a global trajectory planning solution for Quadcopter UAVs to optimize execution time, the number of nodes to be expanded, and ensure mission feasibility and safety. Recognizing the computational performance limitations of traditional algorithms such as Dijkstra, the study proposes an improved method based on the A* algorithm combined with an adaptive heuristic strategy. The core of the method is a flexible conversion mechanism between Manhattan and Euclidean standards based on the actual distance threshold to the target, minimizing the search space and increasing execution speed. To ensure physical feasibility, we integrate a safety boundary mechanism to minimize collision risk and Cubic Spline interpolation techniques to smooth discrete trajectories, ensuring continuity in velocity and acceleration. Simulation scenarios performed on MATLAB R2023b with medium to high obstacle densities have demonstrated the superior effectiveness of the proposed method. Quantitative results show that the proposed algorithm minimizes the number of expansion-required nodes compared to A* and Dijkstra, while significantly improving execution time to meet stringent real-time requirements, creating a solid foundation for deployment on embedded UAV systems in complex 3D environments.

Keywords: *UAV quadcopter; Global flight path optimization; Cubic spline adaptive heuristic strategy; A* algorithm.*

Abbreviations

UAVs	Unmanned Aerial Vehicles
VTOL	Vertical Take-Off and Landing
GPP	Global Path Planning
LPP	Local Path Planning
GCS	Ground Control Station
3D	Three-dimensional
GA	Genetic Algorithm
GWO	Gray Wolf Optimization
FPA	Flower Pollination Algorithm
PSO	Particle Swarm Optimization
FL	Fuzzy Logic
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
ROS	Robot Operating Systems

1. Introduction

Currently, UAVs play a key role in both civilian and defense [1]. Among them, quadcopter UAVs have attracted much research interest due to their compact structure, reasonable operating costs, and flexible VTOL capabilities in confined spaces [2]. However, due to the characteristic of consuming a large amount of energy to maintain lift, the trajectory planning problem has become an urgent requirement to optimize flight paths and save energy for quadcopters, thereby improving the overall efficiency of the mission.

Normally, this problem is divided into two main types: static trajectory planning and dynamic trajectory planning, also known as GPP and LPP [3]. Specifically, GPP is used to create optimal trajectories in a known global environment, while LPP excels in obstacle avoidance and requires on-device online computation to effectively adapt to dynamic environments [4]. In practice, for reconnaissance or surveillance missions, mission plans are usually established in advance on the GCS through specialized software such as QGroundControl [5] and Mission Planner [6]. Therefore, focusing research on GPP solutions to establish the shortest flight path not only has profound scientific significance but also high practical value, helping to save flight time, energy, and ensure mission safety.

However, planning the optimal and safe trajectory for UAVs is always a challenging task because they must operate in 3D space. To date, many GPP methods have been studied and widely applied to UAVs and can be divided into four main approaches as in [7-18]. First, there are traditional graph-based search algorithms such as Dijkstra [7] and A* [8], which are the foundation for many modern navigation systems. In addition, the group of sampling-based methods, such as RRT* [9], has shown efficiency in rapid exploration in multidimensional space. In parallel, the group of biologically inspired methods has also attracted much research interest such as GA [10], improved A* combined with gray wolf optimization (GW-A*) [11], A* combined with flower pollination algorithm (Hybrid A*-FPA) [12], improved swarm optimization (LGPSO) [13], improved particle swarm optimization combined with genetic algorithm (PSO-GA) [14]. Finally, the development of machine learning and intelligent computing has opened up approaches such as using FL [15], RL methods [16,17], and hybrid algorithms combining deep reinforcement learning and fuzzy logic (DRL-FL) [18].

Received: 03 February 2026; Revised: 3 April 2026; Accepted: 20 April 2026.

Dijkstra [7] is an algorithm for finding the shortest distance between any two nodes on a graph with non-negative weights. However, a drawback of Dijkstra is that it performs searches in all directions without specific orientation toward the target. In large 3D spaces with many nodes, Dijkstra consumes large computational resources and processing time, making it less efficient for practical UAV applications. To overcome this limitation, the A* algorithm [8] was developed and became one of the most popular trajectory optimization methods. Currently, A* is widely applied in the field of robotics and is integrated into popular development platforms such as ROS, acting as an efficient global planner in the navigation framework [19,20]. A* improves search speed by combining the actual cost with a heuristic function to estimate the distance to the target, helping to orient the node expansion process more intelligently. However, traditional A* often uses a fixed heuristic function such as the Euclidean or Manhattan distance [21]. In complex environments with many obstacles, rigidly applying a heuristic can cause the algorithm to expand too many unnecessary nodes, increasing processing time. Research [9] proposed the RRT* algorithm in UAV trajectory planning, which allows the creation of asymptotically optimal reference paths and ensures collision-free operation in environments with complex obstacle structures. This method operates based on a random sampling mechanism to build a tree structure containing feasible paths, thereby determining the optimal path from the starting position to the destination [22]. In essence, RRT* is an enhanced version of RRT by incorporating an additional local optimization step in the tree expansion process. However, RRT* has several limitations, including high stochasticity, increased time cost, slow convergence rate, and requiring a large number of iterations to reach a better path [23].

Research [10] applied GA to the problem of UAV trajectory planning in 3D space. This method showed effectiveness in avoiding static and dynamic obstacles, with feasibility verified through both simulation and experiment. However, the inherent disadvantage of GA is the slow convergence speed and large computational load due to having to process the population over many generations, making it less efficient in applications requiring fast processing speed and limited hardware. Research [11] proposed the GW-A* algorithm for UAV path planning, which yields superior results compared to traditional A* by creating safer and more adaptable paths. However, the GW-A* algorithm significantly increases the computational load and system complexity compared to A* because the GWO weight adjustment process requires a long processing time to converge to the optimal solution, making it difficult for embedded UAV systems with limited hardware resources. Furthermore, the conclusion of this study also indicates that GW-A* is not superior to A* in all cases, and in some situations, A* is still better in terms of performance, while GW-A* has a higher computational cost. Study [12] applied A*-APF to optimal path planning for UAVs, exploiting A*'s high-speed discovery capabilities to find local optimal paths on a grid map and combining it with APF to refine and obtain the best optimal path. The performance of this method was compared with A*, FPA, GA, and PSO and showed optimal efficiency in all scenarios, especially in improving the best path length. Study [13] proposed the LGPSO algorithm for UAVs, which provided a 23% performance improvement over classical PSO in terms of optimal path distance. Using the same approach, study [14] proposed the PSO-GA algorithm for optimal 3D flight path planning of UAVs. Compared to traditional PSO, this algorithm enhances global search capabilities in the early stages and local optimization capabilities in the later stages. However, generally, swarm intelligence-based methods such as GWO, PSO, and FPA often have inherent limitations regarding slow convergence speed, computational resource consumption, and randomness of results (typically the dependence on the initial population state in GWO [24]). These factors make it difficult to ensure the stability and consistency of the mission in practical UAV applications.

Research [15] applied FL to the real-time flight path planning problem of UAVs. The authors combined A* and FL, resulting in a significant improvement in flight path smoothness. However, the effectiveness of FL is heavily dependent on the establishment of fuzzy rules based on experience [25], making it difficult to optimize flight paths in complex 3D environments. Meanwhile, studies [16,17] have applied RL based on the Q-Learning algorithm to improve the optimal flight planning of UAVs. Taking a more modern approach, research [18] combined DRL and FL to form a hybrid trajectory planning framework (DRL-FL) for UAVs, significantly enhancing training efficiency. Simulation results of DRL-FL have demonstrated effectiveness compared to traditional DRL. However, a common limitation of machine learning-based methods, such as RL and DRL, is the demanding requirement for sample data and training time. In addition, generalization ability is also considered a limitation of these methods. Conversely, grid-based search algorithms such as A* show superior stability due to their ability to directly process all spatial data without relying on predetermined training processes.

In summary, from the analyses above, although sampling-based methods, swarm intelligence, and machine learning have achieved certain successes, balancing computational speed, stability, and trajectory optimization in a 3D environment remains a major challenge for embedded UAV systems. Among these, the A* algorithm is still considered the most reliable solution due to its high determinism and stability. However, the biggest obstacle of traditional A* is the lack of flexibility of its fixed heuristic function, leading to wasted computational resources on unnecessary expansion nodes. To thoroughly address this problem, this paper proposes an improved trajectory planning method for A*. The core of the proposed method is the introduction of an adaptive heuristic strategy, allowing the algorithm to automatically switch between Manhattan and Euclidean standards based on the actual distance threshold to the target. This strategy helps the UAV significantly narrow the search space in the initial phase to increase target search speed, while minimizing the number of nodes that need to be expanded during target approach. In addition, to meet the smooth flight characteristics of the Quadcopter, the trajectory after search will be smoothed through cubic spline interpolation [26], ensuring continuity of velocity and acceleration. The proposed method was validated through simulations on MATLAB R2023b software and compared with other optimal flight trajectory methods based on grid graphs including

Dijkstra [7] and A* [8]. The results show a significant superiority of the proposed method in execution time and reduction of the number of nodes to be explored.

Thus, the main contributions of this study include: First, we propose an adaptive heuristic strategy to minimize the number of expansion nodes and reduce computation time in the optimal flight trajectory planning problem for Quadcopter UAVs. Second, we integrate Cubic Spline techniques [26] to smooth discrete trajectories, ensuring a continuous flight trajectory in terms of velocity and acceleration, consistent with the physical characteristics of the UAV. In addition, we verify the effectiveness of the proposed method through simulation scenarios compared with Dijkstra [7] and A* [8], showing significant superiority in execution time and reduction of the number of nodes to be explored. Unlike several existing improved A* variants—which often rely on stochastic optimization, learning mechanisms, or complex parameter tuning—the proposed method utilizes a simple yet well-directed heuristic function switching mechanism. This approach enhances search efficiency without significantly increasing computational complexity. Compared to swarm intelligence-based A* hybrids such as GW-A* [11] and A-APF* [12], the proposed method avoids reliance on stochastic optimization processes and iterative loops for convergence. Consequently, it reduces computational overhead while ensuring the safety of the global trajectory. Specifically, the flexible switching mechanism between Manhattan and Euclidean distances based on a distance threshold allows the algorithm to improve search directivity, thereby reducing the number of expanded nodes in the space. Furthermore, in contrast to other adaptive A* variants like A-FL* [15], the proposed method does not depend on expert knowledge for establishing fuzzy rules, which enhances the consistency and reproducibility of the results.

The rest of the paper is structured as follows: Section 2 presents the mathematical model and proposed method. Section 3 presents the simulation results and analysis. Finally, Section 4 provides conclusions and directions for future research.

2. Mathematical model and proposed method

2.1 Mathematical model of UAV Quadcopter

Currently, there are many different variants of UAV quadcopters, of which the three most common types include the "X" type (X-Type), the "+" type (+-Type), and the "H" type (H-Type). Each type has distinct operating characteristics, with the "X" type frame structure being the most common and applied in many commercial products, providing high flexibility and balanced thrust distribution, details of which can be observed in Figure 1 [27]. To simplify the calculation process while ensuring physical accuracy, the assumptions established include that the quadcopter frame is an absolute rigid body with a perfectly symmetrical structure, the center of gravity (CG) of the device coincides with the geometric center of the frame, and the influence of air resistance and environmental disturbances is negligible under ideal conditions.

Mathematically, the model of the Quadcopter UAV is described by a vector of 12 state variables, defined in two main reference coordinate systems, the Earth frame (E-Frame) and the body frame (B-Frame), as follows [2]:

$$X = \left[x, y, z, \phi, \theta, \psi, u, v, w, p, q, r\right]^T \tag{1}$$

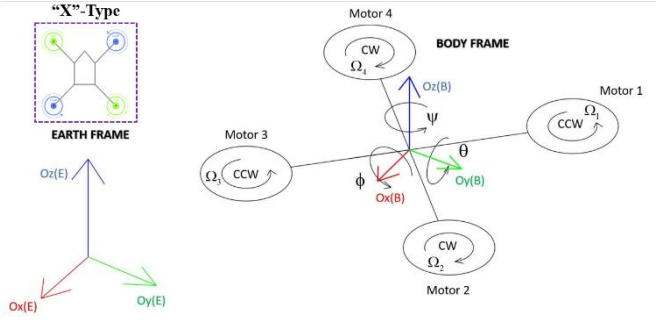

Figure 1: "X-Type" Quadcopter Model [27]

With (x, y, z) are the UAV's position within the Earth frame, (ϕ, θ, ψ) are the UAV's Euler angles which describe its orientation, (u, v, w) are the UAV's translational velocity as defined within the body frame, and (p, q, r) illustrate the corresponding angular velocities along the body frame's axes.

To convert the parameters between the E-Frame and B-Frame, we use the following matrices :

$$R = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi s\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$
$$T = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi \sec\theta & c\phi \sec\theta \end{bmatrix} \tag{2}$$

With, s, c, t are the mathematical notations for sin, cos, tan and $\sec\theta = (1/\cos\theta)$. The rotation matrix R is used to convert the translational velocity from the B-Frame to the E-Frame, and T is the conversion matrix to translate the angular velocity from the B-Frame to the rate of change of angular direction in the E-Frame.

The complete mathematical model of the Quadcopter UAV is described as follows [2,28]:

$$\begin{cases} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R \begin{bmatrix} u \\ v \\ w \end{bmatrix}; \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T \begin{bmatrix} p \\ q \\ r \end{bmatrix} \\ \ddot{x} = \frac{1}{m}\left((c\phi s\theta c\psi + s\phi s\psi)U_1\right) \;\; ; \;\; \ddot{\phi} = \dot{\theta}\dot{\psi}\frac{I_y - I_z}{I_x} + \frac{U_2}{I_x} \\ \ddot{y} = \frac{1}{m}\left((c\phi s\theta s\psi - s\phi c\psi)U_1\right) \;\; ; \;\; \ddot{\theta} = \dot{\phi}\dot{\psi}\frac{I_z - I_x}{I_y} + \frac{U_3}{I_y} \\ \ddot{z} = \frac{1}{m}\left(c\phi c\theta U_1\right) - g \;\; ; \;\; \ddot{\psi} = \dot{\phi}\dot{\theta}\frac{I_x - I_y}{I_z} + \frac{U_4}{I_z} \end{cases} \tag{3}$$

With, g is the acceleration due to gravity, and m is the mass of the device. Additionally, I_x, I_y, I_z are the moments of inertia with respect to the coordinate axes in three-dimensional space, and U_1, U_2, U_3, U_4 are the four control inputs, respectively.

Remark 1: The mathematical model of the Quadcopter UAV is a complex nonlinear system, fully described by a system of 12 tightly coupled state variables, which serves as the foundation for closed-loop feedback controllers to perform the task of tracking flight and maintaining stability.

However, in terms of practicality on currently popular flight platforms such as ArduPilot and PX4 [29], the operation process is clearly layered. Users only need to provide the system of waypoints in space, while the calculation and conversion to state variables of attitude and velocity will be automatically handled by the tracking control layers integrated in the device's firmware.

Therefore, within the scope of the global flight trajectory planning problem proposed in this study, we focus mainly on optimizing the flight trajectory related to the three coordinate positions (x, y, z) of the UAV in the E-Frame to find the shortest and safest route. Meanwhile, the state variables of attitude and velocity will be processed in the tracking control layer to implement the discrete trajectory smoothed by Cubic Spline [26], ensuring that the UAV moves accurately according to the established nodal points. It should be noted that, although the design of the trajectory tracking controller is an essential component for UAV operation, this study does not focus on developing closed-loop feedback control algorithms. Furthermore, it should be emphasized that the presentation of the mathematical model in this study aims to clarify the kinematic and dynamic nature of the UAV and the interrelationship between state variables, thereby ensuring that the generated trajectories are physically consistent. Although the global trajectory planner does not directly incorporate dynamic constraints, the resulting trajectories are designed to be continuous and geometrically feasible, allowing them to be efficiently followed by tracking controllers. This hierarchical approach is common in practical UAV systems, where the motion planning and control problems are decoupled to reduce computational complexity and enhance deployment flexibility.

2.2 Proposed method

2.2.1 The structure of the proposed method

To address the challenge of determining the optimal trajectory for a Quadcopter in a complex 3D environment, this study presents an enhanced trajectory planning method based on A* using an adaptive heuristic strategy. This allows the algorithm to automatically switch between Manhattan and Euclidean standards based on the actual distance threshold to the target. This strategy significantly narrows the search space in the initial phase to increase target acquisition speed, while minimizing the number of nodes that need to be expanded during the approach to the target. The structural diagram of the proposed method is generally described in Figure 2. The proposed method receives input data, including the starting position, target position, and environment map. Based on this information, the proposed method performs calculations through several key mechanisms including safety mechanisms, adaptive heuristic function strategies, and trajectory smoothing.

The safety mechanism is a pre-processing step in map data management that aims to establish safe buffer zones around obstacles. This phase ensures the UAV maintains the necessary distance from the obstacle to reduce the risk of collision due to tracking control errors or the actual physical size of the device. The adaptive heuristic strategy is a core component of the research, improving the A* algorithm by using a flexible conversion mechanism between Manhattan and Euclidean standards based on the actual distance threshold to the target. This strategy significantly narrows the search space in the initial phase to increase execution speed, while minimizing the number of nodes needed to expand to effectively approach the target. Finally, the discrete path obtained after the search is passed through a Cubic Spline interpolation block to transform the node points into a continuous curve, the output being the optimal trajectory ready for input into tracking controllers. This process ensures continuity in velocity and acceleration, making the trajectory kinematically feasible and guaranteeing smooth UAV operation.

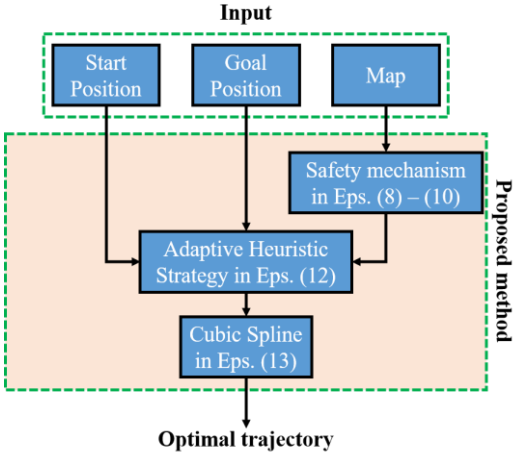

Figure 2: The schematic of the proposed method

2.2.2 Mathematical description of the proposed method

In the trajectory planning problem, processing the real-world environment into computer-understandable data is a prerequisite step. This phase includes two main processes: discretizing the space and expanding obstacles to ensure safety for the system. The UAV's workspace, denoted as $\Omega \subset \mathbb{R}^3$, is defined as a finite set bounded by the physical limitations of the environment or mission range. In this study, we transform the workspace into a 3D Occupancy Grid data structure, composed of unit cubes called Voxels, with the size of each Voxel defined by the resolution δ (m). Any real-world coordinate of a node $P = [x, y, z]^T$ in this space must strictly adhere to the following boundary constraints:

$$x_{\min} \le x \le x_{\max}, y_{\min} \le y \le y_{\max}, z_{\min} \le z \le z_{\max} \quad (4)$$

With $(x_{\min}, y_{\min}, z_{\min})$ represent the minimum coordinate values of the workspace along each corresponding axis. These values serve as the origin of the map, allowing the algorithm to locate the UAV even in areas with negative coordinates. Conversely, the maximum coordinate value $(x_{\max}, y_{\max}, z_{\max})$ of the workspace define the furthest distance the UAV is permitted to travel during a specific mission.

The relationship between actual coordinates P and the unit grid index (i, j, k) is established through the expression:

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_{\min} + i\delta \\ y_{\min} + j\delta \\ z_{\min} + k\delta \end{bmatrix} \quad (5)$$

With (i,j,k) as the grid cell address, constrained by the total number of corresponding grid nodes on each axis.

The grid node constraint is expressed as follows:

$$0 \le i \le N_x, 0 \le j \le N_y, 0 \le k \le N_z \tag{6}$$

With N_x, N_y, N_z are positive integers, defined as follows:

$$\begin{cases} N_x = \dfrac{x_{\max} - x_{\min}}{\delta} + 1 \\ N_y = \dfrac{y_{\max} - y_{\min}}{\delta} + 1 \\ N_z = \dfrac{z_{\max} - z_{\min}}{\delta} + 1 \end{cases} \tag{7}$$

In a complex 3D environment, the set of obstacles is denoted as $\Theta \subset \Omega$, including N_{obs} obstacles of any shape and size: $\Theta = \bigcup_{k=1}^{N_{obs}} \Theta_k$. To ensure the safety of the UAV, in this study, we have established a safety set $S \subset \Omega$ based on the Euclidean distance function and defined as follows:

$$S = \left\{ P \in \mathbb{R}^{3\times 1} : d(P,\Theta) > \gamma \right\} \tag{8}$$

Where $d(P,\Theta)$ is the shortest distance from a grid node P to the nearest point Q in the set of obstacles Θ and is determined by:

$$d(P,\Theta) = \min_{Q\in\Theta} \left(\|P - Q\|_2 \right) \tag{9}$$

The state of a grid node $P \in \mathbb{R}^{3\times 1}$ is determined by a collision indicator function $C(P)$ through the shortest distance from P to the set of obstacles Θ, taking into account the safety boundary γ to minimize the risk of physical collision in realistic scenarios. To be compatible with the A* algorithm, the state of each node is digitized through a collision indicator function and is defined by us as follows:

$$C(P) = \begin{cases} 1 & if \quad d(P,\Theta) \le \gamma \\ 0 & if \quad d(P,\Theta) > \gamma \end{cases} \tag{10}$$

In which $C(P) = 1$ is the dangerous region corresponding to $P \notin S$, and $C(P) = 0$ is the safe region corresponding to $P \in S$. Assigning the value 1 to the dangerous region and the value 0 to the safe region helps the proposed algorithm easily identify and eliminate unfeasible nodes during the search expansion process.

Remark 2: In practical scenarios, to minimize computational costs, obstacles Θ_k are often enclosed by simple geometric shapes. If the obstacle index k is enclosed by a sphere O_k centered with radius R_k, then the collision condition becomes $\|P - O_k\|_2 \le R_k + \gamma$. The γ determination depends on the actual physical size of the device and the margin of error arising during the orbital tracking flight.

After establishing the grid map and building the safety mechanism, the proposed method is implemented based on the structure of the improved A* algorithm to optimize the general cost function as follows [21]:

$$f(n) = g(n) + h(n) \tag{11}$$

With, $f(n)$ is the general cost function, $g(n)$ is the cost from the starting node to the target node, n and $h(n)$ is the heuristic function that estimates the distance from the node n to the target node.

To increase computational performance and 3D spatial orientation, this study proposes an adaptive heuristic strategy, allowing the algorithm to automatically and dynamically switch between distance standards based on an adaptive threshold η as follows:

$$h(n) = \begin{cases} \|P_n - P_G\|_1 & if \quad \|P_n - P_G\|_2 > \eta \\ \|P_n - P_G\|_2 & if \quad \|P_n - P_G\|_2 \le \eta \end{cases} \tag{12}$$

In which, $\|.\|_1$ representing the Manhattan standard and $\|.\|_2$ representing the Euclidean standard, P_n is the coordinates of the node n and P_G is the coordinates of the target. Applying the Manhattan standard when the UAV is still far from the target helps the algorithm significantly narrow the search space and increase the speed of target convergence. When the UAV enters the vicinity of the target, i.e., the distance is less than or equal to the threshold η, the algorithm automatically switches to the Euclidean standard to ensure that the established trajectory achieves optimal geometric shortness. This adaptive heuristic strategy minimizes the number of nodes that need to be expanded and optimizes the processing time for the system.

Remark 3: The adaptive heuristic strategy in (12) directly impacts the node expansion process of the A* algorithm. Based on the inequality $\|P_n - P_G\|_1 \ge \|P_n - P_G\|_2$, it is evident that when the distance to the target is large ($\|P_n - P_G\|_2 > \eta$), employing the Manhattan norm increases the heuristic value compared to the Euclidean norm. This, in turn, elevates the total cost function $f(n)$, enabling the algorithm to prioritize nodes closer to the optimal direction and prune less promising nodes early. Consequently, the search space is significantly narrowed, and the number of expanded nodes is reduced. Conversely, as the robot approaches the target ($\|P_n - P_G\|_2 \le \eta$), switching to the Euclidean norm becomes decisive for trajectory refinement. Since the Euclidean norm represents the shortest distance between two points in Euclidean space, it ensures the admissibility and consistency of the heuristic function. Furthermore, from the perspective of traditional A* theory, the number of expanded nodes depends directly on the set of nodes satisfying the condition $f(n) \le f^*(n)$, where $f^*(n)$ is the optimal total cost. Since the adaptive heuristic strategy enhances the heuristic value in regions far from the target and refines it in the vicinity, the set of nodes satisfying this condition is substantially smaller than when using a fixed distance norm. Therefore, the switching mechanism in (12) is not only empirically significant but also has a clear analytical foundation for reducing node expansion, thereby enhancing the computational efficiency of the algorithm.

Remark 4: The trajectory obtained from the adaptive A* algorithm can be described as a set of discrete nodes connected by straight lines, causing sudden changes in direction that are not compatible with the kinematic characteristics of the UAV at (3). To overcome this problem, the Cubic Spline interpolation technique [26] was applied to convert the nodes into a continuous and smooth trajectory.

Notably, this technique is effectively implemented through the built-in spline function in MATLAB's standard programming library. Therefore, within the scope of this paper, we focus on applying the Cubic Spline technique to the trajectory planning process to ensure physical feasibility, rather than delving into the fundamental mathematical foundations of widely proven spline interpolation.

To perform interpolation in 3D space, we parameterize the coordinates by the order variable $t \in [t_1, t_2, ..., t_N]$, where N is the index of the last node in the sequence of nodes that make up the discrete trajectory. Then each coordinate component $[x, y, z]^T$ is considered independently as a function of the parameter t. For each coordinate component, the set of interpolation points is $(t_0, P_0), (t_1, P_1), \ldots, (t_N, P_N)$. In each segment between the i and $i+1$ nodes, the Cubic Spline polynomial is defined as follows [26]:

$$Q_i(t) = a_i(t - t_i)^3 + b_i(t - t_i)^2 + c_i(t - t_i) + d_i \quad (13)$$

Where, $i = 0,1,...,N-1$, $\{a_i, b_i, c_i, d_i\}$ are polynomial coefficients defined such that the global interpolation function $Q(t) = Q_i(t)$ when $t \in [t_i, t_{i+1}]$ satisfies strict conditions regarding position continuity, velocity continuity, and acceleration continuity. The final result is an optimal, physically feasible trajectory, ready for execution by tracking controllers in real-world flight missions.

3. Simulation and results analysis

3.1 Environment setup and evaluation criteria

To evaluate the effectiveness of the proposed method, we performed simulation scenarios using MATLAB R2023b software. The performance of the proposed method will be directly compared with the Dijkstra algorithm [7] and the A* algorithm [8] based on important criteria, including processing time, path length, and number of nodes expanded. The scenarios were executed in a 3D grid map environment with a resolution of $\delta = 0.5$ (m). The survey space has the following dimensions: $-10 \le x \le 10$ (m), $-10 \le y \le 10$ (m), $0 \le z \le 15$ (m), corresponding to the 41×41×31 nodes. In the proposed method, the safety margin is determined based on the physical dimensions of the UAV combined with a compensatory margin accounting for the errors of the positioning and control systems, in order to minimize the risk of collisions in real-world operation. For the adaptive threshold, its value is determined through parameter sensitivity analysis. The results indicate that as the adaptive threshold increases, the search space tends to expand, leading to an increase in the number of nodes that need to be explored. Therefore, the selection of the adaptive threshold must strike a balance between limiting the number of expanded nodes and maintaining a low computational cost. Accordingly, in the simulation scenarios of this study, the safety margin radius and the adaptive threshold are set to $\gamma = 0.2$ (m) and $\eta = 1.5$ (m), respectively. It should be noted that these values are kept fixed throughout the simulation process to ensure consistency when comparing different methods. In practical applications, both the safety margin and the adaptive threshold can be flexibly adjusted depending on system characteristics, sensor accuracy, and specific safety requirements. The starting point is set to $P_0 = [0, -5, 0]$ and the target point is set to $P_G = [10, 5, 10]$. The trajectory of the proposed method is shown by the dashed blue line, Dijkstra [7] is the dashed pink line, and A* [8] is the dashed red line. To improve trajectory continuity, a smoothing step is applied as a post-processing stage using a cubic spline interpolation function available in MATLAB with default settings. The discrete trajectory points are interpolated with a parameter step of 0.1 to increase point density. It should be noted that this smoothing step does not take into account the UAV's kinematic constraints and is primarily intended to enhance the smoothness of the trajectory. Since cubic splines guarantee continuity up to the second derivative, the resulting trajectory exhibits smoothly varying velocity and acceleration, which is consistent with the dynamic characteristics of the quadcopter UAV described in (3). Consequently, it can be effectively tracked by trajectory tracking controllers.

In the simulation scenarios, obstacles in the environment are modeled as spheres to simplify distance calculations and collision checks in three-dimensional space. Although this is a geometric simplification, the model still ensures high suitability and reliability for algorithm validation. Dijkstra's algorithms [7] and A* [8] are implemented based on the original structure, finding paths based purely on geometric distance without applying additional safety margins. In contrast, the proposed method applies safety mechanisms (8) to (10) with safety margins to minimize the risk of physical collisions in real-world scenarios.

Processing times are calculated as follows [3]:

$$T = toc - tic \quad (14)$$

With, T is the total processing time used as an evaluation criterion is the total time, and in MATLAB, *tic* and *toc* is a pair of functions that records the time from the start of the search process to its completion

The path length is calculated as follows [3]:

$$L = \sum_{i=1}^{N} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \quad (15)$$

Where L is the total length of the path.

3.2 Environment with medium obstacle density

In this scenario, the UAV must perform trajectory planning from the starting point to the target point in an environment with 7 spherical obstacles. Figure 3 illustrates the trajectory planning process of the methods from different perspectives, while Table 1 details the collected evaluation parameters. The results observed from Figure 3 and the quantitative indicators collected in Table 1 show that the Dijkstra algorithm found the shortest path of 18.003 (m) but performed a wide-area search with a huge number of expansion nodes, up to 42,746 nodes, causing the trajectory to closely follow the obstacle surface due to the lack of safety margins, resulting in a high risk of collision in practice. For the traditional A* algorithm, applying a basic heuristic function significantly improves performance compared to Dijkstra, reducing processing time to 0.0272 seconds and the number of expansion nodes to 344 nodes. However, it can be seen that traditional A* still performs searches over a

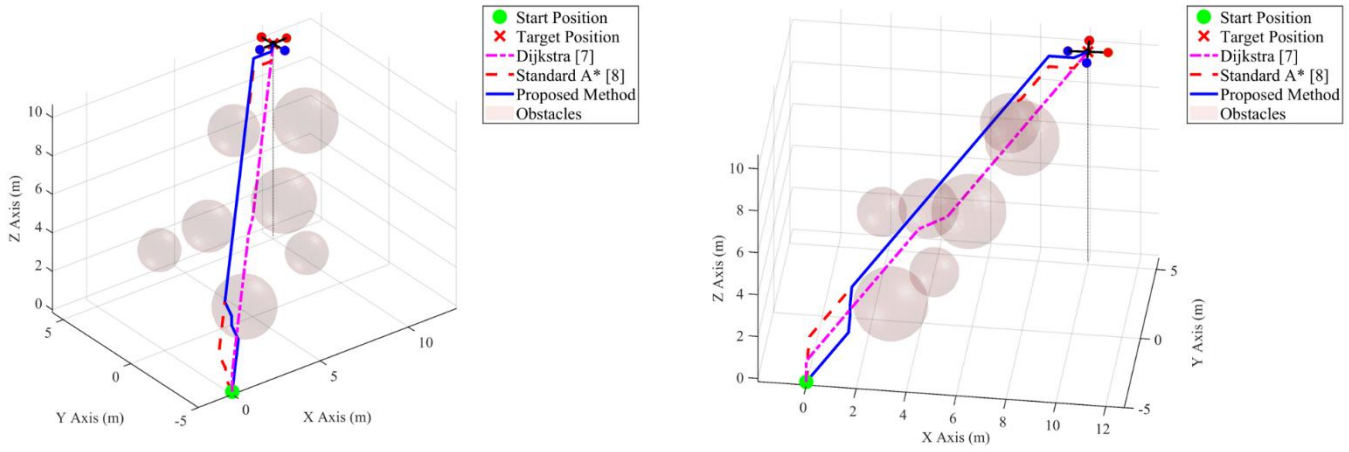

Figure 3: Path planning process in an environment with medium obstacle density

relatively large area, and due to the lack of a safety boundary mechanism, the resulting trajectory still carries the risk of being unsafe when the UAV operates near obstacles. Meanwhile, the proposed method with a safety boundary $\gamma = 0.2$ (m) and an adaptive threshold $\eta = 1.5$ (m) has created a trajectory that maintains a sufficiently safe distance from obstacles, ensuring feasibility for deployment in practical applications. Thanks to a flexible adaptive heuristic strategy between the Manhattan and Euclidean standards, the proposed method only requires 26 nodes and achieves an extremely fast processing time of 0.0044 seconds, thereby maximizing computational efficiency compared to traditional A* and Dijkstra. In particular, the flight path of the proposed method after processing with the Cubic Spline technique created a high-order continuous trajectory, which is fully compatible with the Quadcopter system dynamics model in equation (3) and ready for trajectory tracking controllers in actual flight missions.

In summary, the simulation results confirmed the effectiveness of the proposed algorithm in balancing safety, processing speed, number of expansion nodes, and trajectory quality. Minimizing the number of expansion nodes not only saves the UAV computational resources but also allows for rapid response in missions requiring high real-time performance. The resulting trajectory is fully compatible with the nonlinear dynamic characteristics and mechanical limitations of the Quadcopter UAV, providing a solid foundation for performing autonomous flight missions in real-world environments. The path length generated by the proposed method is 18,344 m, longer than Dijkstra's 18,003 m and equal to traditional A*, but this is a necessary trade-off to ensure the safety of the device. However, compared to Dijkstra [7], the processing time of the proposed method (0.0044 s) is about 603 times faster than Dijkstra (2.6575 s). This huge difference comes from the fact that Dijkstra has to traverse the entire state space and spends 42,746 expansion nodes to find the shortest path, while the proposed method focuses only on the most promising nodes through an adaptive heuristic function. In addition, the proposed method has a processing time more than 6 times faster than the traditional A* (0.0272 s). This demonstrates that the dynamic switching between Manhattan and Euclidean norm, depending on the target distance, has made the algorithm faster, reducing unneces-sary computations even compared to an algorithm with a basic heuristic. With 26 nodes to expand, the proposed method improves by 92.44% compared to the traditional A* method (344 nodes) and is significantly superior to Dijkstra.

Table 1: Evaluation parameters in an environment with medium obstacle density

Methods	Processing Time	Path Length	Number of ex-pansion nodes
Dijkstra [7]	2.6575 (s)	**18.003** (m)	42746
A* [8]	0.0272 (s)	18.344 (m)	344
Proposed Method	**0.0044** (s)	18.344 (m)	**26**

3.3 Environments with high obstacle density

In this scenario, the complexity of the environment is increased with the appearance of 15 densely arranged spherical obstacles in the survey space. These obstacles create many narrow passages and potential hazard zones, requiring the algorithm to have extremely precise orientation capabilities to find the optimal trajectory without consuming too many computational resources. Figure 4 illustrates the search process and trajectory results of the methods in a high-obstacle-density environment from different perspectives. Visually, it can be seen that the Dijkstra algorithm [7] and the traditional A*[8] continue to prioritize optimizing absolute distance by cutting the edges of the obstacles. However, in a high-density environment, going close to the obstacles makes this trajectory extremely dangerous and almost impossible to execute for the physical model of the Quadcopter. Conversely, the trajectory of the proposed method still maintains the necessary safety threshold, thanks to the buffer edge $\gamma = 0.2$ (m). Despite having to maneuver around obstacles to ensure safety, the algorithm still finds a continuous and smooth path. Thanks to Cubic Spline technology, the proposed trajectory does not have abrupt breaks, ensuring continuity of velocity and acceleration, reducing the load on the UAV's actuator system when having to navigate through narrow spaces.

The quantitative evaluation parameters collected from this scenario are detailed in Table 2. It can be seen that, even when the obstacle density doubles compared to the previous scenario, the proposed method only needs to expand by 24 nodes, significantly less than traditional A* (61 nodes) and far superior to Dijkstra (41,446 nodes). This confirms that the

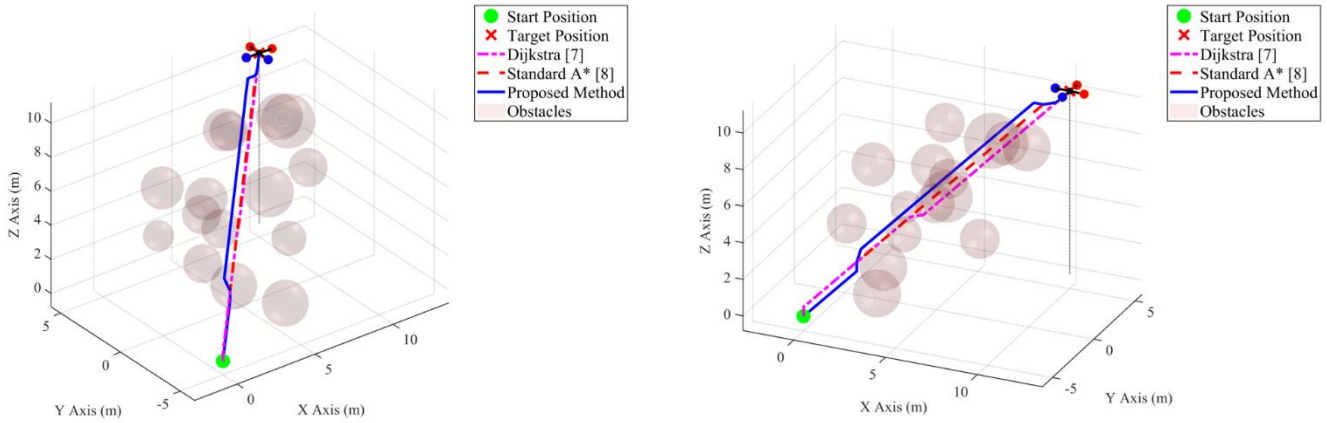

Figure 4: Path planning process in an environment with high obstacle density

adaptive heuristic strategy, combined with a safety mechanism, possesses excellent directional capabilities, allowing the algorithm to react safely and efficiently to surrounding obstacles. The processing time of the proposed method reaches 0.0069 s, 2.3 times faster than traditional A* (0.0162 s) and approximately 391 times faster than Dijkstra (2.702 s). This superior processing speed confirms the strong real-time responsiveness of the proposed method, allowing the UAV system to quickly plan its trajectory even in new missions requiring updates or changes to the operating environment map. Furthermore, the proposed method's range is 18,003 m, only about 1.9% longer than traditional methods (17,662 m). This increase is entirely worthwhile in exchange for absolute safety for the aircraft in complex environments. In summary, the proposed method demonstrated impressive performance in high-density scenarios. As environmental complexity increased, the algorithm's node expansion remained more efficient than A* and Dijkstra. This result confirms that combining flexible adaptive heuristics and safety margins not only makes UAVs safer but also optimizes computational resources to the maximum, laying a solid foundation for deployment on real-world UAVs.

Table 2: Evaluation parameters in an environment with high obstacle density

Methods	Processing Time	Path Length	Number of expansion nodes
Dijkstra [7]	2.702 (s)	**17.662** (m)	41446
A* [8]	0.0162 (s)	**17.662** (m)	61
Proposed Method	**0.0069** (s)	18.003 (m)	**24**

Although the processing time of the algorithms is evaluated in a MATLAB simulation environment, the obtained results still provide an important quantitative measure of the real-time capability of the proposed method. It should be noted that the computational cost on embedded systems depends not only on hardware capabilities but also on the efficiency of the algorithm's search process, which is typically reflected by the number of nodes expanded during trajectory planning. In addition, environmental characteristics such as grid resolution and obstacle density also influence the search space, thereby indirectly affecting the computational cost. In this study, under identical simulation conditions, the proposed method significantly reduces the number of expanded nodes compared to A* and Dijkstra, as quantitatively presented in Tables 1 and 2, thereby directly lowering the computational cost. Notably, algorithms such as A* and Dijkstra have been widely implemented in many real-world systems. With a substantial reduction in the number of expanded nodes compared to these two methods, the proposed approach demonstrates strong potential for efficient deployment in practical systems. At the same time, with the rapid development of high-performance embedded processors, the feasibility of real-world implementation is further enhanced. From the perspective of environmental complexity, it can be observed that as obstacle density increases, the processing time of the proposed method increases only slightly, while the number of expanded nodes remains low compared to A* and Dijkstra. This indicates that the adaptive heuristic strategy effectively controls the growth of computational cost, thereby improving scalability over conventional methods. Furthermore, based on the processing times reported in Tables 1 and 2, the proposed method exhibits the lowest computational cost among A* and Dijkstra, while maintaining stability across different simulation scenarios. However, implementation and evaluation on real embedded hardware, with CPU and memory constraints, remain an important direction for future work to fully validate the performance of the proposed method in practical UAV systems.

4. Conclusion and future works

This study successfully developed and validated an algorithm for optimizing the global flight trajectory of a Quadcopter UAV in a complex 3D environment. By introducing a flexible adaptive heuristic strategy between the Manhattan and Euclidean standards, the proposed method thoroughly addresses the limitations of execution time and redundant expansion nodes of traditional Dijkstra and A* methods. Quantitative simulation results in medium to high obstacle density scenarios demonstrate the superior performance of the proposed method. Furthermore, the practicality of the research is strongly reinforced through the safety boundary mechanism and Cubic Spline smoothing technique. The resulting trajectory not only reduces the risk of physical collisions but also ensures continuity in velocity and acceleration, perfectly matching the nonlinear dynamics

and mechanical limitations of the aircraft. In the future, research will focus on extending the algorithm to applications in environments with dynamic obstacles by simultaneously combining global and local planners. At the same time, close integration with intelligent trajectory tracking controllers will be considered to ensure that the UAV not only follows a geometrically optimal trajectory but also tracks it stably and accurately in the presence of disturbances and model uncertainties. The combination of the trajectory path-planning layer and the trajectory-tracking control layer aims to establish a comprehensive, closed-loop, and highly efficient solution for UAV systems in practical missions. Additionally, integrating the energy consumption model directly into the cost function is also a potential approach to optimizing the long-term operational performance of Quadcopter UAVs.

References

[1] Sy, M. T. N., Dinh, X. M., Do, Q. Q., Ta, N. T., Pham, V. D., Ta, N. D., & Bui, X. M. (2024, December). *Development of Autonomous Landing System for Reconnaissance UAVs*. In International Conference on Applied Mathematics and Computer Science (pp. 475-487). Cham: Springer Nature Switzerland.

[2] Dinh, X. M., Ngoc, T. A. T., Ta, N. T., Do, Q. Q., Pham, V. D., & Sy, M. T. N. (2025). *Adaptive Fuzzy Proportional Derivative Backstepping Hierarchical Sliding Mode Control for Trajectory Tracking of a Quadcopter UAV*. In 10th International Conference on Advanced Engineering – Theory and Applications (AETA). Springer.

[3] Cheriet, H., Badra, K. K., & Samira, C. (2024, November). *Comparative Analysis of UAV Path Planning Algorithms for Efficient Navigation in Urban 3D Environments*. In 2024 International Conference of the African Federation of Operational Research Societies (AFROS) (pp. 1-8). IEEE.

[4] DANG, T. V. (2024). *Optimization Hybrid Path Planning Based on A-star Algorithm Combining with DWA*. MM Science Journal, 10, 7551-7555.

[5] Ramirez-Atencia, C., & Camacho, D. (2018). *Extending QGroundControl for automated mission planning of UAVs*. Sensors, 18(7), 2339.

[6] Chintanadilok, J., Patel, S., Zhuang, Y., & Singh, A. (2022). *Mission planner: An open-source alternative to commercial flight planning software for unmanned aerial systems: Ae576/ae576, 8/2022*. EDIS, 2022(4).

[7] Dhulkefl, E., Durdu, A., & Terzioğlu, H. (2020). *Dijkstra algorithm using UAV path planning*. Konya Journal of Engineering Sciences, 8, 92-105.

[8] Mandloi, D., Arya, R., & Verma, A. K. (2021). *Unmanned aerial vehicle path planning based on A* algorithm and its variants in 3d environment*. International Journal of System Assurance Engineering and Management, 12(5), 990-1000.

[9] Tran, N. H., Dao, Q. T., & Ngoc-Tam, B. U. I. (2025). *Trajectory and parameter optimization in robust tracking control of a quadrotor*. IEEE Access.

[10] Gutierrez-Martinez, M. A., Rojo-Rodriguez, E. G., Cabriales-Ramirez, L. E., Estabridis, K., & Garcia-Salazar, O. (2025). *Genetic algorithm-based path planning of quadrotor UAVs on a 3D environment*. The Aeronautical Journal, 129(1334), 902-938.

[11] Haidar Ahmad, A., Zahwe, O., Nasser, A., & Clement, B. (2024). *Path planning for unmanned aerial vehicles in dynamic environments: A novel approach using improved A* and Grey Wolf Optimizer*. World Electric Vehicle Journal, 15(11), 531.

[12] Kareem, A. A., Mohamed, M. J., & Oleiwi, B. K. (2024). *Unmanned aerial vehicle path planning in a 3D environment using a hybrid algorithm*. Bulletin of Electrical Engineering and Informatics, 13(2), 905-915.

[13] Cheng, Q., Zhang, Z., Du, Y., & Li, Y. (2024). *Research on Particle Swarm Optimization-Based UAV Path Planning Technology in Urban Airspace*. Drones, 8(12), 701.

[14] Li, C., Zhao, Q., & Che, C. (2025). *3D Flight Path Planning for UAV Based on Improved Particle Swarm Optimization Algorithm*. IEEE Access.

[15] Zhou, Y., Wu, Z., & Qu, Y. (2024, September). *A Novel Real-Time Path Planning Method for UAV Based on Fuzzy Logic Sparse A* Algorithm*. In International Conference on Autonomous Unmanned Systems (pp. 281-291). Singapore: Springer Nature Singapore.

[16] Tu, G. T., & Juang, J. G. (2023, January). *UAV path planning and obstacle avoidance based on reinforcement learning in 3d environments*. In Actuators (Vol. 12, No. 2, p. 57). MDPI.

[17] de Carvalho, K. B., de OB Batista, H., Fagundes-Junior, L. A., de Oliveira, I. R. L., & Brandão, A. S. (2025). *Q-learning global path planning for UAV navigation with pondered priorities*. Intelligent Systems with Applications, 25, 200485.

[18] Xia, B., Mantegh, I., & Xie, W. F. (2024, July). *Hybrid Framework for UAV Motion Planning and Obstacle Avoidance: Integrating Deep Reinforcement Learning with Fuzzy Logic*. In 2024 10th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 2662-2669). IEEE.

[19] Dang, S. T., Dinh, X. M., Kim, T. D., Xuan, H. L., & Ha, M. H. (2023). *Adaptive backstepping hierarchical sliding mode control for 3-wheeled mobile robots based on RBF neural networks*. Electronics, 12(11), 2345.

[20] Dang, S. T., Dinh, X. M., Than Ngoc, T. A., Nguyen, V. T., Kim, D. T., & Le, X. H. (2025). *Adaptive Sliding Mode Control for Robust Trajectory Tracking of Car-Like Mobile Robots in Autonomous Navigation Systems*. Journal of Applied and Computational Mechanics, e19517.

[21] Liu, H., Cao, J., & Wang, Z. (2025). *Research on path planning of mobile robot in complex environment*. Discover Applied Sciences, 7(4), 1-18.

[22] Kumar, P., Pal, K., & Govil, M. C. (2025). *Comprehensive review of path planning techniques for unmanned aerial vehicles (uavs)*. ACM Computing Surveys, 58(3), 1-44.

[23] Li, X., Li, G., & Bian, Z. (2024). *Research on autonomous vehicle path planning algorithm based on improved RRT* algorithm and artificial potential field method*. Sensors, 24(12), 3899.

[24] Yu, M., Xu, J., Liang, W., Qiu, Y., Bao, S., & Tang, L. (2024). *Improved multi-strategy adaptive Grey Wolf Optimization for practical engineering applications and high-dimensional problem solving*. Artificial Intelligence Review, 57(10), 277.

[25] Dinh, X. M., Luong, T. G., Le, X. H., Nguyen, V. T., & Kim, D. T. (2025). *Adaptive Fuzzy Dynamic Surface Control with Sliding Mode Control for Enhanced Trajectory Tracking of Delta Robots*. Journal of Control, Automation and Electrical Systems, 1-25.

[26] De Boor, C., & De Boor, C. (1978). *A practical guide to splines* (Vol. 27, p. 325). New York: springer.

[27] Bui, X. V., Dinh, X. M., Ta, N. T., Ngoc, T. A. T., Do, Q. Q., Le, X. H., & Thai, H. N. (2025). *Trajectory tracking control for a quadcopter UAV based on sliding mode control and PD combined with fuzzy logic*. Journal of Science and Technology–HaUI.

[28] Madebo, N. W., Abdissa, C. M., & Lemma, L. N. (2024). *Enhanced trajectory control of quadrotor UAV using fuzzy PID based recurrent neural network controller*. IEEE Access.

[29] Mendoza-Mendoza, J. A., Gonzalez-Villela, V. J., Sepulveda-Cervantes, G., Mendez-Martinez, M., & Sossa-Azuela, H. (2020). *Advanced robotic vehicles programming: an ArduPilot and Pixhawk approach*. Apress.

[30] Dang, T. V., & Bui, N. T. (2023). *Obstacle avoidance strategy for mobile robot based on monocular camera*. Electronics, 12(8), 1932.