IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 September 2023, accepted 20 October 2023, date of publication 26 October 2023, date of current version 1 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327928

RESEARCH ARTICLE

ACACT: Adaptive Collision Avoidance Algorithm Based on Estimated Collision Time for Swarm UAVs

SEWOONG MIN AND HAEWOON NAM, (Senior Member, IEEE)
Department of Electrical and Electronic Engineering, Hanyang University, Ansan 15588, South Korea

Corresponding author: Haewoon Nam (hnam@hanyang.ac.kr)

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-RS-2023-00258639) supervised by the IITP (Institute for Information and Communications Technology Planning & Evaluation).

ABSTRACT The Adaptive Collision Avoidance Algorithm Based on the Estimated Collision Time (ACACT) is proposed in this paper, representing a novel approach designed for effective and efficient collision avoidance and path planning in highly dynamic environments, notably those with swarm Unmanned Aerial Vehicles (UAVs). The fundamental challenge in swarm UAV operations revolves around dynamic collision avoidance and nimble path planning. Addressing this, the ACACT algorithm exhibits the capability of predicting imminent collisions by estimating their likely occurrence times and then adeptly adjusting the UAVs' trajectories in real-time. A significant facet of the algorithm is the employment of adaptive target velocity, updated in accordance with the predicted collision timelines. This ensures not only that UAVs can sidestep potential collisions but also that they can pursue more direct and efficient routes in comparison to conventional methodologies. Highlighting its superiority over existing techniques, the ACACT algorithm successfully resolves some long-standing issues linked with the Artificial Potential Field (APF) method, especially concerning unreachability and oscillation. This is accomplished by integrating a strategic contingency plan coupled with enhanced obstacle navigation, particularly in proximity to target locations. For a comprehensive evaluation of the algorithm's prowess in collision avoidance and path planning, a novel metric named the Path Traveling Time Ratio (PTTR) is introduced. PTTR assesses both the traveling time taken for a vehicle to reach its target position and the duration it spends within collision-prone zones. This metric offers a more advanced evaluation method than merely comparing path lengths, collision counts, or traveling times. Through rigorous experimentation, it is observed that the ACACT algorithm enhances collision avoidance and path planning by an impressive margin of up to 20% compared to its traditional counterparts. Furthermore, a distinct advantage of the ACACT is its ability to uniformly tackle obstacles, irrespective of their speeds and independent of PID gain variations. It not only boosts the safety parameters but also amplifies operational efficiency, setting new benchmarks for UAVs to reach their target points with swiftness and security.

INDEX TERMS Swarm UAVs, collision avoidance of swarm UAVs, path planning of swarm robots, swarm robotics.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs), prevalent in robotics and military applications, are recognized for their agility and deployment versatility. These attributes make them suitable for diverse operations such as surveillance, search and tracking, cargo transportation, and farm management [1]. However, the scope of tasks that a single UAV can execute is finite, necessitating the exploration of swarm control for UAVs to optimize their utilization. Recent

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

studies have been focusing on tasks execution while concurrently avoiding potential collisions with obstacles or other agents [2], [3], [4], [5].

Key technologies for managing swarm UAVs encompass path planning and collision avoidance. In the absence of a collision avoidance algorithm during group movement or formation alignment, collisions may occur, potentially causing damage to the UAVs. A path planning algorithm that simultaneously avoids inter-UAV collisions, maintains a predetermined formation, and swiftly reaches the destination is integral to swarm air control. Nonetheless, it is challenging for UAVs to establish a real-time, smooth, and safe collision-free path. Over the past few years, extensive research has been conducted on path planning in an unmapped environment, where the UAV avoids real-time sensor-detected obstacles and navigates toward the destination. Several methodologies have been proposed, including the A* algorithm for offline path planning [6], [7], Rapidly-exploring Random Tree (RRT) for quickly exploring uncharted areas [8], [9], [10], Model Predictive Control (MPC) [11], [12], geometric-based methods [13], [14], and artificial potential fields (APF) [15], [16].

The A* algorithm, a heuristic search algorithm extensively employed, introduces global information to verify all possible nodes of the shortest path and estimates the distance from the current node to the destination. This estimation reflects the probability that the current node is part of the shortest path. The A* algorithm executes a global search for path planning under UAVs' dynamic constraints, generates a series of intermediate points, and initiates a local search to circumvent obstacles. Due to these characteristics, the A* algorithm is computationally complex and memory-intensive when planning a 3D path for UAVs. Additionally, post-processing to smooth the path is often required to apply the path to the UAV [6], [7].

The RRT is a sampling-based probabilistic algorithm designed to swiftly traverse unexplored areas in a search space. By randomly generating nodes and connecting them to the direction of their nearest neighbor, it achieves effective exploration. Kothari and Postlethwaite [9] applied chance constraints within the RRT to limit the probability of constraint violation and to navigate uncertain dynamic obstacles. To optimize computation time, Kuffner and LaValle [8] introduced a bidirectional RRT that conducts simultaneous searches from both the starting point and the destination. Furthermore, the RRT can find feasible paths in environments with high dimensional spaces and complexity. By integrating control functions with specific input parameters, this approach can be adapted to various constraints. However, paths derived from RRT often require refinement using the Dijkstra algorithm, which can extend the time needed for path adjustment.

MPC is a type of control system that optimizes control input considering current and expected future operations. This method uses the dynamic model of the system to predict future outputs and calculates the optimal control signal based on these predictions. MPC heavily depends on the accuracy of the model, and the calculation complexity can be high.

Geometric-based methods primarily find the optimal collision avoidance strategy based on the geometric properties of fixed-wing forms. For example, a geometric optimization approach to resolving collisions between planes minimizes the change in the velocity vector required to resolve collisions to obtain the minimum deviation from the normal trajectory. These methods are very efficient, but because they require a specific geometric relationship between the UAV and the obstacle, they can be difficult to apply in complex and dynamic environments [13], [14].

UAVs are commonly operated in dynamic environments characterized by high degrees of freedom and fast speeds. In such dynamic environments, each UAV in a swarm must be able to autonomously avoid collisions. The APF algorithm offers a mathematically simple and physically realistic approach that provides smooth paths suitable for robot and UAV applications. Consequently, the APF algorithm is well-suited for swarm UAVs that require agile and rapid movements. Because the APF algorithm does not guarantee complete collision avoidance, various algorithms have been explored to mitigate collision risks [17], [18].

Woods and La introduced an advanced version of the conventional potential field controller, termed the extended potential field controller (ePFC). This enhancement permits a drone to track dynamic targets while evading obstructions, offering improved performance in terms of smoother paths and reduced settling times. The stability of the ePFC was verified using the Lyapunov approach, and its effectiveness was demonstrated through lab experiments [19]. Meanwhile, Sun et al. focused on the complexities of path planning for drone formations in dense settings. They revamped the traditional artificial potential field (APF) algorithm, addressing challenges in multi-drone path planning in three-dimensional spaces. Their innovative approach ameliorated path oscillation issues and integrated a target exchange mechanism to circumvent local optimization traps. This improved algorithm was tested on a formation of 500 drones, proving its practical value in real-flight scenarios [20]. In parallel, Singletary et al. explored the relationship between the longstanding APFs and the more recent Control Barrier Functions (CBFs). They provided a theoretical linkage, indicating that a broad category of APFs can lead to the creation of CBFs that promise safety. These newly derived CBFs showcase versatility and are applicable to nonlinear systems. Practical tests further affirmed the potential of their method, especially for obstacle avoidance in both simulated and real environments, including for quadrotors with non-deterministic dynamics [21].

One challenge in implementing the APF algorithm is the setting of appropriate gains in the collision prevention Proportional-Integral-Differential (PID) controller [22]. If the gains are set too high, the resulting path becomes excessively long, while setting them too low compromises collision prevention. This indicates that the performance of the existing

PF algorithm in collision avoidance is highly sensitive to the values of PID gains. Achieving optimal gain values for dynamic mission environments often necessitates extensive trial and error.

Furthermore, the APF algorithm reduces the velocity of UAVs when maneuvering around obstacles, resulting in increased time required to reach the target. Moreover, it makes UAVs susceptible to local minima, which is a common issue in PF-based approaches. Additionally, accurately predicting the timing of collision with obstacles remains challenging within the existing PF algorithm, as collision probability increases based on obstacle or UAV velocities. Given the fragile nature of UAVs and their vulnerability to even minor collisions, collision probability directly impacts mission capabilities. Existing collision avoidance algorithms struggle to handle obstacles with varying speeds, thus increasing the likelihood of collisions.

To address these limitations, this paper proposes an adaptive collision avoidance algorithm based on the Estimated Collision Time (ACACT) algorithm. The goal of the proposed algorithm is to save travel time by shortening the path length to the destination while avoiding collisions with obstacles, regardless of their speed. It exhibits robustness in dynamic environments and reduced sensitivity to PID gains.

A. CONTRIBUTIONS

1) This paper presents a novel collision avoidance algorithm based on a potential field with adaptive target velocity. Specifically designed for swarm UAVs in dynamic environments, it prioritizes the shortest path to the target and effectively mitigates collisions using the estimated collision time. Additionally, a unique Path Traveling Time Ratio metric has been introduced, offering a comprehensive assessment of the algorithm's performance in both collision avoidance and path planning.
2) The ACACT algorithm showcased noteworthy enhancements, achieving up to a 20% improvement in collision avoidance and path planning over traditional methods. This includes providing consistent collision avoidance capabilities across varying-speed obstacles, regardless of PID gain modifications, thus bolstering the algorithm's versatility.
3) To comprehensively compare the established and the new algorithms, we devised a realistic simulator anchored on a physics engine, facilitating a meticulous evaluation of collision avoidance under diverse conditions.

II. PROBLEM STATEMENT

In this paper, the UAV employed is a quadrotor drone with a configuration of four rotors. This rotor arrangement grants the UAV unrestricted motion in all six degrees of freedom, enabling it to maneuver through three-dimensional paths in order to avoid obstacles and reach the target position.

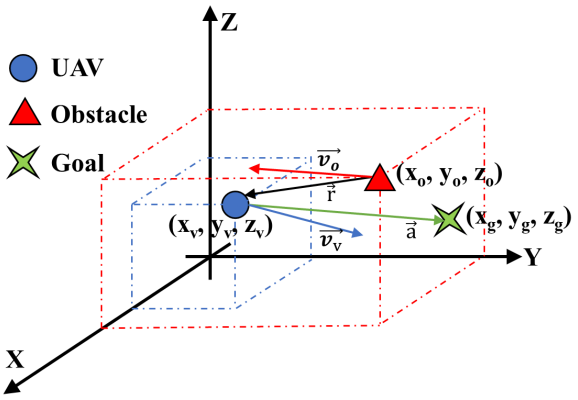

FIGURE 1. Attractive force and repulsive force at the UAV generated concerning the position of the obstacle and position of the goal.

The practical implementation of the algorithm proposed necessitates two foundational assumptions:

1) Every UAV within the swarm continually disseminates its respective position and velocity data via a dependable network infrastructure.
2) Each UAV is integrated with a sensor adept at ascertaining both the location and speed of impediments in every direction, having a maximal detection range denoted as r_s.

To simplify the calculations in the algorithm, each UAV is treated as a particle in 3D space. Therefore, although this paper primarily focuses on small UAVs, by increasing the size of the particles, the algorithm can be applied regardless of the UAV's size.

Figure 1 illustrates a UAV positioned in a Cartesian coordinate system commonly used as the East-North-Up (ENU) coordinate system [23]. The origin represents the starting position of the UAV, denoted as $\vec{p_v} = (x_v, y_v, z_v)$ in the coordinate system. $\vec{p_o} = (x_o, y_o, z_o)$ represents the position of the obstacle, and $\vec{p_g} = (x_g, y_g, z_g)$ represents the target position of the UAV. The velocities of the UAV and the obstacle are represented by vectors $\vec{v_v}$ and $\vec{v_o}$, respectively.

The control method employed for swarm drones is based on centralized control. While decentralized approaches, which offer scalability and improved system stability, are actively researched, they fall outside the scope of this paper's focus and will not be discussed in detail [24], [25].

For any vector $\vec{x}$, $\hat{\vec{x}}$ denotes a unit vector with a magnitude of 1 in the direction of $\vec{x}$.

A. GAIN OPTIMIZATION PROBLEM

Gain optimization is a common challenge in many control systems, where the performance is influenced by the choice of gain values. In most cases, a single gain optimization process can yield consistent performance when the control environment remains static. However, in the context of UAV path planning, the speeds of obstacles or other agents can vary, necessitating repeated gain optimization for different

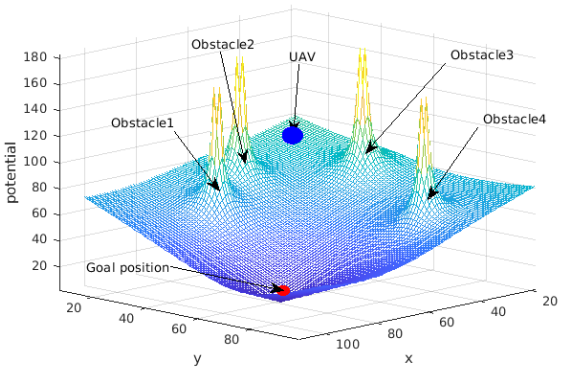

FIGURE 2. Example of an artificial potential field with a high potential obstacle and low potential target.

service environments. This results in the challenge of potential collision probabilities with fast-moving obstacles. There's a trade-off between optimizing gains to reduce these risks and ensuring efficient trajectories that save travel time and energy with slow-moving obstacles.

B. TARGET UNREACHABILITY PROBLEM

The target unreachability problem is a well-known limitation of existing APF algorithms, particularly when UAVs encounter static obstacles on their path to the target position. In such cases, the UAV may become trapped in local minima, preventing it from reaching the intended target position. This occurs when the repulsive forces to avoid obstacles and the attractive forces towards the target position balance each other, resulting in a net force of zero and stagnation of the UAV's motion.

C. OSCILLATION PROBLEM

The oscillation problem arises when obstacles are located near the UAV's target position. The oscillation problem occurs when the UAV experiences repetitive oscillations near the target position due to the interplay of increasing repulsive forces from obstacles and diminishing attractive forces as the UAV approaches the target. Over time, the amplitude of these oscillations tends to decrease, eventually leading to the UAV reaching the target position after a significant duration.

III. PROPOSED COLLISION AVOIDANCE ALGORITHM FOR SWARM UAVS

UAVs are required to respond rapidly to changes in their surrounding environment and generate real-time paths that are both efficient and safe. The APF algorithm is widely used for path planning in robotics, particularly in scenarios where map information is limited and dynamic obstacles are prevalent. The appeal of the APF algorithm lies in its simplicity and its ability to generate smooth trajectories that facilitate collision avoidance. Unlike approaches that search for a global trajectory, the APF algorithm generates local paths in close proximity, resulting in smoother paths for robots to follow. However, the APF algorithm necessitates careful optimization of gain values to effectively avoid collisions. Furthermore, even with gain optimization, there remains the possibility of collisions with faster obstacles that were not accounted for during the optimization process. Conservative gain optimization can result in overly long detours around slower obstacles, leading to inefficient path planning. To address these limitations, we propose an enhancement to the APF algorithm by introducing the concept of estimated collision time. Specifically, we adjust the change in path direction based on the velocity of obstacles, enabling the algorithm to provide stable collision avoidance paths for obstacles with varying speeds. As a result, our proposed algorithm ensures UAV safety even in scenarios where gain optimization has not been performed, distinguishing it from previous approaches that exhibited varying collision avoidance performance based on gain optimization.

A. APF ALGORITHM

Fig. 2 illustrates the concept of an artificial PF algorithm. In the traditional APF algorithm introduced by Khatib in 1986, the avoidance of collisions is achieved through the utilization of attractive and repulsive forces [26]. The attractive force, represented as a force vector, is directed towards the desired target location to guide the robot in reaching its destination. Conversely, the repulsive force, also expressed as a force vector, points in the opposite direction of obstacles, aiming to prevent collisions by creating a repulsive effect.

In Fig. 1, the attractive force vector $\vec{a}$ pulling the UAV to the goal position is given as

$$\vec{a} = (x_g - x_v, y_g - y_v, z_g - z_v). \tag{1}$$

In addition, the relative position vector $\vec{r}$, representing the repulsive force to the UAV according to the position of the obstacle, is obtained as

$$\vec{r} = (x_v - x_o, y_v - y_o, z_v - z_o). \tag{2}$$

The repulsive force that increases as the distance to the obstacle decreases is obtained as

$$\vec{r}_p = \frac{r_s^2}{\|\vec{r}\|^2}\hat{\vec{r}}, \text{ if } \|\vec{r}\| < |r_s|\,, \tag{3}$$

where the radius r_s represents the range within which obstacles are considered for applying the repulsive force. The unit direction vector $\hat{\vec{r}}$ is derived from the vector $\vec{r}$, indicating the direction from the UAV to the obstacle. The value of r_s is determined by the radius of the sensor installed on the UAV, which enables obstacle detection. (3) illustrates that the magnitude of $\vec{r}_p$ is 1 when the obstacle is located at the maximum sensing range of the UAV. Conversely, if the obstacle is outside the sensing range, the magnitude of $\vec{r}_p$ is 0. Consequently, as obstacles approach the UAV within the radius *rs*, a stronger repulsive force is exerted to prevent a collision. It should be noted that in the PF algorithm, the effectiveness of collision avoidance is influenced by the repulsive force.

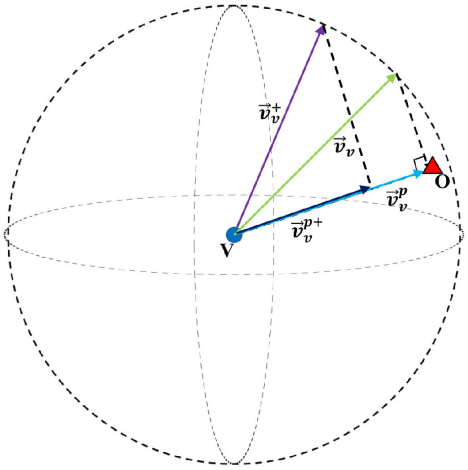

FIGURE 3. Target velocity $\vec{v}_v^+$ updated by projection target velocity $\vec{v}_v^{p+}$.

Let $\vec{v}_v$ represent the target velocity that governs the trajectory of the UAV. It serves as a control input to ensure that the current velocity of the UAV quickly converges to the target velocity. The value of $\vec{v}_v$ is determined by combining attractive and repulsive forces according to the following as

$$\vec{v}_v = k_{pa}\vec{a} + k_{pp}\vec{r}_p, \tag{4}$$

where k_{pa} denotes the proportional (p) gain of the PID control for the attractive force, and kpp represents the p gain for the repulsive force. Choosing excessively high values for these gains leads to a rapid increase in UAV movement upon entering the sensing range due to the repulsive force, which can result in flight instability or oscillation. Conversely, setting these gains too low may lead to collision incidents as obstacles approach. Therefore, determining appropriate values for the p gains is crucial. The stability and collision avoidance performance of the system are highly sensitive to these gains. However, finding an optimal value for the gains proves challenging, as they need to be adjusted based on the obstacle or UAV's speed. It is important to note that while (4) employs only p gains for simplicity, in practical implementations, the p, i, and d gains should all be set to their optimal values to achieve effective collision avoidance performance.

B. ADAPTIVE COLLISION AVOIDANCE BASED ON ESTIMATED COLLISION TIME

1) ROBUST PATH GENERATION BASED ON ESTIMATED COLLISION TIME

In order to enhance the collision avoidance performance in the presence of obstacles with different velocities, an additional repulsive force that takes into account the relative velocity between the UAV and the obstacles is incorporated. The relative velocity is calculated as

$$\vec{v}_r = \vec{v}_o - \vec{v}_v. \tag{5}$$

The repulsive force generated by the relative velocity vector $\vec{v}_r$ is directed perpendicular to the relative velocity vector, following the approach proposed in [27]. Since the UAV operates in a three-dimensional space, there are multiple vectors that are perpendicular to the relative velocity vector. In this paper, only the vectors perpendicular to the direction moving away from the obstacle in the plane are considered, which includes the vectors $\vec{r}$ and $\vec{v}_r$. The repulsive force in the normal direction is defined as

$$\vec{r}_{vn} = \begin{cases} \dfrac{\hat{\vec{r}}}{\hat{\vec{r}} \cdot \hat{\vec{v}}_r} - \hat{\vec{v}}_r & \text{if } 0 < \hat{\vec{r}} \cdot \hat{\vec{v}}_r < 1 \\ 0 & \text{else .} \end{cases} \tag{6}$$

The equation is applicable when the angle between $\vec{r}_p$ and $\vec{v}_r$ is less than 90°, satisfying the condition $0 < \hat{\vec{r}} \cdot \hat{\vec{v}}_r < 1$. Additionally, the target velocity of the UAV can be determined by considering $\vec{r}_{vn}$ as

$$\vec{v}_v = k_{pa}\vec{a} + k_{pp}\vec{r}_p + k_{pv}\vec{r}_{vn}. \tag{7}$$

The concepts of projected target velocity and estimated collision time are employed for adaptive collision avoidance. These concepts allow for the anticipation of the target velocity and the prediction of the time until a potential collision. By incorporating these estimates, the system can dynamically adjust its trajectory and take proactive measures to avoid potential collisions. The estimated collision time is obtained by considering the distance to the obstacle and the target velocity. For objects moving at a constant velocity, the time it takes to traverse a certain distance can be determined by dividing the distance by the velocity. Similarly, when the UAV maintains its current target velocity, the estimated collision time is defined as the duration required to collide with the obstacle. However, since the direction of the target velocity may not align with the obstacle, the target velocity is projected in the direction of the obstacle for precise calculations. By continuously updating the target velocity based on obstacle avoidance, the UAV's trajectory is dynamically generated.

The target velocity $\vec{v}_v$ can be obtained as explained in (7). In Fig. 3, $\vec{v}_v^p$ refers to the target velocity projected to $-\vec{r}$, which is calculated as

$$\vec{v}_v^p = -\|\vec{v}_v\| \cos\theta \cdot \hat{\vec{r}}, \tag{8}$$

where θ means the angle between $-\vec{r}$ and $\vec{v}_v$. furthermore, $\cos\theta$ is as

$$\cos\theta = \frac{-\vec{v}_v \cdot \vec{r}}{\|\vec{v}_v\| \cdot \|\vec{r}\|}. \tag{9}$$

Substituting (9) into (8), we get

$$\vec{v}_v^p = \frac{\vec{v}_v \cdot \vec{r}}{\|\vec{r}\|} \cdot \hat{\vec{r}}. \tag{10}$$

Let t_c be the estimated time until the UAV collides with an obstacle in the future, assuming that $\vec{v}_v$ and $\vec{r}$ are on the same line. The estimated collision time t_c in the direction of the obstacle can be obtained as

$$t_c = \frac{\|\vec{r}\|}{\|\vec{v}_v^p\|}, \tag{11}$$

which is obtained using (10) and (11) as

$$t_c = \frac{\|\vec{r}\|^2}{|\vec{v}_v \cdot \vec{r}|}. \tag{12}$$

Let t_s be the time required to avoid collision when a collision with an obstacle is expected. It is the only parameter that should be set appropriately in the proposed ACACT algorithm and can be calculated physically as opposed to the PID gains that must be set experimentally in general. For example, if r_s is 8 m and the maximum speed of the UAV is 2 m/s, t_s can be set to a maximum of 4 s. This prioritizes collision avoidance, even if it results in slightly longer paths. If t_s is set between 2-3 s, the algorithm will choose a path that is slightly closer to the obstacle, which, although dangerous, can reach the target point faster. If $t_c < t_s$, the conventional algorithms do not have enough time to avoid collisions, thus the potential collision probability is high. On the other hand, the proposed ACACT algorithm performs the following to make t_c larger than t_s to reduce the potential collision probability. First, when t_s is set to a specific value, the updated projection target velocity $\vec{v}_v^{p+}$ is calculated as

$$\left\|\vec{v}_v^{p+}\right\| = \frac{\|\vec{r}\|}{t_s}. \tag{13}$$

The adjustment of $\vec{v}_v^{p+}$ aims to increase the collision time t_c to be greater than or equal to the expected collision time t_s. This adjustment ensures that the UAV has sufficient time to avoid collisions with obstacles. The updated target velocity $\vec{v}_v^{+}$ is derived from $\vec{v}_v^{p+}$, allowing the UAV to navigate towards the target point efficiently while maintaining its speed. To calculate $\vec{v}_v^{v}$, a vertical vector is required, which plays a crucial role in determining the direction of motion that avoids obstacles. The calculation of the vertical vector $\vec{v}_v^{v}$ is as

$$\vec{v}_v^{v} = \vec{v}_v - \vec{v}_v^{p}. \tag{14}$$

This value is used to obtain the updated target velocity $\vec{v}_v^{+}$ as

$$\vec{v}_v^{+} = \left\|\vec{v}_v^{p+}\right\| \hat{\vec{v}}_v^{p} + \sqrt{\|\vec{v}_v\|^2 - \left\|\vec{v}_v^{p+}\right\|^2}\,\hat{\vec{v}}_v^{v}, \tag{15}$$

where, by taking the squared sum of the coefficients of the two integrated vectors, it becomes evident that the magnitude of $\vec{v}_v$ remains unchanged. The updated $\vec{v}_v^{+}$ leverages the direction vector of the projection velocity and the vertical velocity components within $\vec{v}_v$ to determine the shortest path while avoiding obstacles.

The obtained $\vec{v}_v^{+}$ generates a path in a safe direction based on the estimated collision time derived from the projection velocity aligned with the obstacle's direction. In this regard, the ACACT algorithm demonstrates adaptive collision avoidance performance that is less reliant on collision avoidance PID gains, thanks to the utilization of estimated collision time. Moreover, the parameter t_s, which is easily comprehensible, ensures that the projection velocity remains collision-free within a specified timeframe, thereby determining the path for obstacle avoidance. Notably, t_s, which relates to the maneuverability of the UAV, must be accurately calculated, eliminating the need for extensive trial and error typically associated with conventional collision avoidance algorithms.

Algorithm 1 Contingency Plan

```
if ‖ā‖ > 1, ‖v⃗_v^+‖ < 0.5 then
  for all obstacle o ∈ r_s do
    compute p⃗_r = p⃗_v − p⃗_o
    compute phi = arccos(p⃗_r.x / r_s)
    compute theta = arctan(p⃗_r.z, p⃗_r.y)
    append values in the ob_poses(phi, theta)
  end for
  get random value r_phi = uniform(0, π/2)
  get random value r_theta = uniform(0, 2π)
  if ((r_phi, r_theta) ∈ ob_poses then
    get random value r_phi = uniform(0, π/2)
    get random value r_theta = uniform(0, 2π)
  else
    compute x = −0.8 * r_s * cos r_phi
    compute y = 0.8 * r_s * sin r_theta * cos r_theta
    compute z = 0.8 * r_s * sin r_theta * sin r_theta
    return x, y, z
  end if
end if
```

2) CONTINGENCY PLAN

As a characteristic of the APF algorithm, which iteratively generates paths within local regions to reach the target position, it is necessary to have a contingency plan to overcome situations where the algorithm becomes trapped in local minima and fails to reach the target position. To determine whether the UAV is trapped in a local minima, we can assess its position, velocity, and target point. If the magnitude of vector $\vec{a}$ in (1) is greater than 0 and the UAV's velocity approaches zero, it indicates that the UAV is indeed trapped. In such a situation, a contingency plan needs to be initiated.

When trapped in a local minima, obstacles typically obstruct the UAV's forward movement. Therefore, to escape, it is advisable to move backward. Since information about obstacles is only available within a radius of r_s, selecting a random position within this radius in the backward direction without obstacles is preferred. The algorithm for implementing this contingency plan can be summarized as shown in Algorithm 1. The escape radius is set to 80% of the obstacle radius, and a random position is generated within this radius in the backward direction, ensuring it is obstacle-free. Finally, the UAV is moved to the randomly selected position to successfully escape the local minimum.

3) OSCILLATION CANCELLATION

Oscillation phenomena in UAVs caused by obstacles near the destination occasionally occur. While these oscillations

generally diminish or disappear over time, minimizing the time required for UAVs with shorter flight durations to reach the destination is desirable. The first step is to determine whether the UAV has approached the target position and if there are obstacles in close proximity. By selecting an arbitrary distance, denoted as r_{ref}, which is smaller than the detection radius r_s, it is possible to determine if the UAV is near the destination. Additionally, a positive value of the repulsive force $\vec{r}_p$, derived from relative positioning, indicates the presence of nearby obstacles. Situations involving obstacles near the target position can be classified into two categories: those where the obstacles are in motion and those where they are stationary. In the case of moving obstacles, they do not significantly contribute to UAV oscillation, and collision avoidance techniques can be employed. On the other hand, if the position of a stationary obstacle does not align with the target position, temporarily disabling the repulsive force is an effective approach to reduce oscillation. Therefore, (7) is modified as

$$\vec{v}_v = \begin{cases} k_{pa}\vec{a} & \text{if } \|\vec{a}\| < r_{ref}, \vec{r}_p > 0, \vec{r}_{vn} < 0.2 \\ k_{pa}\vec{a} + k_{pp}\vec{r}_p & \\ +k_{pv}\vec{r}_{vn} & \text{else}, \end{cases} \tag{16}$$

C. COLLISION AVOIDANCE FOR SWARM UAVS

The first requirement for controlling a swarm of UAVs is a unified global coordinate system. Typically, each UAV maintains its own local coordinate system for controlling its own position. The system that controls the entire swarm issues commands based on the global coordinate system, so each UAV shares its coordinates with the system by transforming them using the following as

$$\begin{bmatrix} x_l \\ y_l \\ z_l \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & t_x \\ \sin\theta & \cos\theta & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_g \\ y_g \\ z_g \\ 1 \end{bmatrix} \tag{17}$$

where x_l, y_l, and z_l represent the local coordinate system of the UAV, whereas x_g, y_g, and z_g represent the unified global coordinate system. Additionally, θ denotes the angle between the local coordinate system and the global coordinate system.

In a swarm UAVs, it is necessary to avoid collisions not only with obstacles but also with other agents within the swarm. The proposed ACACT algorithm encompasses collision avoidance from multiple obstacles, including other agents.

The repulsive force for the ith vehicle in a group of N units is determined as

$$\vec{r}_i = \sum_{j=0, j\neq i, \|\vec{r}_j\| < |r_s|}^{N} \vec{r}_j, \tag{18}$$

where $\vec{r}_i$ represents the resultant force calculated in the direction of the vector sum of all the repulsive forces exerted by known obstacle locations and other vehicles within a radius of r_s around the ith vehicle. The repulsive force acting against the most imminent obstacle to be avoided within the r_s range can be determined as

$$\vec{r}_{min} : \|\vec{r}_i\| \leq \left\|\vec{r}_j\right\|, i \neq j \quad \forall i, j \in N, \tag{19}$$

where $\vec{r}_{min}$ is $\vec{r}$ used to determine t_c of the proposed algorithm.

Algorithm 2 Swarm Collision Avoidance

```
for all vehicle d ∈ Swarm do
  Coordinate Transform p_g top_l
  Compute ā, r̄_p, r̄_vn, r̄_min
  if ‖ā‖ < r_ref, r̄_p > 0, r̄_vn < 0.2 then
    compute v̄_vi = k_pa ā
  else
    compute v̄_vi = k_pa ā + k_pp r̄_pi
  end if
  if ‖ā‖ > 1, ‖v̄_v^+‖ < 0.5 then
    execute Contingency Plan
  end if
  compute t_c for r̄_min
  if t_c < t_s then
    compute v̄_vi^+ for r̄_min
    return v̄_vi^+
  else
    return v̄_vi
  end if
end for
```

Algorithm 2 summarizes the aforementioned algorithm. The ACACT algorithm is, in theory, adaptable to extensive multi-UAV systems, encompassing between 100 and 1,000 UAVs. The scalability of this system hinges on two pivotal assumptions. Firstly, the position and velocity of every UAV in the swarm should be disseminated in real-time via a robust network infrastructure. Secondly, the ACACT algorithm necessitates a decentralized control architecture for individual UAV control. When realized through this distributed control paradigm, the algorithm exhibits a computational complexity of $O(n)$, seemingly escalating with the number of UAVs. However, as each UAV is only tasked with avoiding other UAVs within a radius of r_s, the computational demand scales with r_s, independent of the UAVs quantity n. Consequently, the ACACT algorithm assures both scalability and real-time operability.

D. PERFORMANCE EVALUATION METRIC

Dynamic path planning algorithms for UAVs in 3D space are currently limited, and there is a lack of suitable metrics for evaluating their performance. To address this challenge, we propose the path traveling time ratio (PTTR) as a novel measure to assess the effectiveness of collision avoidance and path planning. PTTR is defined as the ratio of the actual path traveling time to the ideal path traveling time, as shown in Fig. 4. It provides a quantitative measure of the efficiency and effectiveness of the UAV's trajectory planning in navigating

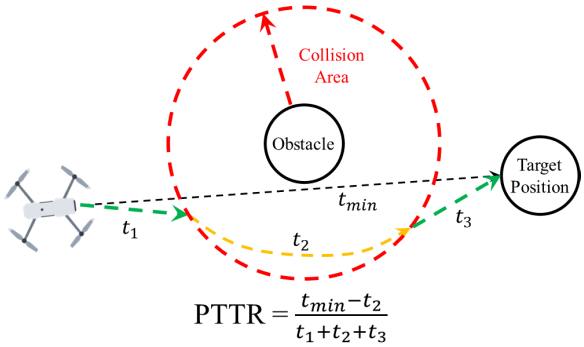

FIGURE 4. An example scenario showing how PTTR is calculated. In this scenario, t_{min} represents the time it takes to fly the shortest path at maximum velocity.

complex environments.PTTR is defined as

$$PTTR = \frac{\frac{d}{v_{max}} - t^c_{travel}}{t_{travel}}, \tag{20}$$

where d is the distance between the current position and goal, t_{travel} is the time taken to move from the current position to the goal position, v_{max} is the maximum velocity of the UAV, and t^c_{travel} is the time spent flying in a collision risk area. The first term of PTTR is introduced to evaluate the path planning performance of algorithm. This is defined as the travel time ratio (TTR). If the maximum velocity set in the UAV is v_{max}, d/v_{max} can be regarded as the minimum time for the UAV to reach the target point, which means the shortest path. Thus, TTR is ideally 1 when the UAV moves at the highest speed over the shortest path, and a value gets close to 0 when the actual arrival time takes longer. Therefore, if an algorithm is evaluated using TTR, one can see how quickly it reaches its goal position. In other words, TTR measures how closely the UAV follows the shortest path. However, TTR alone does not indicate whether a vehicle experiences collisions.

Therefore, the second term in (20) is defined as the collision area occupancy time ratio (CTR), which quantifies the proportion of time the UAV spends within the collision risk area relative to its total travel time. In this paper, the collision risk area is determined as a radius of 2 meters around the UAV, considering its size to be 0.5 meters. It is recommended to set the collision risk area as approximately 1.5 meters larger than the UAV size to account for variations in performance evaluation. For instance, if the UAV spends 1 second within the collision risk area during a total travel time of 8 seconds, the CTR would be calculated as 0.125. A CTR value of 0 indicates optimal collision avoidance performance, while smaller values signify more effective avoidance of collisions.

PTTR is theoretically bounded between 1 and -1, but in practice, TTR is typically greater than CTR, resulting in PTTR values ranging from 1 to 0. PTTR serves as a comprehensive metric for assessing path planning performance, encompassing both collision avoidance capability and overall path quality. A PTTR value close to 1 indicates superior path planning performance with effective collision avoidance. The PTTR metric is applicable across diverse environments. In a given scenario, if Algorithm A exhibits a superior PTTR value compared to Algorithm B, it suggests that Algorithm A is a more efficient path-planning algorithm. To compute the PTTR value, one requires the UAV's initial position, its target position, the total traveling time, and the time expended within the collision-area. To evaluate the algorithms, at least 30 experiments are conducted per scenario, and the highest and lowest PTTR values are excluded before computing the average. The PTTR graphs for each algorithm are presented in Section IV-B.

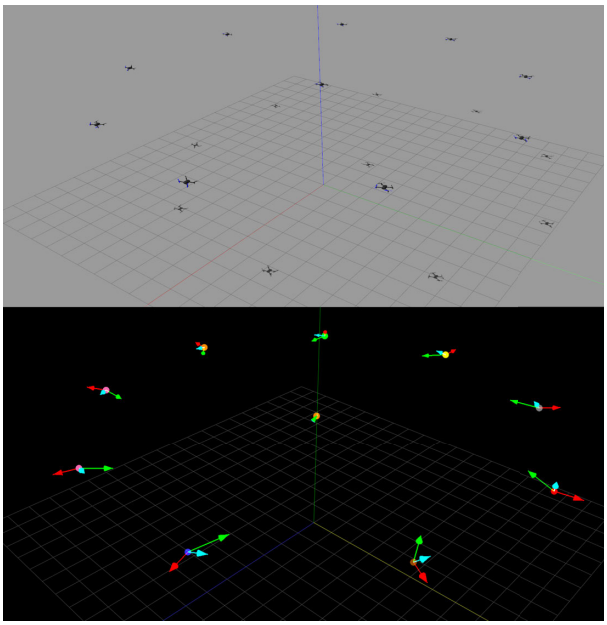

FIGURE 5. Formation flight with 10 iris in Gazebo environment(upper). Attractive and repulsive forces and the target velocity as a result of the ACACT algorithm acting on each UAV(lower).

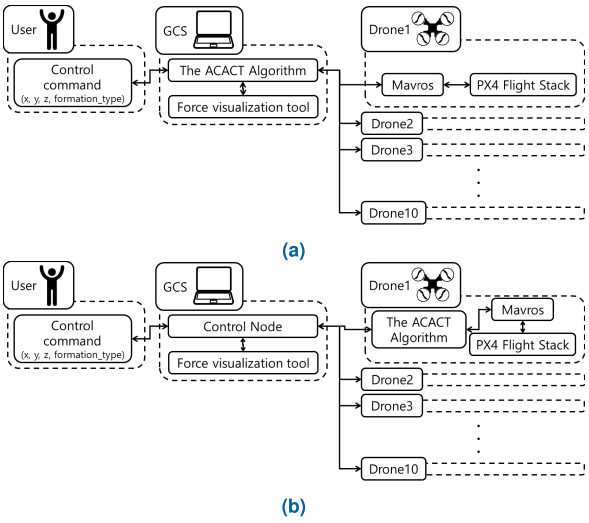

FIGURE 6. Software architecture of the simulator. (a) centralized architecture, (b) decentralized architecture.

IV. SIMULATION AND EXPERIMENTAL RESULTS

A. SIMULATION ENVIRONMENT

To evaluate the ACACT algorithm, a simulator has been implemented using Gazebo, a widely adopted platform in

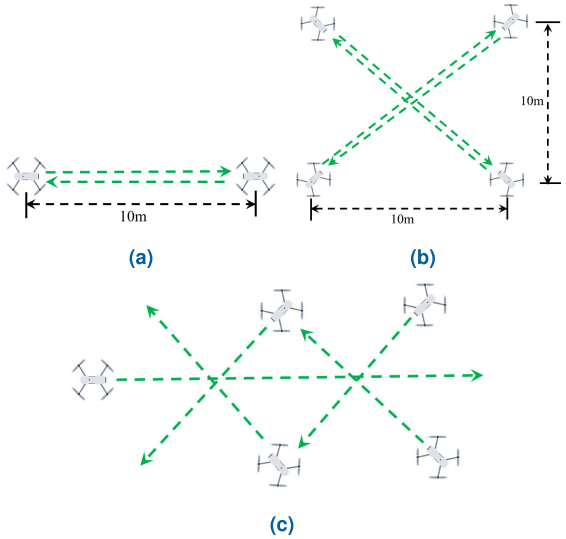

FIGURE 7. Three distinct experimental scenarios designed to assess the algorithm's performance. (a) Scenario1, (b) Scenario2, (c) Scenario3.

conjunction with the Robot Operating System (ROS) [28]. The simulator incorporates a physics engine that emulates real-world dynamics, while also offering sensor models capable of introducing Gaussian noise to mimic real-world conditions. The simulation employs the 3DR Iris drone model, and the drone's control software utilizes the PX4 flight stack, which is a prevalent tool in drone research. The loading process for the PX4 flight stack and Iris models in the Gazebo simulation is detailed in [29]. For drone control within the ROS ecosystem, Mavros, a ROS application, is utilized to establish a connection with the PX4 flight stack [30]. The simulation was conducted on a computer equipped with a 10th generation Intel i7 CPU, 48GB of RAM, and without a dedicated GPU.

In Fig. 5, a formation of ten Iris drones is employed, and the ACACT algorithm was applied to the simulation. In the simulation, the drones maintain a formation centered around a reference drone, with the remaining nine drones rotating counterclockwise while adhering to the formation. A flight video showcasing the formation flight is available in [31]. Fig. 6 depicts the software architecture of the simulation. The algorithm proposed is amenable to both centralized and decentralized control approaches. Our developed simulator for swarm UAV formation control is available for download at [32].

A force visualization tool has been developed, as illustrated in Fig. 5, to evaluate the algorithm's efficacy during formation flight simulations [33]. This tool graphically depicts the attractive force (displayed in green), the repulsive force (displayed in red), and the target velocity (displayed in light blue), conveying both their magnitude and direction. Moreover, the tool presents the three-dimensional path followed by each UAV. Such a visual representation facilitates a deeper understanding of the workings of the proposed algorithm.

B. EXPERIMENT

A comprehensive set of experiments were carried out to assess the effectiveness of the proposed collision avoidance algorithm across various UAV contexts. Fig. 7 delineates three scenarios representative of the challenges typically encountered when UAVs undergo formation changes in swarm settings. Detailed results pertaining to Scenario1 are illustrated in Fig. 8. This figure contrasts the performance of the newly proposed ACACT algorithm with the existing APF algorithm. In this test case, two UAVs were initially stationed at points A(0, 5, 5) and B(0, −5, 5). With both UAVs navigating concurrently toward opposite directions, the aim was to gauge the proficiency of each algorithm in averting collisions and ensuring that the drone adheres to the most direct trajectory while reaching the target position quickly.

The image in Fig. 8 displays the flight paths and the proximity between the drones. The space between drones was measured 30 times a second. When they came less than 2 m apart, it was counted as a collision. Results indicate that the older PF method led to several collisions, evident in Fig. 8a, 8b, and 8d. These collisions occurred because the forces pushed the drones towards each other, making them follow longer paths and reducing their speeds. In contrast, the new ACACT method prevented these collisions and chose clearer paths, leading to faster destination arrivals.

The ACACT algorithm consistently exhibited superior performance in collision avoidance compared to other established methods. The comprehensive analysis of experimental results was conducted using the PTTR metric, as introduced in Section III-D. Fig. 9 displays the efficacy of collision avoidance path planning based on this metric. As previously discussed, a PTTR value closer to 1 indicates a more efficient UAV performance in both collision avoidance and expedited target approach. Within the figure, the term "offset" delineates the deviation from the optimal tuning point and indicates the extent to which the PID gain deviates from this optimal point. Typically, as the PID gain diminishes, so does its collision avoidance capability. Hence, a rising offset correlates with a declining PTTR. Conversely, when the PID gain is elevated, the resultant path tends to elongate, leading to a prolonged duration to reach the target, which in turn causes the PTTR to decline. Across all examined scenarios, the ACACT algorithm consistently outshined its counterparts.

The second experiment, similar to Scenario1, becomes more challenging with the inclusion of four UAVs. UAV No. 1 transitions from A(5, 5, 5) to D(−5, −5, 5), while UAV No. 2 goes from B(5, −5, 5) to C(−5, 5, 5). UAVs 3 and 4 navigate from D to A and from C to B, respectively. Commands for all UAV movements are issued concurrently. Conditions were deliberately made challenging by setting the k_{pp} extremely low, as illustrated in Fig. 10, or by having high obstacle speeds, as shown in Fig. 11, increasing the likelihood of collisions.

Fig. 12 provides a multi-angle view of the Scenario2 results. Notably, the path length chosen by the proposed

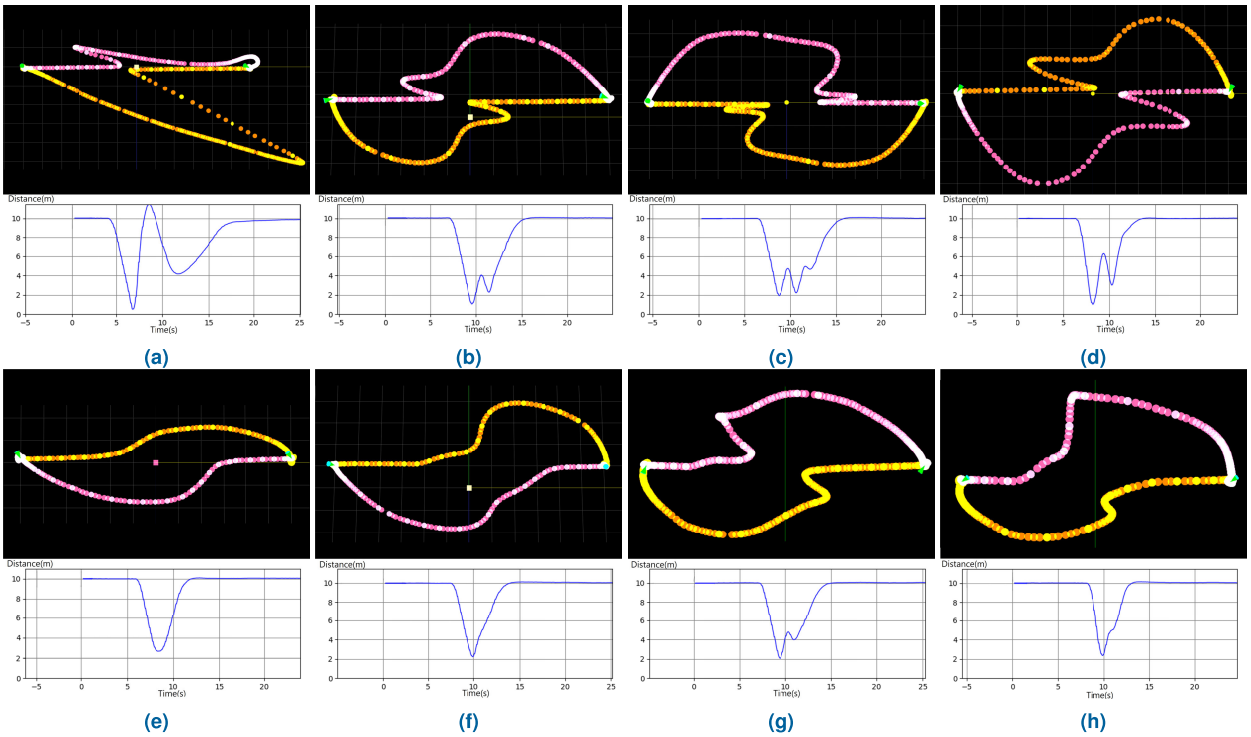

FIGURE 8. For Scenario1: Comparison of performance of (a), (b), (c), (d) with conventional algorithm and (e), (f), (g), (h) with ACACT algorithm in the four cases where the gain k_{pp} is 0.24, 0.8, 1.5 with the maximum speed of the UAV is 3 m/s and k_{pp} is 1.5 with 5 m/s in potential field-based collision avoidance.

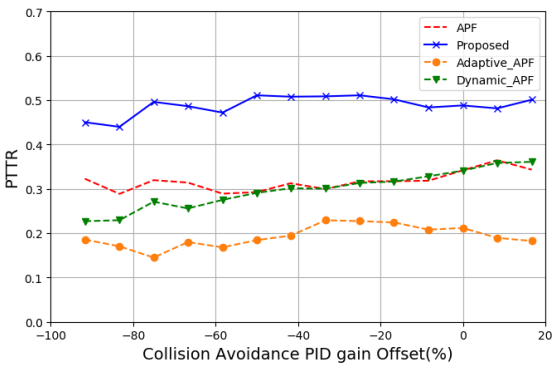

FIGURE 9. For Scenario1: Evaluation based on the proposed average PTTR metric for each algorithm and situation.

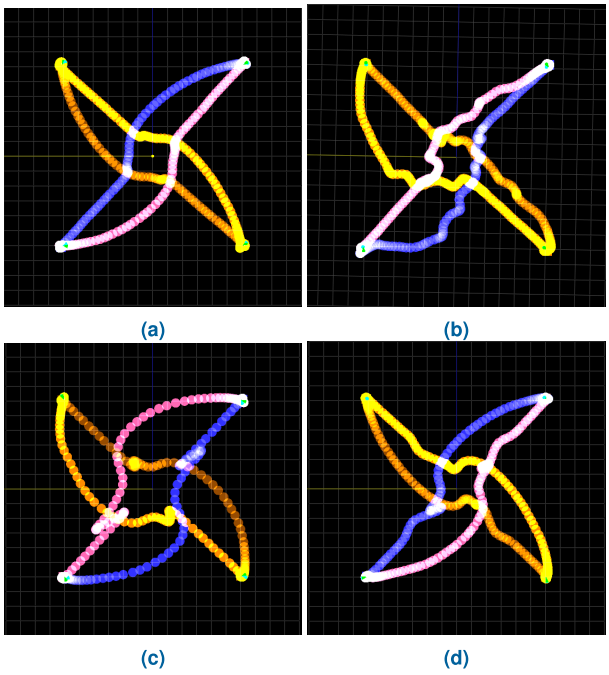

FIGURE 10. Top view of the Scenario2 experimental path for four algorithms, with the PID gain k_p set to an extremely low value (k_{pp} = 0.5). (a) APF, (b) Adaptive APF, (c) Dynamic APF, (d) Proposed.

algorithm, as seen in the left of Fig. 12a, is marginally longer than other algorithms. This implies a preference for slightly extended routes to ensure safe collision avoidance. Despite the longer path, due to the algorithm's aim to minimize deceleration while evading collisions, the destination is reached much quicker than with other algorithms, as evident in Fig. 12b. When assessed using the proposed metric, which gauges both collision avoidance and rapid target attainment, the performance of the proposed algorithm surpasses others in all situations, as shown in Fig. 12c.

In the third experiment, a UAV moving in a straight line faces multiple obstacles intermittently charging towards it. UAV No. 1 transitions from A(−17, 0, 5) to B(7, 0,5). UAV No. 2 moves from C(−5, 5, 5) to D(−15, −5, 5), while UAV No. 3 travels from E(−5, −5, 5) to F(−15, 5, 5). Approximately 2.5 to 3 seconds later, UAV No. 4 starts from G(5, 5, 5) to reach E, and UAV No. 5 sets off from H(5, −5, 5) targeting C. As with Scenario2, conditions were made more

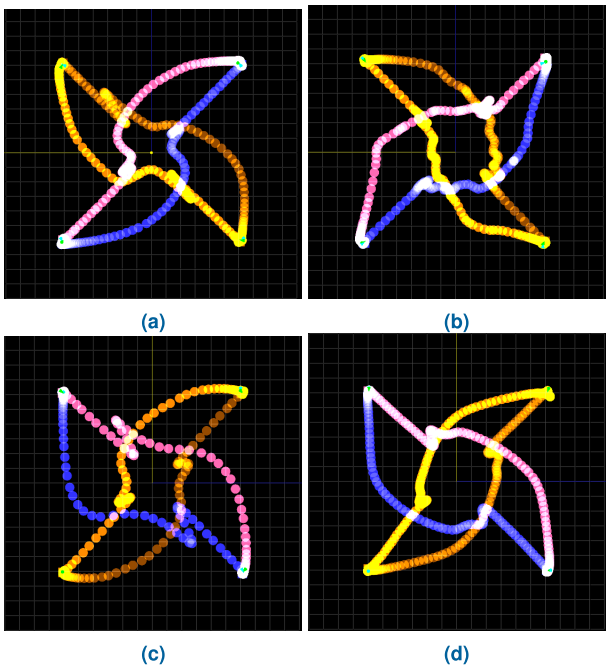

FIGURE 11. Top view of the Scenario2 experimental path for four algorithms, under conditions where the obstacles' velocities and the UAV's maximum velocity exceed anticipated values (5m/s). (a) APF, (b) Adaptive APF, (c) Dynamic APF, (d) Proposed.

TABLE 1. PID gains and parameters used in the experiment.

Scenario	Case	Max. Vel. (m/s)	k_{pp}	k_{pv}	Safety time (s)
1	(a)	3	0.24	0	0
	(b)	3	0.8	0	0
	(c)	3	1.5	0	0
	(d)	5	1.5	0	0
	(e)	3	0.24	0.3	1
	(f)	3	0.8	0.3	1
	(g)	3	1.5	0.3	1
	(h)	5	1.5	0.3	2.5
2	(a)	3	0.5	0	0
	(b)	3	0.5	0.3	0
	(c)	3	0.5	0.3	0
	(d)	3	0.5	0.3	2
	(a)	5	1.3	0	0
	(b)	5	1.3	0.3	0
	(c)	5	1.3	0.3	0
	(d)	5	1.3	0.3	2
3	(a)	3	0.5	0	0
	(b)	3	0.5	0.3	0
	(c)	3	0.5	0.3	0
	(d)	3	0.5	0.3	2
	(a)	5	1.3	0	0
	(b)	5	1.3	0.3	0
	(c)	5	1.3	0.3	0
	(d)	5	1.3	0.3	2

challenging by setting k_{pp} very low, as depicted in Fig. 13, or by elevating the speed of the obstacles, as seen in Fig. 14.

Fig. 15 displays the results of Scenario3 from different perspectives. Mirroring the findings from Scenario2, the path determined by the proposed algorithm, as shown in the left side of Fig. 15a, is marginally elongated. Nevertheless, it facilitates the fastest arrival at the target position, and its PTTR score significantly surpasses those of the other algorithms.

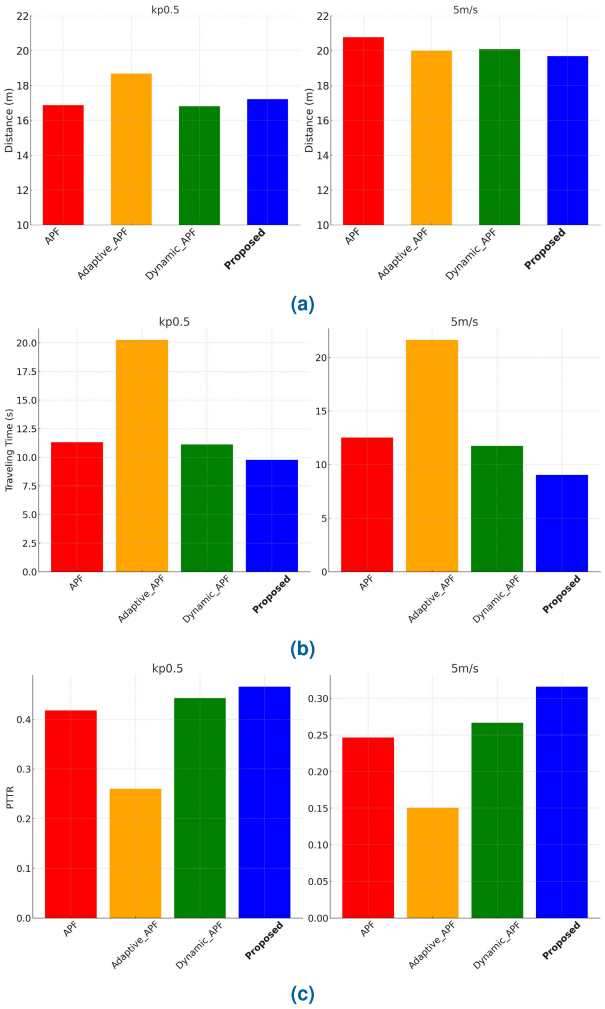

FIGURE 12. For Scenario2: (a) Average traveling distance for each algorithm under varied situations; (b) Corresponding average traveling time; (c) Evaluation based on the proposed average PTTR metric for each algorithm and situation.

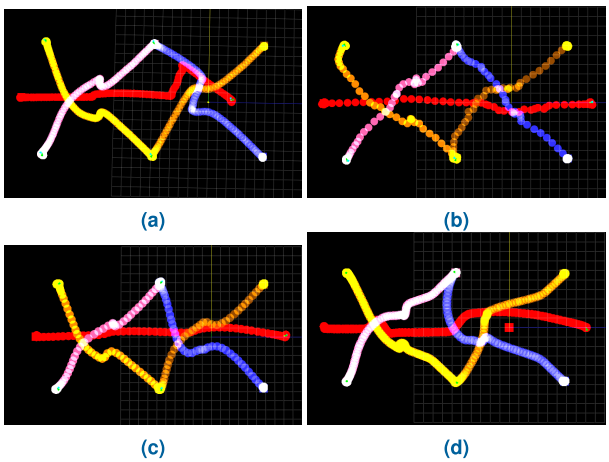

FIGURE 13. Top view of the Scenario3 experimental path for four algorithms, with the PID gain k_p set to an extremely low value (k_{pp} = 0.5). (a) APF, (b) Adaptive APF, (c) Dynamic APF, (d) Proposed.

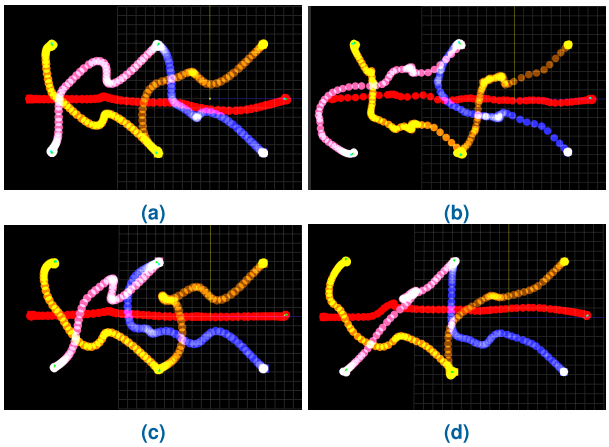

FIGURE 14. Top view of the Scenario3 experimental path for four algorithms, under conditions where the obstacles' velocities and the UAV's maximum velocity exceed anticipated values (5m/s). (a) APF, (b) Adaptive APF, (c) Dynamic APF, (d) Proposed.

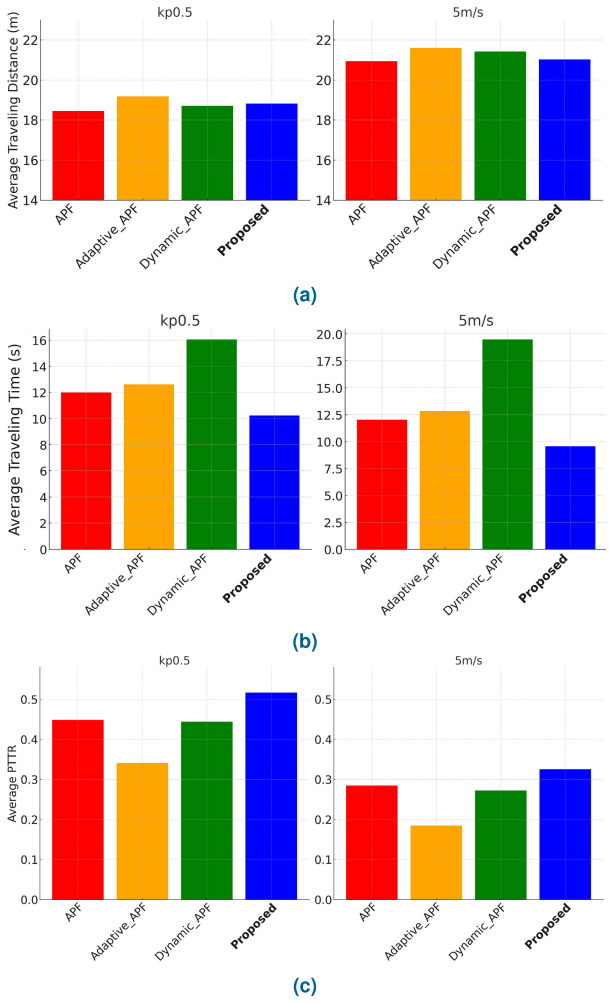

FIGURE 15. For Scenario3: (a) Average traveling distance for each algorithm under varied situations; (b) Corresponding average traveling time; (c) Evaluation based on the proposed average PTTR metric for each algorithm and situation.

The parameters employed in the experiments are enumerated in Table 1. Parameters consistent across experiments are excluded from the table for clarity. The sensing range, denoted as r_s, was configured at 7m. Regarding the PID gains, k_{ip} was established at 0.1, while k_{dp} was set at 0.4. As variations in k_{dp} and k_{ip} exert a negligible influence on path determination, the experiments predominantly centered on modifications in k_{pp}.

In the case of the conventional algorithm, it was observed that when the maximum speed was set to 5 m/s, the overall PTTR value was lower compared to the case with a speed of 3 m/s. This can be attributed to the increased inertial force acting on the drone when it avoids collisions at higher speeds, resulting in longer travel distances and a higher likelihood of passing through the collision zone.

Furthermore, for the conventional algorithm, noticeable differences were observed between cases where collision avoidance PID gains were appropriately set and cases where they were not. However, in the proposed ACACT algorithm, the performances of these two cases were almost identical, with a PTTR value of approximately 0.5. These results suggest that the ACACT algorithm consistently delivered excellent performance, regardless of the specific settings of the collision avoidance PID gains.

In terms of the CTR, a value of 0 was recorded for all cases of the ACACT algorithm (except vel5), as indicated in Fig. 9. This indicates that the proposed algorithm effectively avoids collisions, and no obstacles enter the collision area of the UAV. When the maximum speed is set to 5 m/s, the CTR values are comparable between the proposed and conventional algorithms. However, the proposed algorithm exhibits improved Travel Time Ratio (TTR) and overall performance.

V. CONCLUSION

This research presents an advanced collision avoidance algorithm based on estimated collision time, addressing the limitations of the conventional potential field method. The algorithm calculates the estimated collision time using the projected velocity. This allows unmanned aerial vehicles (UAVs) to adjust their direction to avoid collisions without slowing down, provided the estimated collision time remains below a predefined safety threshold. This approach significantly improves the UAV's path smoothness and reduces unnecessary energy consumption, resulting in faster arrival at the target position. Extensive experiments are conducted in a simulated environment to assess the efficacy of the proposed algorithm. A novel performance metric called Path Traveling Time Ratio (PTTR) is introduced to compare and analyze the performance of the proposed algorithm against existing approaches. The experimental results reveal an average PTTR improvement of 0.126, representing a maximum 20% enhancement in collision avoidance performance when considering the full metric range. The simulations, incorporating a physics engine that closely emulates real-world conditions, demonstrate the algorithm's practical applicability and offer an effective means for its evaluation.

REFERENCES

[1] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles*, vol. 1. Cham, Switzerland: Springer, 2015.

[2] N. Zunli, Z. Xuejun, and G. Xiangmin, "UAV formation flight based on artificial potential force in 3D environment," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 5465–5470.

[3] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Cooperative coverage path planning for visual inspection," *Control Eng. Pract.*, vol. 74, pp. 118–131, May 2018.

[4] S. S. Mansouri, C. Kanellakis, D. Kominiak, and G. Nikolakopoulos, "Deploying MAVs for autonomous navigation in dark underground mine environments," *Robot. Auto. Syst.*, vol. 126, Apr. 2020, Art. no. 103472.

[5] S. S. Mansouri, C. Kanellakis, E. Fresk, B. Lindqvist, D. Kominiak, A. Koval, P. Sopasakis, and G. Nikolakopoulos, "Subterranean MAV navigation based on nonlinear MPC with collision avoidance constraints," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9650–9657, 2020.

[6] R. Song, Y. Liu, and R. Bucknall, "Smoothed A* algorithm for practical unmanned surface vehicle path planning," *Appl. Ocean Res.*, vol. 83, pp. 9–20, Feb. 2019.

[7] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Robot. Auto. Syst.*, vol. 86, pp. 13–28, Dec. 2016.

[8] J. J. Kuffner Jr. and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2000, vol. 2, pp. 995–1001.

[9] M. Kothari and I. Postlethwaite, "A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees," *J. Intell. Robotic Syst.*, vol. 71, no. 2, pp. 231–253, Aug. 2013.

[10] Y. Lin and S. Saripalli, "Sampling-based path planning for UAV collision avoidance," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3179–3192, Nov. 2017.

[11] Z. Chao, L. Ming, Z. Shaolei, and Z. Wenguang, "Collision-free UAV formation flight control based on nonlinear MPC," in *Proc. Int. Conf. Electron., Commun. Control (ICECC)*, Sep. 2011, pp. 1951–1956.

[12] T. J. Stastny, A. Dash, and R. Siegwart, "Nonlinear MPC for fixed-wing UAV trajectory tracking: Implementation and flight experiments," in *Proc. AIAA Guid. Navigat. Control Conf.*, 2017, p. 1512.

[13] J. Seo, Y. Kim, S. Kim, and A. Tsourdos, "Collision avoidance strategies for unmanned aerial vehicles in formation flight," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 6, pp. 2718–2734, Dec. 2017.

[14] H. L. N. N. Thanh and S. K. Hong, "Completion of collision avoidance control algorithm for multicopters based on geometrical constraints," *IEEE Access*, vol. 6, pp. 27111–27126, 2018.

[15] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Trans. Robot. Autom.*, vol. 8, no. 1, pp. 23–32, Jun. 1992.

[16] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Trans. Robot. Autom.*, vol. 16, no. 5, pp. 615–620, Jun. 2000.

[17] B. Lu, G. Li, H. Yu, H. Wang, J. Guo, D. Cao, and H. He, "Adaptive potential field-based path planning for complex autonomous driving scenarios," *IEEE Access*, vol. 8, pp. 225294–225305, 2020.

[18] X. Zhu, Y. Liang, and M. Yan, "A flexible collision avoidance strategy for the formation of multiple unmanned aerial vehicles," *IEEE Access*, vol. 7, pp. 140743–140754, 2019.

[19] A. C. Woods and H. M. La, "A novel potential field controller for use on aerial robots," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 4, pp. 665–676, Apr. 2019.

[20] H. Sun, J. Qi, C. Wu, and M. Wang, "Path planning for dense drone formation based on modified artificial potential fields," in *Proc. 39th Chin. Control Conf. (CCC)*, 2020, pp. 4658–4664.

[21] A. Singletary, K. Klingebiel, J. Bourne, A. Browning, P. Tokumaru, and A. Ames, "Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2021, pp. 8129–8136.

[22] M. A. Johnson and M. H. Moradi, *PID Control*. Cham, Switzerland: Springer, 2005.

[23] S. P. Drake, "Converting GPS coordinates [ϕ, λ,h] to navigation coordinates (ENU)," Surveill. Syst. Division Electron. Surveill. Res. Lab., Edinburgh, SA, Australia, Tech. Rep. DSTO-TN-0432, 2002.

[24] D. H. Kim, H. Wang, and S. Shin, "Decentralized control of autonomous swarm systems using artificial potential functions: Analytical design guidelines," *J. Intell. Robot. Syst.*, vol. 45, no. 4, pp. 369–394, 2006.

[25] M. A. Toksöz, S. Oguz, and V. Gazi, "Decentralized formation control of a swarm of quadrotor helicopters," in *Proc. IEEE 15th Int. Conf. Control Autom. (ICCA)*, Jul. 2019, pp. 1006–1013.

[26] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, Mar. 1986.

[27] Y. Du, X. Zhang, and Z. Nie, "A real-time collision avoidance strategy in dynamic airspace based on dynamic artificial potential field algorithm," *IEEE Access*, vol. 7, pp. 169469–169479, 2019.

[28] *Gazebo Homepage*. Accessed: Oct. 29, 2023. [Online]. Available: http://gazebosim.org/

[29] *Iris Model With PX4*. Accessed: Oct. 29, 2023. [Online]. Available: https://docs.px4.io/v1.12/en/simulation/gazebo.html

[30] *Mavros for Connecting With the PX4 Flight Stack in the ROS Ecosystem*. Accessed: Oct. 29, 2023. [Online]. Available: https://github.com/mavlink/mavros

[31] S. Min. *The Swarm Drones Formation Flight Simulation*. Accessed: Oct. 29, 2023. [Online]. Available: https://youtu.be/1PBxLBN2fe0

[32] *The Swarm UAVs Formation Control With Collision Avoidance Simulator*. Accessed: Oct. 29, 2023. [Online]. Available: https://github.com/ICSL-hanyang/swarm_ctrl_pkg/tree/adaptive_PF_PathPlanning_working

[33] *The Forces Visualization Tool for Swarm UAVs*. [Online]. Available: https://github.com/ICSL-hanyang/vehicle_monitor

SEWOONG MIN received the bachelor's degree in electronic engineering from Hanyang University, Ansan, South Korea, in 2015, where he is currently pursuing the Ph.D. degree. He is a Researcher specializing in robot path planning, swarm UAVs path planning, and reinforcement learning. Throughout his academic journey, he was a recipient of a prestigious scholarship from the Korean government, which has supported his studies during both the bachelor's and Ph.D. programs.

HAEWOON NAM (Senior Member, IEEE) received the B.S. degree from Hanyang University, Seoul, South Korea, the M.S. degree from Seoul National University, and the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin, Austin, TX, USA. From 1999 to 2002, he was with Samsung Electronics, Suwon, South Korea, where he was engaged in the design and development of code division multiple access and global system for mobile communications (GSM)/general packet radio service baseband modem processors. In Summer 2003, he was with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA, where he performed extensive radio channel measurements and analysis at 60 GHz. In Fall 2005, he was with the Wireless Mobile System Group, Freescale Semiconductor, Austin, where he was engaged in the design and test of the worldwide interoperability for microwave access (WiMAX) medium access control layer. His industry experience also includes working with the Samsung Advanced Institute of Technology, Giheung, South Korea, where he participated in the simulation of multi-input-multi-output systems for the third-generation partnership project (3GPP) long-term evolution (LTE) standard. In October 2006, he joined the Mobile Devices Technology Office, Motorola Inc., Austin, where he was involved in algorithm design and development for the 3GPP LTE mobile systems, including modeling of 3GPP LTE modem processor. In 2010, he was with Apple Inc., Cupertino, CA, USA, where he worked on the research and development of next-generation smart mobile systems. Since March 2011, he has been with the Division of Electrical Engineering, Hanyang University, Ansan, South Korea, where he is currently a Professor. He received the Korean Government Overseas Scholarship for the Ph.D. degree in electrical engineering.