A Real-Time RRT-APF Approach for Efficient Multi-Robot Navigation in Complex Environments

Kaihuai Zhang, Mohsen Zahmatkesh, Martin Stefanec, Farshad Arvin, *Senior Member, IEEE*, and Junyan Hu, *Senior Member, IEEE*

***Abstract*—This paper proposes a real-time multi-robot navigation method that integrates the Rapidly-exploring Random Tree (RRT) algorithm with the improved Artificial Potential Field (APF) approach. Since traditional path planning methods often face problems such as generating non-smooth paths and inefficient obstacle avoidance in changing environments, the RRT algorithm is used for initial path planning to pass through obstacles. Aiming to obtain a smooth collision-free path, Catmull-Rom spline smoothing is then introduced, which smooths the initially obtained trajectory and ensures that the curvature of the trajectory remains continuous. By combining the improved APF method, networked robots can then achieve safe navigation and effective obstacle avoidance in dynamic environments. The effectiveness of the proposed RRT-APF method is verified by both simulations and hardware experiments using real micro unmanned aerial vehicles.**

I. Introduction

Autonomous robots offer significant advantages across many areas, but their use in dynamic and unstructured environments remains a major challenge [1]. Developing robust coordination approaches for mobile robots could provide a solution to this problem. To address such natural and complex conditions, the integration of path planning and collision avoidance is essential to increase the efficiency and safety of route planning and decision-making. Such methods can be applied to many real-world applications, such as observations of bee hives [2], [3] and formation of multiple unmanned aerial vehicles (UAVs) [4], [5], which motivates researchers to develop more advanced algorithms for better practical performance.

The Rapidly-exploring Random Tree (RRT) algorithm is one of the most commonly used sampling-based methods for path planning in dynamic environments [6]. RRT does not require modeling of the planning space and can quickly generate feasible paths by randomly exploring the search space, demonstrating strong search capabilities [7]. However, due to its lack of motion constraints, the paths generated by RRT often have unsmooth points, which poses a problem for robot swarms that generally require smooth movements [8], [9]. The lack of smoothness can lead to jerky motions and increased wear and tear on the robotic platforms, making it essential to incorporate path smoothing techniques [10]. To address the smoothness issue, various path smoothing techniques have been proposed. Bezier curves and B-splines are commonly used for this purpose, but they often lack the local control and natural transitions needed for complex environments [11].

Collision avoidance is another critical aspect of multi-robot systems. The artificial potential field (APF) method is a popular approach for collision avoidance, where robots can be treated as particles moving under the influence of artificial forces [12]. The APF method generates attractive forces towards the goal and repulsive forces away from obstacles, guiding the robots along a collision-free path. However, the traditional APF method has several limitations, such as local minima and poor real-time performance [13]. Local minima can trap robots in positions where the net force is zero, preventing them from reaching their goals [14].

This work was partially supported by EU H2020 RoboRoyale [964492] and Horizon Europe Sensorbees [101130325] projects.

K. Zhang, M. Zahmatkesh, F. Arvin and J. Hu are with the Department of Computer Science, Durham University, DH1 3LE, Durham, UK. (e-mail:{kaihuai.zhang; mohsen.zahmatkesh; farshad.arvin; junyan.hu}@durham.ac.uk)

M. Stefanec is with the Department of Zoology, Institute of Biology, University of Graz, Graz, Austria (e-mail:martin.stefanec@uni-graz.at)

II. Methodology

The goal of this paper is to develop a comprehensive approach that integrates the RRT algorithm, an improved APF method, and Catmull-Rom spline smoothing to ensure that all robots move towards a common goal, achieve force equilibrium, and effectively avoid active obstacles.

A. Smoothing RRT Path Planning with Catmull-Rom Splines

The first step is to generate an initial path for each robot by using the RRT algorithm. The RRT algorithm works by exploring the area around the starting point and builds a tree of possible paths leading to the goal [15]. Here's how the RRT path planning process works:

Algorithm 1 RRT-Based Path Planning

```
Start the tree with the robot's initial position.
for each step do
    Randomly pick a point in the search area.
    Find the closest node in the tree to this point.
    Extend the tree from the nearest node towards the picked point.
    if goal position is reached then
        Stop the process.
    end if
end for
Return the path from the start to the goal.
```

The RRT algorithm can be mathematically described as follows. Let $\mathcal{X}$ be the search space, x_{init} be the initial position, and x_{goal} be the goal position. The tree T is initialized with the root node x_{init}. At each iteration, a random sample x_{rand}

is drawn from $\mathcal{X}$. The nearest node x_{nearest} in T to x_{rand} is found using the distance metric $\|x_{\text{rand}} - x_{\text{nearest}}\|$. A new node x_{new} is then created by moving from x_{nearest} towards x_{rand} by a step size ϵ. The process continues until x_{goal} is reached or a maximum number of iterations is exceeded [16].

$$x_{\text{new}} = x_{\text{nearest}} + \epsilon \frac{x_{\text{rand}} - x_{\text{nearest}}}{\|x_{\text{rand}} - x_{\text{nearest}}\|} \tag{1}$$

To find the closest node, we use a measure of distance, which is calculated as follows:

$$\|x_{\text{rand}} - x_{\text{nearest}}\| = \left(\sum_{i=1}^{n} (x^i_{\text{rand}} - x^i_{\text{nearest}})^2 \right)^{1/2} \tag{2}$$

As the tree grows, the probability of sampling the goal position increases. This means that eventually, the algorithm will find a path to the destination [17].

Once the initial path is generated using the RRT algorithm, it is often not smooth and may contain sharp turns. To address this, we apply Catmull-Rom spline interpolation to smooth the path.

Catmull-Rom splines, a type of interpolating spline, are chosen for their simplicity and efficiency. Unlike other spline methods, Catmull-Rom splines do not require solving equations, making them computationally efficient. This is especially useful in real-time applications where computing power might be limited. Catmull-Rom splines make sure the path goes through all control points, which is important for keeping the planned path in multi-robot systems [18].

The Catmull-Rom spline is defined by a set of control points $\{P_i\}$. For a given segment between control points P_i and P_{i+1}, the spline is given by:

$$\begin{aligned} P(t) = 0.5\big[&(2P_i) + (-P_{i-1} + P_{i+1})t \\ &+ (2P_{i-1} - 5P_i + 4P_{i+1} - P_{i+2})t^2 \\ &+ (-P_{i-1} + 3P_i - 3P_{i+1} + P_{i+2})t^3\big], \end{aligned} \tag{3}$$

where t is the parameter that varies from 0 to 1 along the segment [19].

The smoothed path ensures that the robots can move smoothly and efficiently towards their goals.

B. Collision Avoidance with Improved APF

While following the smoothed path, the robots must avoid collisions with obstacles and other robots. The APF method uses the second point from the smoothed path as the goal point to calculate attractive and repulsive forces, guiding the robot in real-time. The traditional APF method generates attractive forces towards the goal and repulsive forces away from obstacles, guiding the robots along a collision-free path [12]. However, the traditional APF method has the following limitations, which could be overcome by the proposed design:

1) High Attractive Force at a Distance: When far from the goal, the attractive force is very intense, while the repulsive force from obstacles is relatively minor. This imbalance can lead to collisions as the robot is less pushed away from obstacles. To address this, we modify the attractive potential function. The improved attractive potential field $U_{\text{att}}(q)$ is given by:

$$U_{\text{att}}(q) = \begin{cases} \frac{1}{2} K_{\text{att}} \|q - q_g\|^2, & \text{if } \|q - q_g\| \le d \\ d K_{\text{att}} \|q - q_g\| - \frac{1}{2} K_{\text{att}} d^2, & \text{if } \|q - q_g\| > d \end{cases} \tag{4}$$

where K_{att} is the attractive scale factor, q is the robot's position, q_g is the goal position, and d is a specific distance called the distance factor.

The corresponding attractive force $F_{\text{att}}(q)$ is:

$$F_{\text{att}}(q) = -\nabla U_{\text{att}}(q) = \begin{cases} K_{\text{att}}(q_g - q), & \text{if } \|q - q_g\| \le d \\ -d K_{\text{att}} \frac{q - q_g}{\|q - q_g\|}, & \text{if } \|q - q_g\| > d \end{cases} \tag{5}$$

2) Local Minima Problem: The traditional APF method can get stuck in local minima, where the robot might be unable to find a path to the goal within the allowed iterations. To prevent robots from getting trapped in local minima, an additional random perturbation is introduced. This is achieved by periodically adding a small random value to the force components, helping the robot to escape from local minima. The random perturbation can be represented as:

$$F_{\text{perturb},x} = F_{\text{att},x} + (rand() - 0.5) * 0.1, \tag{6}$$

$$F_{\text{perturb},y} = F_{\text{att},y} + (rand() - 0.5) * 0.1, \tag{7}$$

where $rand()$ generates a random number between 0 and 1.

3) Difficulties Near Obstacles: When obstacles are near the goal, the attractive force decreases and the repulsive force increases. This can make it hard for the robot to reach the goal as strong repulsive forces may push it away, even when it's close. To address this, we modify the repulsive potential function. The improved repulsive potential field $U_{\text{rep}}(q)$ is given by:

$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2} \eta \left(\frac{1}{\rho(q, q_{\text{obs}})} - \frac{1}{\rho_0} \right)^2 \rho^n(q, q_g), & \text{if } \rho(q, q_{\text{obs}}) \le \rho_0 \\ 0, & \text{if } \rho(q, q_{\text{obs}}) > \rho_0 \end{cases} \tag{8}$$

where η is a positive constant, $\rho(q, q_{\text{obs}})$ is the distance to the obstacle, ρ_0 is the influence distance, q_g is the goal position, and n is a positive number (typically $n = 2$).

The corresponding repulsive force $F_{\text{rep}}(q) = -\nabla U_{\text{rep}}(q)$ is:

$$\text{if } \rho(q, q_{\text{obs}}) \le \rho_0 :$$

$$a_1 \eta \,(A)\, \frac{\rho^n(q, q_g)}{\rho^2(q, q_{\text{obs}})} + a_2 \frac{n}{2} \eta \,(A)^2 \cdot \rho^{n-1}(q, q_g), \tag{9}$$

$$\text{if } \rho(q, q_{\text{obs}}) > \rho_0 : \quad 0, \tag{10}$$

where

$$\begin{aligned} A &= \frac{1}{\rho(q, q_{\text{obs}})} - \frac{1}{\rho_0}, \\ a_1 &= \nabla \rho(q, q_{\text{obs}}) = \frac{q - q_{\text{obs}}}{\|q - q_{\text{obs}}\|}, \\ a_2 &= -\nabla \rho(q, q_g) = -\frac{q - q_g}{\|q - q_g\|}. \end{aligned}$$

Additionally, to ensure collision avoidance among robots, each robot is also considered as a dynamic obstacle to other robots. This means that the repulsive forces from neighboring

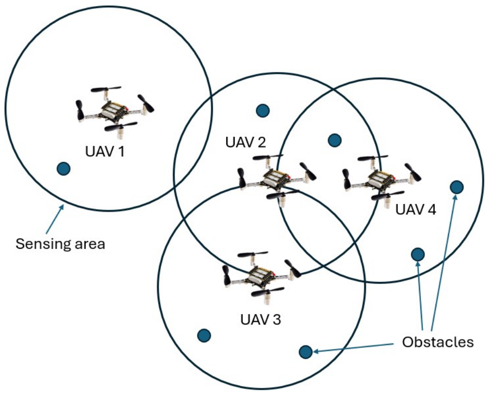


Fig. 1. The principle of the repulsive forces generated by nearby obstacles and UAVs within sensing range.

TABLE I
SIMULATION PARAMETERS

Parameter	Low	High
Grid size	100x100	100x100
Goal pos.	(90, 90)	(85, 85)
No. of robots	3	6
Start zone	[0, 20]	[0, 30]
Safe zone	[80, 100]	[70, 100]
Obstacles	10	20
RRT iter.	1000	1000
RRT step	1 unit	1 unit
Attr. gain	0.5	0.5
Repl. gain	0.5	0.5
Infl. dist.	25 units	25 units
Step rate	0.05	0.05
Epoch	10000	10000
Force Thresh.	0.03	0.035

robots within the sensing range are also considered, which is shown in Fig. 1. This setup helps in maintaining a safe distance between robots, preventing collisions and ensuring smooth navigation.

The total force acting on each robot is the sum of the attractive force and the repulsive forces from both dynamic obstacles and other robots. The total force $F_{\text{total}}(q)$ is given by:

$$F_{\text{total}}(q) = F_{\text{att}}(q) + \sum_{i} F_{\text{rep},i}(q), \tag{11}$$

where $F_{\text{att}}(q)$ is the attractive force towards the goal, and $F_{\text{rep},i}(q)$ is the repulsive force from the i-th neighboring obstacle or robot.

These improvements enhance the overall efficiency and reliability of the APF method, making it more suitable for dynamic and complex environments.

III. SIMULATION RESULTS

In this section, the feasibility and performance of the proposed approach are verified by multiple simulations.

A. Simulation Setup

The simulation grid size was set to 100x100 units. The goal position was set in the center of the top-right corner.

1) Robot Density Scenarios: To validate the performance and applicability of our method, we conducted two experiments with different densities of robots and obstacles.

In the low robot density scenario, the starting positions of 3 robots were randomly generated within a defined start zone ranging from [0, 20]. In the low robot density scenario, 10 dynamic obstacles were randomly placed. These obstacles were positioned in such a way that they did not overlap with the robots' goal's safe zone. The safe zone was defined as [80, 100]. Both obstacles and robots were generated to prevent any overlap at the start, avoiding collisions at the start.

In the high robot density scenario, the starting positions of 6 robots were randomly generated within a defined start zone ranging from [0, 30]. In this scenario, 20 dynamic obstacles were randomly placed. These obstacles were positioned such that they did not overlap with the robots' goal's safe zone. The safe zone was defined as [70, 100]. Both obstacles and robots were generated to prevent any overlap at the start, avoiding collisions at the start. It is also reasonable that the start zone and safe zone are larger in the high-density scenario for the increased number of robots and obstacles.

Within the safe zone, the repulsive forces from non-robot obstacles were ignored, but the repulsive forces from other robots were still considered. This ensured that robots could navigate safely within the safe zone without being affected by dynamic obstacles, while still avoiding collisions with each other.

To escape local minima, each robot will also receive small random perturbations.

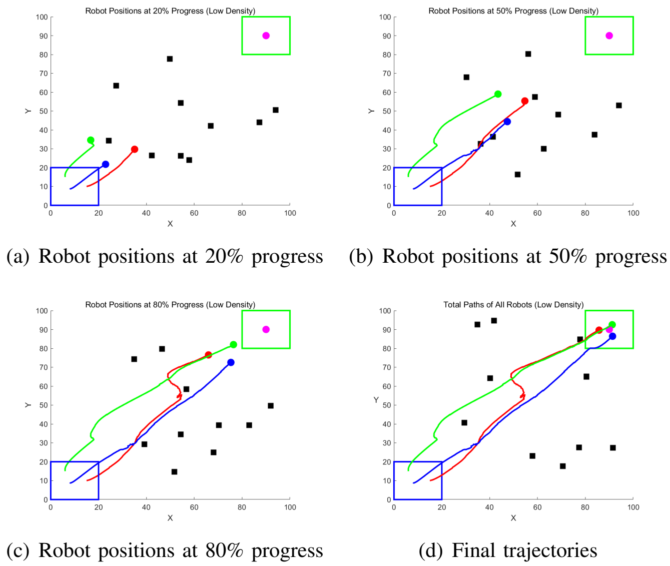
2) Multi-Robot Attractive-Repulsive Force Balance Control: To save computational resources and improve efficiency, we introduced the concept of a force equilibrium threshold. When the resultant force on each robot is less than this force equilibrium threshold, the robots are considered to be in a state of force equilibrium.

Multiple robots share a common goal, and the simulation ends when the robots reach a state of force equilibrium. The attractive force is reduced and the repulsive force between robots is increased when they enter the safe zone. This adjustment ensures that the robots maintain a certain distance from each other and form a stable geometric layout, such as a triangle, near the goal.

3) Simulation Parameters: The simulation parameters used in both the low and high density scenarios are summarized in Table I. These parameters include the grid size, goal position, RRT algorithm settings, and the specific gains for the attractive and repulsive forces in the artificial potential field method. The chosen parameters ensure a consistent and fair evaluation of the proposed method across different scenarios.

B. Robot Positions at Different Progress Points

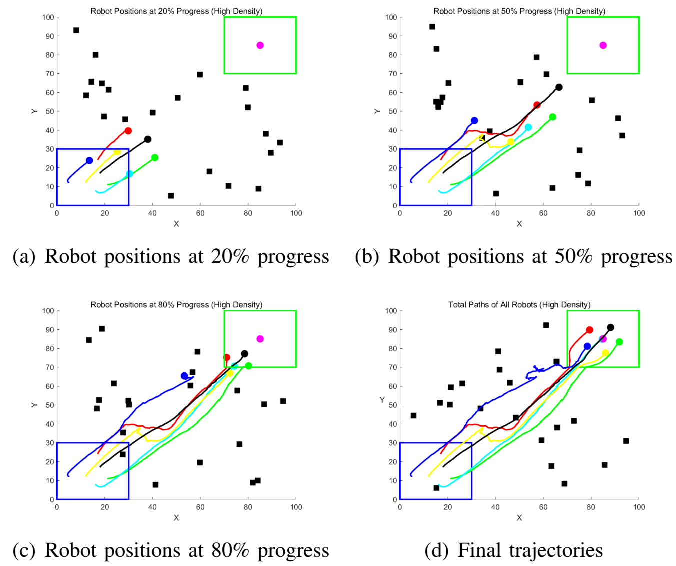
Figure 2 shows the positions of the robots at 20%, 50%, and 80% of the total journey, as well as the total paths taken by all robots from their initial positions to the goal, for the low



(a) Robot positions at 20% progress (b) Robot positions at 50% progress

(c) Robot positions at 80% progress (d) Final trajectories

Fig. 2. Robot positions and trajectories under low-density moving obstacles.



(a) Robot positions at 20% progress (b) Robot positions at 50% progress

(c) Robot positions at 80% progress (d) Final trajectories

Fig. 3. Robot positions and trajectories under high-density moving obstacles.

robot density. These figures illustrate how the robots navigate through the environment and avoid obstacles. Note that the positions of the obstacles change over time due to the dynamic nature of the environment, which is reflected in the different progress points.

From these figures, it is evident that the robots are able to navigate towards the goal while avoiding obstacles effectively. The positions at different progress points show a clear trajectory towards the goal, indicating the effectiveness of the proposed method in guiding the robots. At 20% progress, the robots have just started their journey and are beginning to navigate around the obstacles. By 50% progress, the robots have successfully avoided several obstacles and are halfway to the goal. At 80% progress, the robots are nearing the goal, demonstrating the method's ability to maintain a clear path even as the robots approach their destination. The total paths taken by all robots provide a comprehensive view of the entire journey, illustrating the effectiveness of the proposed method in guiding multiple robots through the environment while avoiding obstacles. The dynamic nature of the environment causes the positions of the obstacles to change over time, which is why the obstacles' positions differ at each progress point.

C. Effect of Increased Density

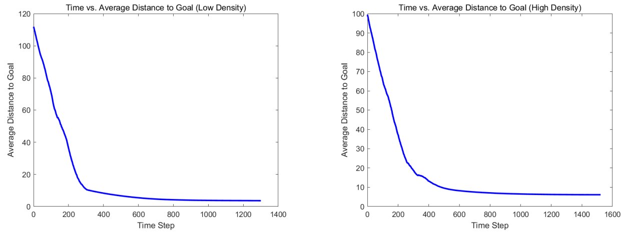
To evaluate the robustness of the proposed method under higher robot density, we increased the number of robots and obstacles. Figure 3 shows the positions of the robots at 20%, 50%, and 80% of the total journey, as well as the total paths taken by all robots in the high density scenario. Similar to the low density scenario, the positions of the obstacles change over time due to the dynamic nature of the environment.

The high density scenarios demonstrate that the proposed method can handle higher robot densities effectively. The robots maintain a clear trajectory towards the goal, and the avoidance of obstacles remains effective. This indicates the scalability of the method in more complex environments with higher robot densities. At 20% progress, the robots are beginning to navigate the denser environment, showing initial adjustments to avoid collisions. By 50% progress, the robots have successfully navigated around more obstacles and are halfway to the goal, demonstrating the method's ability to handle increased complexity. At 80% progress, the robots are nearing the goal, maintaining effective obstacle avoidance and clear paths despite the higher density. The total paths taken by all robots highlight the method's ability to manage multiple robots in a more crowded environment, ensuring that each robot reaches its goal while avoiding collisions.

D. Time vs. Distance to Goal

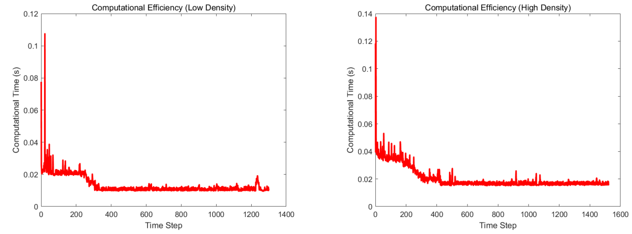
Figure 4(a) shows the relationship between time and the average distance to the goal for the robots in the low density scenario. Figure 4(b) shows the same relationship for the high density scenario.

In both scenarios, the average distance to the goal decreases over time, indicating that the robots are effectively moving towards the goal. The rate of decrease in distance is slightly slower in the high density scenario, which is expected due to the higher complexity and potential for more interactions between robots. However, the overall trend remains consistent, demonstrating the robustness of the proposed method. The time-distance graphs show that the robots are consistently reducing their distance to the goal, with a steady decline in both scenarios. The high density scenario shows a slightly slower rate of decline, reflecting the added complexity and interactions, but the overall effectiveness of the method is not compromised.



(a) (b)

Fig. 4. Time vs. average distance to goal at (a) low-density and (b) high-density scenarios.



(a) (b)

Fig. 5. Computational efficiency in (a) low-density and (b) high-density scenarios.

E. Additional Evaluation Metrics

In addition to the previously discussed performances and matrices, some additional evaluation metrics used to assess the performance of the proposed method include:

- **Computational Efficiency:** The time taken to generate and smooth the paths, as well as the time taken for real-time adjustments.
- **Path Smoothness:** The smoothness of the generated paths, measured by the number of sharp turns and changes in direction.
- **Collision Rate:** The number of collisions with obstacles and other robots during the navigation.

1) Computational Efficiency: The computational efficiency was measured by recording the computation time at each time step. The experiments were conducted on a system with a 12th Gen Intel(R) Core(TM) i5-12400 2.50 GHz CPU, using MATLAB R2022a. Figure 5(a) shows the computational efficiency for the low-density scenario, while Figure 5(b) shows the same for the high-density scenario.

In the low-density scenario, the computation time ranged from approximately 0.01 to 0.02 seconds. There were larger fluctuations when the robots encountered obstacles. However, as the robots approached their target points and began to enter a state of near force equilibrium, the computation time decreased from 0.02 to around 0.01 seconds. This shows the algorithm's efficiency and ability to manage real-time path planning.

In the high-density scenario, the computation time ranged from approximately 0.02 to 0.04 seconds. The increased complexity led to more fluctuations. Similarly, as the robots neared their target points and started to reach a state of near force equilibrium, the computation time decreased from 0.04 to around 0.02 seconds. Even with the higher density, the algorithm still achieved reasonable computation times, highlighting its robustness and efficiency.

2) Path Smoothness and Collision Counts: The path smoothness and collision counts were also evaluated. The path smoothness is measured as the average angular change along the path, and collisions are counted when the distance between robots is less than 1 unit. The minimum safe distance between robots is primarily reflected in the repulsive force calculations to ensure that robots maintain a safe distance from each other.

TABLE II
PATH SMOOTHNESS AND COLLISION COUNTS

Metric	Value
Low-Density Scenario	
Average Path Smoothness of Each robot	0.2648, 0.2497, 0.1428
Average Total Collisions	0.00
Increased-Density Scenario	
Average Path Smoothness of Each robot	0.3038, 0.2075, 0.2601, 0.1894, 0.1144, 0.0815
Average Total Collisions	0.10

TABLE III
EXPERIMENTAL TEST PARAMETERS

Parameter	Value (cm)
Flight zone size	200x200
Goal position	(170, 170)
UAV 1 initial point	(18, 42, 20)
UAV 2 initial point	(14, 10, 50)
UAV 3 initial point	(44, 10, 70)
Obstacle 1 position	(80, 80)
Obstacle 2 position	(100, 120)
Obstacle 3 position	(130, 100)

The data in Table II reveal that the average path smoothness is slightly higher in the high-density scenario, suggesting more angular changes because of the increased complexity. The average total collisions are also greater in the high-density scenario, showing a higher chance of interactions between robots. However, the low number of collisions in both scenarios shows that the proposed method successfully prevents accidents, proving that the repulsive force calculations are effective in keeping safe distances.

IV. EXPERIMENTAL VALIDATION

To assess the feasibility and effectiveness of the proposed method in practice, an experimental test is conducted using three aerial vehicles with six degrees of freedom (6DoF). The UAV platform used is Crazyflie V2.1, and the localization system is equipped with Lighthouse V2.0 base stations. The flight environment is an indoor area with three obstacles. The details of the experimental setup are illustrated in Table III.

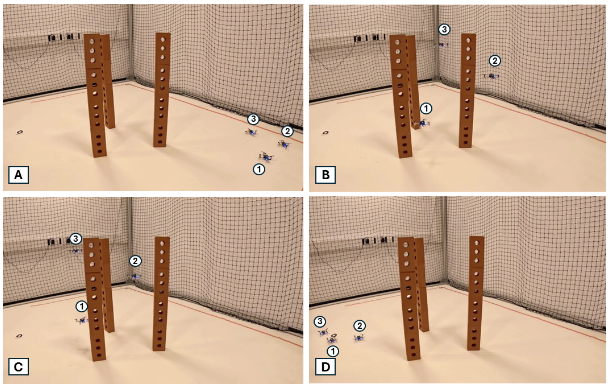


Fig. 6. Progress of the task being achieved by a team of three UAVs at different time instants.

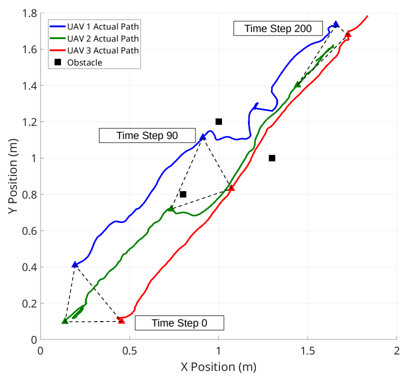


Fig. 7. Real-world flight trajectories of the UAVs during the task.

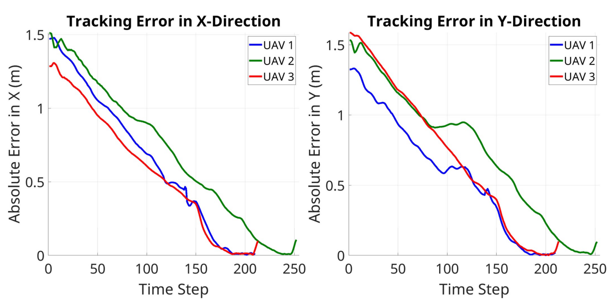


Fig. 8. Trajectory tracking error in XY directions.

Some snapshots of the experiments at different time instants are shown in Fig. 6. As illustrated in Fig. 7, UAV 1 and UAV 2 approach the obstacles at time step 90, creating smooth curves. Meanwhile, UAV 3 finds its optimal linear path without encountering any obstacles. Ultimately, all three UAVs successfully complete their optimal trajectories and reach the target point with 200 time steps. Fig. 8 shows the absolute tracking errors of the three UAVs in the X and Y directions. It is evident that all three tracking errors approach zero, indicating that the UAVs successfully reached their goal points. During the landing procedure after the completion of the task, however, two of the UAVs slightly diverge from their goal points to avoid potential collisions.

V. Conclusion

In this paper, we propose a novel method which combines RRT and an improved APF to achieve efficient multi-robot navigation in complex dynamic environments. The improved APF dynamically adjusts repulsive forces to escape local minima and improve overall efficiency. The effectiveness of the proposed method is validated by multiple simulations and real flight analysis. Future directions include optimizing the computational complexity of the planning algorithm and incorporating adaptive mechanisms to adjust the parameters based on the environment and robot dynamics.

References

[1] H. Choi, C. Crump, C. Duriez, A. Elmquist *et al.*, "On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward," *Proceedings of the National Academy of Sciences*, vol. 118, no. 1, p. e1907856118, 2021.

[2] J. Ulrich, M. Stefanec, F. Rekabi-Bana, L. A. Fedotoff *et al.*, "Autonomous tracking of honey bee behaviors over long-term periods with cooperating robots," *Science Robotics*, vol. 9, no. 95, p. eadn6848, 2024.

[3] J. Halloy, F. Mondada, S. Kernbach, and T. Schmickl, "Towards bio-hybrid systems made of social animals and robots," in *International Conference on Biomimetic and Biohybrid Systems*, 2013, pp. 384–386.

[4] K. Wu, J. Hu, Z. Li, Z. Ding, and F. Arvin, "Distributed collision-free bearing coordination of multi-uav systems with actuator faults and time delays," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[5] F. Rekabi-Bana, J. Hu, T. Krajník, and F. Arvin, "Unified robust path planning and optimal trajectory generation for efficient 3d area coverage of quadrotor uavs," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[6] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[7] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," in *Technical Report TR 98-11*. Computer Science Department, Iowa State University, 1998.

[8] H. Bae, G. Kim, J. Kim, D. Qian, and S. Lee, "Multi-robot path planning method using reinforcement learning," *Applied Sciences*, 2019.

[9] C. Agachi, F. Arvin, and J. Hu, "Rrt*-based leader-follower trajectory planning and tracking in multi-agent systems," in *2024 IEEE International Conference on Intelligent Systems*, 2024, pp. 1–6.

[10] Z. Wu, W. Su, and J. Li, "Multi-robot path planning based on improved artificial potential field and b-spline curve optimization," in *2019 Chinese Control Conference (CCC)*, 2019, pp. 4691–4696.

[11] R. T. Farouki, "The bernstein polynomial basis: A centennial retrospective," *Computer Aided Geometric Design*, vol. 29, no. 6, pp. 379–419, 2012.

[12] S. S. Ge and Y. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous robots*, 2002.

[13] X. Tong, S. Yu, G. Liu, X. Niu, C. Xia, J. Chen, Z. Yang, and Y. Sun, "A hybrid formation path planning based on a* and multi-target improved artificial potential field algorithm in the 2d random environments," *Advanced Engineering Informatics*, vol. 54, p. 101755, 2022.

[14] F. Hart and O. Okhrin, "Enhanced method for reinforcement learning based dynamic obstacle avoidance by assessment of collision risk," *Neurocomputing*, vol. 568, p. 127097, 2024.

[15] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[16] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," *arXiv preprint arXiv:1404.2334*, 2014.

[17] L. Jaillet and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 635–646, 2010.

[18] E. Catmull and R. Rom, *A class of local interpolating splines*. University of Utah, 1974.

[19] C. Yuksel, "On the parameterization of catmull–rom curves," *Computer-Aided Design*, vol. 43, no. 7, pp. 786–793, 2011.