

Article

Reducing Oscillations for Obstacle Avoidance in a Dense Environment Using Deep Reinforcement Learning and Time-Derivative of an Artificial Potential Field

Zhilong Xi ^{1,†} , Haoran Han ^{1,†} , Jian Cheng ^{1,*}  and Maolong Lv ²

¹ School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; xijohnston@gmail.com (Z.X.); hanadam@163.com (H.H.)

² Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an 710051, China; maolonglv@163.com

* Correspondence: chengjian@uestc.edu.cn

† These authors contributed equally to this work.

Abstract: Obstacle avoidance plays a crucial role in ensuring the safe path planning of quadrotor unmanned aerial vehicles (QUAVs). In this study, we propose a hierarchical framework for obstacle avoidance, which combines the use of artificial potential field (APF) and deep reinforcement learning (DRL) for training low-level motion controllers. Unlike traditional potential field methods, our approach modifies the state information received by the motion controllers using the outputs of the APF path planner. Specifically, the assumed target position is pushed away from obstacles, resulting in adjustments to the perceived position errors. Additionally, we address path oscillations by incorporating the target's velocity information, which is calculated based on the time-derivative of the repulsive force. Experimental results have validated the effectiveness of our proposed framework in avoiding collisions with obstacles and reducing oscillations.

Keywords: quadrotor unmanned aerial vehicle (QUAV); obstacle avoidance; artificial potential field (APF); deep reinforcement learning (DRL); oscillations



Citation: Xi, Z.; Han, H.; Cheng, J.; Lv, M. Reducing Oscillations for Obstacle Avoidance in a Dense Environment Using Deep Reinforcement Learning and Time-Derivative of an Artificial Potential Field. *Drones* **2024**, *8*, 85. <https://doi.org/10.3390/drones8030085>

Academic Editors: Jihong Zhu, Heng Shi, Zheng Chen and Minchi Kuang

Received: 10 January 2024

Revised: 26 February 2024

Accepted: 27 February 2024

Published: 1 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Due to their high flexibility, quadrotor unmanned aerial vehicles (QUAVs) have gained significant popularity in various applications, including parcel delivery [1], precision agriculture [2–4], search and rescue [5,6], and surveillance [7]. In these scenarios, the QUAV is typically required to autonomously navigate to a target position. However, in dense environments, unexpected obstacles can obstruct the path and lead to collisions. This task is even more challenging in a multi-agent system [8,9]. Therefore, obstacle avoidance becomes a crucial task for ensuring safe path planning. The QUAV must find an unobstructed path to the target while considering its physical limitations. Conventional approaches to obstacle-free path planning include Dijkstra, probabilistic roadmap (PRM), rapidly-exploring random trees (RRT), and artificial potential field (APF) [10–13]. Since these approaches do not assume the physical realization of the agent to be of a certain type, they are applicable to QUAV navigation. APF faces a critical challenge in dense environments. When multiple obstacles narrow the passageway to the target, the agent may experience oscillations due to the repulsive force created by the obstacles that conflicts with the attractive force towards the target. As a result, reaching the target smoothly becomes difficult.

Most existing approaches to reducing APF oscillations either rely on second-order optimization theory techniques or employ an escaping mechanism when the agent detects oscillations. However, these approaches are not fully compatible with a hierarchical path planning framework because they directly modify the forces applied to the QUAV without considering its often nonlinear dynamics. Additionally, they commonly attribute oscillations to local minima, which is not always the case. One possible solution to this

problem is to utilize global obstacle information to bend an imaginary rubber band path [14]. However, this approach poses challenges for real-time decision-making, as it requires prior knowledge of obstacle positions.

Based on the symmetrical motion controller design in [15], our previous work [16] enabled the utilization of an APF in a hierarchical framework, utilizing only local information provided by sensors mounted on the QUAV. In this framework, an APF path planner generates a state that incorporates modified position errors, while low-level motion controllers are responsible for both position and attitude control. In [17], we introduced the Hessian matrix of the APF as a damping term, which effectively dissipates system energy, and we proved its stability using an energy-based Lyapunov function. In this current work, we utilize the time-derivative of the APF to further reduce oscillations in QUAV path planning within a hierarchical control framework.

Considering that the QUAV dynamic system is nonlinear and strongly coupled, we use deep reinforcement learning (DRL) to train the low-level motion controllers [18]. DRL not only alleviates the labor-intensive process of parameter tuning, as compared to the traditional proportional–integral–derivative (PID) method, but also demonstrates remarkable fitting capability with the aid of deep neural networks. The key contributions of this study are outlined as follows.

1. We incorporate the time-derivatives of the potential function to account for the velocities of intermediate target points, thereby resolving the conflict between the fixed-point chasing logic of the DRL motion controller and the potentially erratic movements of the intermediate target points.
2. To enhance the control performance of a high-dimensional nonlinear system, we reconfigure the states of the DRL motion controller to eliminate any asymmetry and ensure stability.
3. We conducted comparative simulation experiments to validate the effectiveness of the proposed method in reducing oscillations and preventing overshooting.
4. The complex dynamics of the QUAV system are considered, which is different from other approaches that either do not discuss agent dynamics or simplify the dynamics as a simple second-order point mass model.

The structure of this paper is outlined as follows. In Section 2, we provide a comprehensive overview of the existing research on obstacle-free path planning. In Section 3, we introduce the necessary background knowledge, including the principles of conventional artificial potential fields, drone dynamics, and deep reinforcement learning. Section 4 elaborates on the methodology that we have employed in this study. We present the experiments conducted in Section 5 to validate the effectiveness of the proposed method in reducing oscillation and overshooting. Section 6 discusses limitations of this work and its applicability to dynamic and irregular environments. Finally, in Section 7, we conclude our findings.

2. Related Work

We identify three major approaches to achieve obstacle-free path planning. Since they do not set restrictions on the type of agents being used, these approaches are applicable to QUAV obstacle avoidance. The first approach is grid search, exemplified by algorithms such as Dijkstra, A*, and D* [10,19,20]. These algorithms are guaranteed to find a viable least-cost path when the environment is finite. However, they suffer from the curse of dimensionality, as the number of vertices to be explored grows exponentially with the dimensionality of the working space. Additionally, the need for higher control resolutions further increases the computational burden. The second approach is sampling-based search, with PRM [11] and RRT [12] being typical examples. For example, Farooq et al. [21] guided the QUAV to avoid dangerous zones in a dynamic environment by computing the PRM. Compared to grid search, these algorithms are capable of quickly producing paths in high-dimensional spaces. However, since sampling-based methods rely on global information, such as environment boundaries, they are not well-suited for real-time control where sensor data are used to

generate actions for the next step. The third approach is APF, which assumes that both the target and obstacles generate potential fields that influence the movement of the agent. By leveraging the gradient of the summed potential field, the agent is attracted to the target while being repelled from nearby obstacles. For example, Ma'arif et al. [22] used APF to guide a single QUAV to reach the target and avoid collisions with dynamic obstacles. APF is widely used in QUAV navigation due to its ease of implementation and ability to guide the QUAV using only local information.

Despite its numerous advantages, APF has certain inherent limitations [23]. One recognizable limitation is that the agent can easily become trapped in local minima, where the attraction and repulsion forces cancel each other out. Algorithms designed to address this problem can generally be classified into three categories. The first category is local minimum removal. For example, Kim et al. [24] employed harmonic functions to construct potential fields, allowing for the selection of singularity locations and the elimination of local minima in free space. The second category is local minimum escape. For instance, Park et al. [25] combined APF with simulated annealing, which introduces randomness to the agent's actions, enabling it to escape from local minima. Wang et al. [26] proposed the Left Turning scheme, which effectively handles U-shaped obstacles and helps the agent escape local minima. Lai et al. [27] proposed a dynamic step-size adjustment method to help the multi-UAV escape the local minima. The third category is local minimum avoidance. For instance, Doria et al. [28] utilized the deterministic annealing approach to expand the repulsive area and shrink the attractive area. This allows the agent to avoid local minima at the beginning, when the potential function is convex due to a high initial temperature. Additionally, Ge et al. [29] addressed a specific case of the local minimum problem known as goals non-reachable with obstacles nearby (GNRON). They considered the relative distance between the agent and the target when designing the repulsive potential function. As the agent approaches the target, this function approaches zero, thereby reducing the repulsive force in the target's vicinity and overcoming the local minimum problem. Overall, these different approaches highlight the efforts made to overcome the limitations of APF and improve its performance in various scenarios.

APF faces another dilemma, which is the occurrence of oscillations in narrow passages with densely distributed obstacles nearby. Previous research on oscillation reduction is relatively limited and mainly draws inspiration from optimization theory techniques. For instance, Ren et al. [30] proposed the use of Levenberg's modification of Newton's method as a solution to oscillation problems. This approach incorporates second-order information by utilizing the Hessian matrix. Additionally, they adjusted the control law to maintain a constant speed, ensuring smooth movement of the agents. Biswas et al. [31] further compared first-order gradient descent methods with two second-order approaches and concluded that Levenberg–Marquardt is more effective in generating smoother trajectories and improving convergence speeds.

The second branch of oscillation reduction algorithms introduces the concept of virtual obstacles or targets. Similar to LME used in tackling local minima, methods belonging to this branch often employ escaping techniques once oscillations are detected. For instance, Zhao et al. [32] enhanced the manipulator's predictive ability by incorporating dynamic virtual target points and utilized an extreme point jump-out function to escape oscillations. Zhang et al. [33] employed tangent APF to avoid local oscillations and introduced the back virtual obstacle setting strategy-APF algorithm, which enables the agent to return to previous steps and withdraw from concave obstacles. In a rule-based fashion, Zheng et al. [34] specified the condition for adding obstacles, compelling the resultant force to deflect when its angle to the obstacle center is too small. The dynamic step-size adjustment method proposed by Lai et al. [27] is also able to escape the jitter area where a local minimum is encountered, but it does not address oscillations in other cases, such as narrow passageways.

Oscillations can also be mitigated by modifying the repulsive forces in a certain manner. Tran et al. [35] estimated the projection of the repulsive force vector onto the

attractive force vector and subtracted this component to prevent the agent from moving in the opposite direction of the attractive force. Martis et al. [36] introduced vortex potential fields to achieve seamless cooperative collision avoidance between mobile agents, and these were validated using Lyapunov stability analysis. For larger obstacles with irregular shapes, Szczepanski [37] combined the benefits of repulsive APF and vortex APF by defining multiple layers around the surface area of the obstacles. This approach surpassed traditional APF and pure vortex APF in terms of path smoothness.

There are also works specifically designed for QUAV path planning and obstacle avoidance. For example, Meradi et al. [38] proposed a nonlinear model predictive control method based on quaternions for QUAVs' obstacle avoidance. Valencia et al. [39] constructed a QUAV obstacle detection and avoidance system using a monocular camera. Gageik et al. [40] discussed the use of complementary low-cost sensors in QUAV collision avoidance. However, the problem of oscillation reduction in QUAV obstacle-free path planning based on APF is largely under-explored.

With the advancement in DRL technology, motion controllers based on DRL have gained widespread usage in QUAV path planning and obstacle avoidance by virtue of their strong fitting capability. In this context, the APF can be seen as an upper layer that indirectly or directly influences the agent's current pursuit position, while the low-level motion control task is handled by the DRL agent. For instance, the RL environment may employ convolutional neural networks to receive information about the surrounding potential energy and generate estimated rewards for various actions [41]. Xing et al. [42] combined the enhanced APF method with deep deterministic policy gradient to navigate autonomous underwater vehicles. In our previous research, we developed a hierarchical framework where the APF path planner was utilized to modify the position errors perceived by the DRL motion controllers, effectively altering the target position [16]. In this study, we further address oscillations by considering the velocity of the virtual target point, which has been validated in a collision-free formation control task [17]. Experimental results demonstrate that this improved algorithm reduces oscillations compared to approaches that do not incorporate second-order information, such as velocities.

3. Preliminaries

3.1. Drone Dynamics

The simulation environment is built based on gym-pybullet-drones [43]. In this work, a QUAV with an "X" configuration is considered. Similar to many fixed-wing UAVs [44], it has six degrees of freedom. Its position and Euler angles are defined as $\mathbf{p} = [x, y, z]^T$ and $\Theta = [\phi, \theta, \psi]^T$, respectively. The Euler angles as shown in Figure 1 are obtained following the intrinsic z-y-x sequence (or, equivalently, extrinsic x-y-z sequence). Therefore, the rotation matrix from the body-fixed frame $\{\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b\}$ to the earth frame $\{\mathbf{x}_e, \mathbf{y}_e, \mathbf{z}_e\}$ is calculated by

$$\mathbf{R} = [\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b] = \begin{bmatrix} c_\theta c_\psi & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ c_\theta s_\psi & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \quad (1)$$

where the coordinates of $\mathbf{x}_b, \mathbf{y}_b$, and \mathbf{z}_b are in the earth frame, $\mathbf{z}_b = [z_b^x, z_b^y, z_b^z]^T$, c^ζ and s^ζ ($\zeta = \phi, \theta, \psi$) are short for $\cos \zeta$ and $\sin \zeta$, respectively, and $\mathbf{R} \in SO(3)$.

In this work, we use the frame of SE(3) control [45] to avoid using a transformation matrix \mathbf{W} between the Euler angles Θ and the angular velocity $\omega = [\omega^x, \omega^y, \omega^z]^T$, which suffers from singularity as θ approaches $\frac{\pi}{2}$. This technique overcomes the small angle restriction. The hat map $\hat{\cdot} : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ is defined as

$$\hat{\mathbf{a}} = \begin{bmatrix} 0 & -\alpha_3 & \alpha_2 \\ \alpha_3 & 0 & -\alpha_1 \\ -\alpha_2 & \alpha_1 & 0 \end{bmatrix}, \quad (2)$$

where $\alpha = [\alpha_1, \alpha_2, \alpha_3]^T$ is a three-dimensional vector. Using the differential property of Lie Group, the QUAV's dynamic system can be formally described by

$$\begin{aligned}\dot{\mathbf{p}} &= \mathbf{v}, \\ m\dot{\mathbf{v}} &= \mathbf{R}\mathbf{F} - m\mathbf{g}, \\ \dot{\mathbf{R}} &= \mathbf{R}\hat{\boldsymbol{\omega}}, \\ \mathbf{J}\dot{\boldsymbol{\omega}} &= -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\tau},\end{aligned}\quad (3)$$

where the velocity $\mathbf{v} = [v^x, v^y, v^z]^T$ and the gravity $\mathbf{g} = [0, 0, g]^T$ are in the earth frame, external thrust $\mathbf{F} = [0, 0, f]^T$ and torque $\boldsymbol{\tau} = [\tau^\phi, \tau^\theta, \tau^\psi]^T$ are in the body-fixed frame, m is the mass, and $\mathbf{J} = \text{diag}(I^\phi, I^\theta, I^\psi)$ is the inertia matrix.

The four rotors are controlled by the pulse width modulation (PWM) signal \mathbf{u} , and the rotation speed $\boldsymbol{\Omega} = [\Omega_1, \Omega_2, \Omega_3, \Omega_4]^T$ is described by

$$T_m \dot{\boldsymbol{\Omega}} = k_m \mathbf{u} + \mathbf{b}_m - \boldsymbol{\Omega}, \quad (4)$$

where T_m , k_m , and \mathbf{b}_m are the motors' coefficients, which remain constant. The thrust \mathbf{F} and torque $\boldsymbol{\tau}$ are then generated by the four rotors according to

$$\begin{bmatrix} f \\ \tau^\phi \\ \tau^\theta \\ \tau^\psi \end{bmatrix} = \begin{bmatrix} C_T & C_T & C_T & C_T \\ -\frac{L}{\sqrt{2}}C_T & -\frac{L}{\sqrt{2}}C_T & \frac{L}{\sqrt{2}}C_T & \frac{L}{\sqrt{2}}C_T \\ -\frac{L}{\sqrt{2}}C_T & \frac{L}{\sqrt{2}}C_T & \frac{L}{\sqrt{2}}C_T & -\frac{L}{\sqrt{2}}C_T \\ -C_D & C_D & -C_D & C_D \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}, \quad (5)$$

where L is the arm length and C_T, C_D are the thrust and torque coefficients, respectively.

The environment contains N obstacles denoted by $\mathcal{O} = [1, 2, \dots, N]$. The geometric relationship between the QUAV, the target, and obstacle i is shown in Figure 2, in which we define $\mathbf{p} = [x, y, z]^T$ as the QUAV's current position, \mathbf{p}_d as the target position, and \mathbf{o}_i as the closest point of obstacle i to the QUAV. Moreover, $\mathbf{e}_i = \mathbf{p} - \mathbf{o}_i = [e_i^x, e_i^y, e_i^z]^T$ and $\mathbf{e}_d = \mathbf{p} - \mathbf{p}_d = [e^x, e^y, e^z]^T$ are error vectors to obstacle i and the target, respectively.

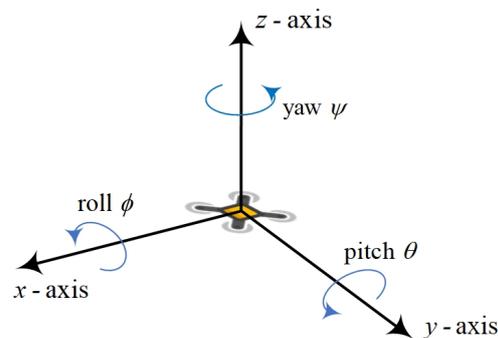


Figure 1. The body-fixed frame and the corresponding Euler angles.

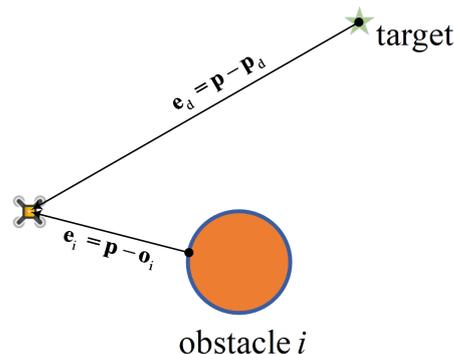


Figure 2. Geometric relationship of the QUAV, the target, and obstacle i .

3.2. Classical APF

The concept of APF was initially introduced by Khatib [13] with the aim of distributing collision avoidance across different levels of control. This approach involves creating a potential field, similar to an electrostatic field, within the agent's working space. In this field, the target location is represented as a valley, while obstacles are represented as peaks. By computing the negative gradient of the total potential function, the agent can determine a feasible path to the target while avoiding collisions with obstacles. From another perspective, the target exerts an attractive force on the agent, while the obstacles exert a repulsive force. To ensure the existence of these attractive and repulsive forces, the potential functions must be continuous and differentiable, except at the target position or within the obstacles. The calculation of the attractive and repulsive forces is performed by

$$\begin{aligned} \mathbf{F}_{\text{att}} &= -\nabla_{\mathbf{p}} U_{\text{att}}(\|\mathbf{e}_d\|), \\ \mathbf{F}_{\text{rep}_i} &= -\nabla_{\mathbf{p}} U_{\text{rep}_i}(\|\mathbf{e}_i\|), \end{aligned} \quad (6)$$

where $U_{\text{att}}(\cdot)$ and $U_{\text{rep}_i}(\cdot)$ are potential functions that receive scalars instead of vectors. This maintains the isotropy of the overall system, since the norms of the APF forces are only determined by the radial distances and not by the relative angles to the attraction or repulsion sources.

Figure 3 depicts the attractive and repulsive forces in the APF algorithm. In this figure, the green star represents the target position, while the orange circles symbolize obstacles with specific regions of influence. The repulsive force generated by obstacle i is denoted as F_{rep_i} , and the attractive force generated by the target is denoted as F_{att} . The resultant force exerted on the agent is represented as F_r . It is worth noting that APF can be applied to both global path planning and local path planning, as illustrated in Figure 3a and Figure 3b, respectively. In the global case, the obstacles have a maximum influence distance ρ_0 , and their positions and shapes are known in advance. However, in the local case, which is employed in this study, the QUAV solely relies on the relative distances and velocities obtained from sensors to calculate the APF forces. The maximum sensing range is denoted as d .

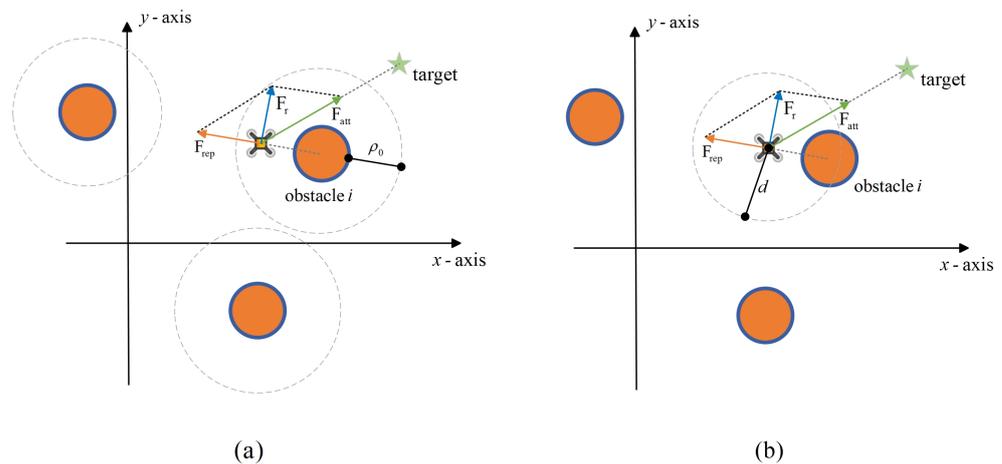


Figure 3. Schematic diagrams of APF forces. (a) The forces are generated based on the information returned by the QUAV's sensors. (b) The forces are generated using global geometric information.

3.3. Deep Reinforcement Learning

RL is a commonly adopted methodology in machine learning. An RL problem is usually characterized as a Markov decision process with the state initialized by a distribution $\mathcal{P}(s_1)$. In this work, we assume that the Markov property is satisfied. In other words, the transition probability for the next state is only determined by the current state and action, which is formalized as $\mathcal{P}(s_{t+1}|s_1, a_1, \dots, s_t, a_t) = \mathcal{P}(s_{t+1}|s_t, a_t)$. At time step t , the agent receives the current state s_t and, depending on whether the policy is stochastic or determin-

istic, takes an action a_t according to its policy $\pi_\alpha(\cdot|s) : \mathcal{S} \rightarrow \mathcal{P}(\mathcal{A})$ or $\mu_\alpha(s) : \mathcal{S} \rightarrow \mathcal{A}$, where α denotes policy parameters. The environment will then give a scalar reward $r_t \sim \mathcal{R}(s_t, a_t)$ and the next state $s_{t+1} \sim \mathcal{P}(s_t, a_t)$. The aim of the agent is to maximize the expected return $G_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i$, where $\gamma \in (0, 1)$ is the discount factor that prevents G_t from approaching infinity and controls how myopic the agent is.

To train the motion controller, we employ twin-delayed deep deterministic policy gradient (TD3), which is a model-free actor-critic DRL algorithm featuring three distinct characteristics. Firstly, it utilizes two critic networks, Q_{w_1} and Q_{w_2} , to tackle the issue of action value overestimation. Secondly, it incorporates smooth regularization to minimize the variance of the target update, which means the action is given by Equation (7), ensuring a more stable and reliable training process, which is crucial for achieving optimal performance:

$$\tilde{a}_{t+1} = \mu_{\alpha'}(s_{t+1}) + \epsilon, \quad \epsilon \sim \text{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c), \quad (7)$$

where $\mu_{\alpha'}$ is the target actor network, ϵ is the Gaussian noise, $\tilde{\sigma}$ is the standard deviation, and c is the noise bound. The loss function for the i -th critic network is represented by

$$L_{\text{TD3}}(w_i) = \mathbb{E}_{s_t, a_t, r_t, s_{t+1} \sim D} [(r_t + \gamma \min_{j=1,2} Q_{w'_j}(s_{t+1}, \tilde{a}_{t+1}) - Q_{w_i}(s_t, a_t))^2], \quad (8)$$

where D is the replay buffer and $Q_{w'_j}$ ($j = 1, 2$) represent the target critic networks.

The third feature of TD3 is that actor networks are updated only after a fixed number of updates d to the critic. The actor network's updates follow

$$\nabla_\alpha J(\alpha) = \mathbb{E}_{a_t = \mu_\alpha(s_t)} [\nabla_\alpha \mu_\alpha(s_t) \nabla_{w_1} Q_{w_1}(s_t, a_t)]. \quad (9)$$

4. Methodology

4.1. Motion Controller Training

As is shown in Figure 4, motion control is divided into two hierarchies: translational control and angular control. Due to the presence of gravity, vertical control differs inherently from horizontal control. Hence, we further subdivide translational control into vertical and horizontal control. As the three angular controllers adhere to the same control logic, and likewise the two horizontal controllers, only three networks are required. The actor networks' parameters for angular, vertical, and horizontal control are denoted by α_a , α_v , and α_h , respectively. Their states are designed as

$$\begin{aligned} s^\xi &= [e^\xi, \omega^\xi]^T, \quad (\xi = \phi, \theta, \psi) \\ s^x &= [e^x, v^x, \text{sign}(e^x)e^z, \text{sign}(e^x)v^z, \bar{\theta}, \dot{\theta}]^T, \\ s^y &= [e^y, v^y, \text{sign}(e^y)e^z, \text{sign}(e^y)v^z, \bar{\phi}, \dot{\phi}]^T, \\ s^z &= [e^z, v^z]^T, \end{aligned} \quad (10)$$

in which $\bar{\theta}$ and $\bar{\phi}$, variables that characterize the pitch and the roll, are defined as

$$\begin{aligned} \bar{\theta} &= \arctan \frac{z_b^x}{z_b^z} \\ \bar{\phi} &= \arctan \frac{z_b^y}{z_b^z}. \end{aligned} \quad (11)$$

There are a few points to note here. First, x_d, y_d, z_d , and ψ_d are given by external sources, while ϕ_d and θ_d are calculated from the outputs of the translational controllers. Second, horizontal states not only consider the corresponding position and velocity errors but also take into account height and angular information. Third, angular errors

e_ζ are defined using the rotation matrix instead of the angular differences, and they are calculated by

$$\mathbf{e}_R = [e^\phi, e^\theta, e^\psi]^T = \frac{1}{2}(\mathbf{R}_d^T \mathbf{R} - \mathbf{R}^T \mathbf{R}_d)^\vee, \tag{12}$$

in which the vee map $\vee : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$ is the inverse of the hat map, and \mathbf{R}_d is the desired rotation matrix uniquely determined by ϕ_d, θ_d , and ψ_d .

To obtain ϕ_d and θ_d , the translational controllers' networks first generate desired forces along each axis following

$$\begin{aligned} f_d^x &= mc_t \mu_{\alpha_h}(s^x), \\ f_d^y &= mc_t \mu_{\alpha_h}(s^y), \\ f_d^z &= m(c_t \mu_{\alpha_v}(s^z) + g), \end{aligned} \tag{13}$$

where the networks' outputs are bounded by $(-1, 1)$, and c_t is the maximum translational acceleration. Then, the desired roll and pitch can be calculated as [15]

$$\begin{aligned} \phi_d &= \arcsin \frac{s_{\psi_d} f_d^x - c_{\psi_d} f_d^y}{\|[f_d^x, f_d^y, f_d^z]^T\|}, \\ \theta_d &= \arctan \frac{c_{\psi_d} f_d^x + s_{\psi_d} f_d^y}{f_d^z}. \end{aligned} \tag{14}$$

The ultimate desired thrust and torques are generated following Equation (15), where c_r is the maximum angular acceleration. We can find the desired steady-state rotor speed $\Omega_{ss} = k_m \mathbf{u} + \mathbf{b}_m$ by taking the inverse matrix in Equation (5), and the PWM signal \mathbf{u} will eventually be obtained.

$$\begin{aligned} f_d &= \frac{f_d^z}{c_\phi c_\theta}, \\ \tau_d^\zeta &= I^\zeta c_r \mu_{\alpha_a}(s^\zeta). \quad (\zeta = \phi, \theta, \psi) \end{aligned} \tag{15}$$

Equation (16) defines the reward function for the angular, vertical, and horizontal reinforcement learning environments. In this equation, e_t^ζ represents the error at the current time, and ζ indicates the specific controller to which the error is applied. The agent receives a reward when the error approaches zero and is penalized otherwise. One notable advantage of this reward design is the absence of hyperparameters that require manual fine-tuning.

$$r_t^\zeta = |e_t^\zeta| - |e_{t+1}^\zeta| \tag{16}$$

During the training process, the reinforcement learning environments receive the agent's action a_t^ζ and output the next state s_{t+1}^ζ as well as the reward r_t^ζ based on the current state s_t^ζ . In this work, the pitch and x -axis controllers are chosen as the prototypes for the angular and horizontal controllers, meaning that $\zeta = \theta, x, z$ during training. The training follows a certain order. First, the roll and vertical controllers are trained independently. When the pitch controller is being trained, we set f_d, τ_d^ϕ , and τ_d^ψ as zero, and $\tau_d^\theta = I^\theta c_r \mu_{\alpha_a}(s^\theta)$, while when the vertical controller is being trained, we set f_d as $m(c_t \mu_{\alpha_v}(s^z) + g)/c_\phi c_\theta$ and the desired torques as zero. The x -axis controller is trained only after α_a and α_h are obtained because s^x contains height and pitch information, which is affected by the vertical and pitch controllers. No obstacles or external disturbances are considered during training.

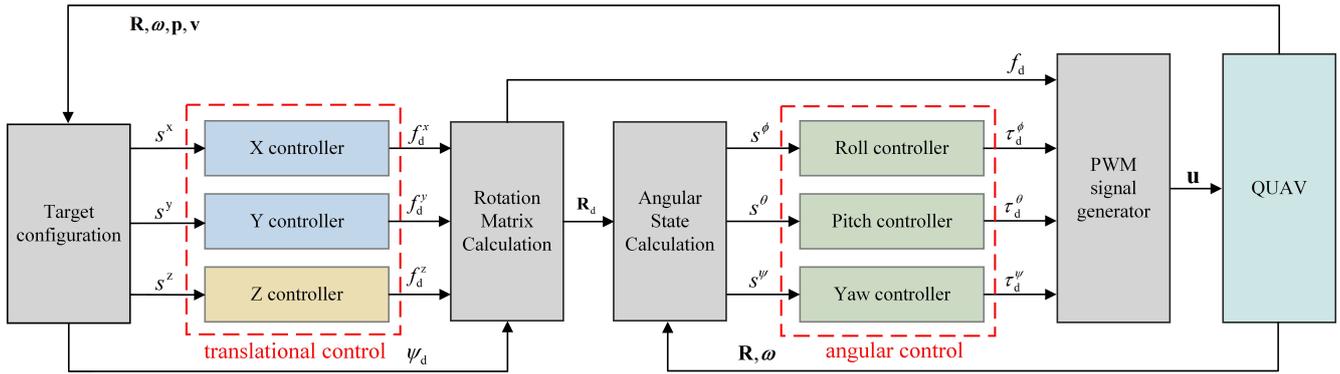


Figure 4. The general framework of the motion controller.

4.2. State Transformation

The trained motion controllers have the ability to navigate the QUAV in the corresponding dimension. However, when the states in the testing phase are too large, it becomes challenging for the networks to generate reasonable outputs based on the limited training samples. This issue typically arises in translational control, where the position errors are not bounded, except by the simulator. Since it is not feasible to gather information for position errors of arbitrary magnitude within a finite training period, using the policy networks in a large or boundless environment can result in uncontrollable behavior, which is inherently dangerous. To mitigate the impact of large horizontal position errors, we employ a hard-clipping technique on the variables e^x and e^y , with a clipping bound of d_c . Although no hard-clipping is applied to e^z , its range is constrained due to the presence of a floor and a ceiling in the working space of this study.

As observed in [18], the trained DRL controllers will decelerate to avoid overshooting, indicating that there exists a natural velocity upper bound \bar{v} , since $\mu_\alpha(s) > 0$ when the velocity is smaller than \bar{v} and $\mu_\alpha < 0$ when it exceeds the bound. Considering that \bar{v} is sometimes too large to guarantee safety in the testing phase, we manually set an upper bound \bar{v} to the velocity in the testing phase to prevent the QUAV from moving too quickly in a free space. To increase data efficiency, we multiply e^{v^x} and e^{v^y} by \bar{v}/\bar{v} , which can map the horizontal velocity errors from $[0, \bar{v}]$ to $[0, \bar{v}]$ and therefore force the actual velocity inputs to have a similar scale as the training samples.

In addition to the aforementioned issues, it is necessary to address the problem of anisotropic control performance. As depicted in Figure 5a, we denote the force exerted on the QUAV as F_1 when it is located at position A and as F_2 when it is at position B. Ideally, if positions A and B are equidistant from the target at a distance r , we expect the two force vectors to have identical magnitudes and angles relative to the corresponding position error vectors. This implies that the control performance should exhibit isotropy. However, the current motion controllers we have obtained do not guarantee isotropy due to two reasons: (1) the x -axis and y -axis controllers independently calculate the force components along each axis based on s^x and s^y , and (2) the policy networks are non-linear. To preserve this property, we transform the horizontal coordinate system such that the force vector components are calculated radially and tangentially to the horizontal error vector $\mathbf{e}_h = [e^x, e^y, 0]^T$. As shown in Figure 5b, the unit radial and tangential vectors are defined as

$$\mathbf{h} = \frac{\mathbf{e}_h}{\|\mathbf{e}_h\|}, \quad T(\mathbf{h}) = \mathbf{h} \times [0, 0, -1]^T. \quad (17)$$

The horizontal states in the testing phase are so far modified as

$$\begin{aligned} s^{\text{rad}} &= [e^{\text{rad}}, (\bar{v}/\bar{v})\mathbf{v}^T\mathbf{h}, \text{sign}(e^{\text{rad}})e^z, \text{sign}(e^{\text{rad}})v^z, \tilde{\theta}, \dot{\tilde{\theta}}]^T, \\ s^{\text{tan}} &= [e^{\text{tan}}, (\bar{v}/\bar{v})\mathbf{v}^T T(\mathbf{h}), \text{sign}(e^{\text{tan}})e^z, \text{sign}(e^{\text{tan}})v^z, \tilde{\phi}, \dot{\tilde{\phi}}]^T, \end{aligned} \quad (18)$$

in which $e^{\text{rad}}, e^{\text{tan}}, \tilde{\theta}$, and $\tilde{\phi}$ are defined as

$$\begin{aligned} e^{\text{rad}} &= \text{clip}(\mathbf{e}_d^T \mathbf{h}, -d_c, d_c), \\ e^{\text{tan}} &= \text{clip}(\mathbf{e}_d^T T(\mathbf{h}), -d_c, d_c), \\ \tilde{\theta} &= \arctan \frac{\mathbf{z}_b^T \mathbf{h}}{z_b^z}, \\ \tilde{\phi} &= \arctan \frac{\mathbf{z}_b^T T(\mathbf{h})}{z_b^z}. \end{aligned} \tag{19}$$

Similarly, the horizontal forces defined in Equation (13) should be modified following

$$[f_d^x, f_d^y, 0]^T = mc_t \mu_{\alpha_h}(s^{\text{rad}}) \mathbf{h} + mc_t \mu_{\alpha_h}(s^{\text{tan}}) T(\mathbf{h}). \tag{20}$$

It should be noted that, up to now, $e^{\text{rad}} = \sqrt{(e^x)^2 + (e^y)^2} \geq 0$ and $e^{\text{tan}} = 0$, and there is no reason for the QUAUV to move along $T(\mathbf{h})$. This is because no obstacles are taken into consideration yet. As we shall see in the next section, they will be further modified by the repulsive potential fields.

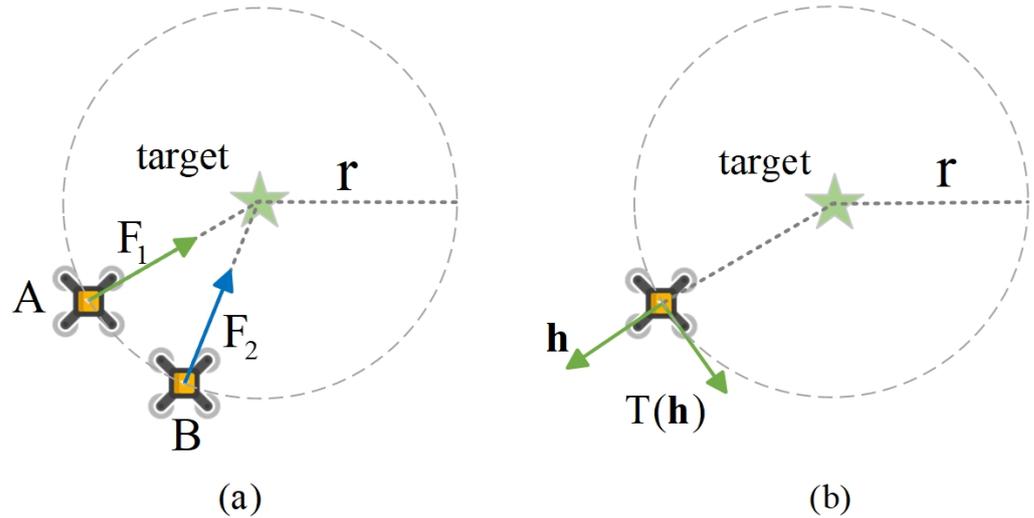


Figure 5. Isotropic control performance. (a) For the property of isotropy to be maintained, the norms of F_1 and F_2 should be the same. (b) The horizontal coordinate system is transformed such that horizontal axes are either parallel or tangential to the horizontal position error vector.

4.3. APF for Network Controller

In a hierarchical framework, RL motion controllers are solely responsible for the navigation of the QUAUV in open spaces and do not address higher-level tasks such as obstacle avoidance. In this study, we incorporate the use of APF to guide the RL motion controllers by influencing the state inputs, thereby indirectly altering the QUAUV’s current pursuit position. Figure 6 provides an illustration of how the target is updated, with the saturated target being the direct result of the clipping operation along the radial direction. Similar to traditional APF methods, any obstacle that falls within the QUAUV’s sensing range generates a repulsive force. However, instead of directly applying this force to the agent, it is utilized to further modify the target’s position. This is accomplished by providing the motion controllers with modified position errors in the following manner:

$$\begin{aligned} e^{\text{rad}} &= \text{clip}(\mathbf{e}_d^T \mathbf{h}, 0, d_c) - \mathbf{F}_{\text{rep}}^T \mathbf{h}, \\ e^{\text{tan}} &= -\mathbf{F}_{\text{rep}}^T T(\mathbf{h}). \end{aligned} \tag{21}$$

It is important to note that distance clipping is performed prior to the modification of errors by \mathbf{F}_{rep} , as depicted in Figure 6a. This implies that the radial and tangential position errors, e^r and e^t , are not strictly limited by the distance d . This is done to ensure that the repulsive forces exerted by nearby obstacles have a significant impact on the movements of the QUAUV. Without this consideration, the actual influence of repulsion on the target would be insignificant, as shown in Figure 6b, particularly when the original target is located far away.

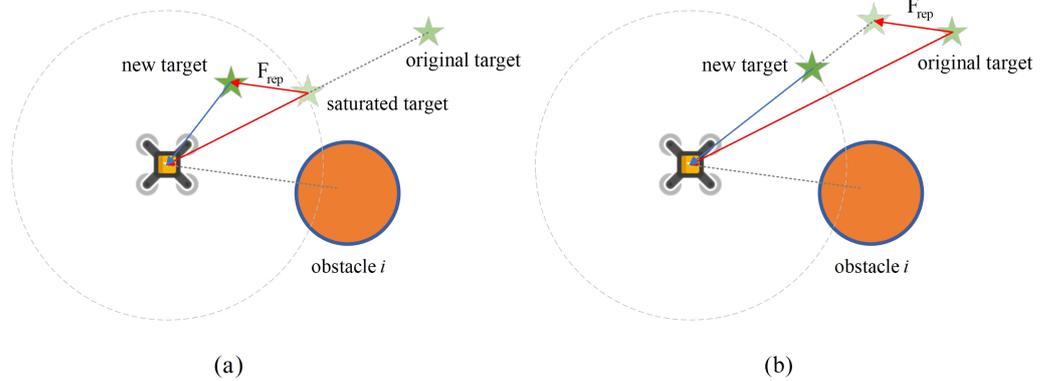


Figure 6. The target position is further adjusted based on the repulsive force. (a) In this study, the order of operations involves conducting the clipping operation before the target is pushed by the APF. (b) The clipping operation is performed at the end, which minimizes the impact of the repulsion.

The newly-introduced repulsive forces are dynamic, as the geometric relationship between the QUAUV and adjacent obstacles changes. This implies that the position of the target also changes dynamically. Therefore, it is not enough to solely consider the agent's current velocity in the motion controllers' states, as the target was assumed to be stationary during the training process. In this study, we calculate the derivative of the repulsive force with respect to time to determine the target's velocity. Using the chain rule, we can obtain that

$$\mathbf{v}^{\text{tar}} = \dot{\mathbf{F}}_{\text{rep}} = - \sum_{i:d_i \leq d} \nabla_{\mathbf{p}}^2 U_{\text{rep}_i}(\|\mathbf{e}_i\|) \dot{\mathbf{p}} = - \sum_{i:d_i \leq d} \mathbf{H}_i \mathbf{v}, \quad (22)$$

in which \mathbf{v}^{tar} is the velocity of the dynamic target, $d_i = \sqrt{(e_i^x)^2 + (e_i^y)^2}$ is the horizontal distance to the i -th obstacle, and \mathbf{H}_i is the Hessian matrix for the i -th obstacle's repulsive field. A few properties are to be satisfied when designing $U_{\text{rep}_i}(\cdot) : (0, d] \rightarrow \mathbb{R}$:

- $U_{\text{rep}_i}(x)$ should be monotonically decreasing with respect to x ;
- $\|\nabla_{\mathbf{p}} U_{\text{rep}_i}(x)\|$ should be monotonically decreasing with respect to x ;
- $\lim_{x \rightarrow 0^+} \|\nabla_{\mathbf{p}} U_{\text{rep}_i}(x)\| = +\infty$ and $\|\nabla_{\mathbf{p}} U_{\text{rep}_i}(d) = 0\|$;
- $U_{\text{rep}_i}(x)$ should be second-order differentiable.

The first requirement is to ensure that the force is repulsive. The second property states that the influence of the obstacle weakens as the distance increases. The third property implies that the repulsion force approaches infinity at the surface of the obstacle, while it becomes zero when the obstacle is at the boundary of the sensing range. The final requirement guarantees the existence of the Hessian matrix. Since the repulsive APF is only applied through its derivative forms, it is reasonable to directly design $\mathbf{F}_{\text{rep}_i}$ as

$$\mathbf{F}_{\text{rep}_i} = \begin{cases} \lambda \left(\frac{1}{d_i^2} - \frac{1}{d^2} \right) \frac{\mathbf{e}_i}{\|\mathbf{e}_i\|} & d_i \leq d \\ \mathbf{0} & d_i > d \end{cases}, \quad (23)$$

where λ is a manually set factor that scales the intensity of repulsion.

Considering that the calculation of \mathbf{H}_i necessitates significant computational resources and is time-consuming, we approximate \mathbf{v}^{tar} by utilizing the time difference of \mathbf{F}_{rep} . If

the radial component of \mathbf{v}^{tar} points towards the target, we eliminate it, as in such cases, the obstacle poses no threat to the agent. To be more specific, we define a distinct target velocity in the radial direction using the following approach:

$$\mathbf{v}^{\text{rad}} = \begin{cases} \mathbf{v}^{\text{tar}} & \mathbf{h}^T \mathbf{v}^{\text{tar}} > 0 \\ \mathbf{0} & \mathbf{h}^T \mathbf{v}^{\text{tar}} \leq 0 \end{cases} . \quad (24)$$

This technique prevents e^r from exceeding the bound d and therefore avoids overshooting. Ultimately, the horizontal states in the testing phase are modified as

$$\begin{aligned} s^{\text{rad}} &= [e^{\text{rad}}, (\tilde{v}/\bar{v})(\mathbf{v} - \mathbf{v}^{\text{rad}})^T \mathbf{h}, \text{sign}(e^{\text{rad}})e^z, \text{sign}(e^{\text{rad}})v^z, \tilde{\theta}, \dot{\theta}]^T, \\ s^{\text{tan}} &= [e^{\text{tan}}, (\tilde{v}/\bar{v})(\mathbf{v} - \mathbf{v}^{\text{tar}})^T T(\mathbf{h}), \text{sign}(e^{\text{tan}})e^z, \text{sign}(e^{\text{tan}})v^z, \tilde{\phi}, \dot{\phi}]^T. \end{aligned} \quad (25)$$

This design enables the seamless integration of RL motion control and APF obstacle avoidance, without requiring any modifications to the underlying structure of the pre-trained motion controllers. The modification of states only occurs during the testing phase. What sets this strategy apart from traditional APF approaches is that, instead of directly applying forces to the QUAUV, they are utilized to propel the target. This approach is equivalent to generating a sequence of waypoints that are carefully designed to avoid any potential obstacles.

5. Simulation Experiments

5.1. Training Setup

The QUAUV's dynamics followed Crazyflie 2.0 [46]. All agents were trained using TD3, whose hyperparameters are summarized in Table 1. The time step was set as 0.02 s to ensure control accuracy, while the upper bounds of translational and angular acceleration were $c_t = 5 \text{ m/s}^2$ and $c_r = 20 \text{ rad/s}^2$, respectively.

Table 1. TD3 hyperparameter settings.

Symbols	Descriptions	Values
$ \mathcal{D} $	Replay Buffer Size	1×10^4
N	Batch Size	512
γ	Discount Factor	0.99
τ	Target Update Rate	0.1
δ	Learning Rate	1×10^{-3}
$\tilde{\sigma}$	Noise Variance	0.2
c	Noise Clip	0.5
d	Policy Update Delay	2

An MLP with two layers, each consisting of 32 nodes, was utilized to train each controller. The optimizer used in this process was Adam. The targets were randomly sampled from a uniform distribution. Specifically, angular targets were chosen from the range of $(-\frac{\pi}{2}, \frac{\pi}{2})$, while translational targets were selected from the range of $[-5, 5]$. It is worth noting that the horizontal network was trained only after the angular and vertical networks had been obtained. The training stage of each controller comprised a total of 200 training episodes, with episode lengths set at 100 and 500 for angular and translational control respectively. This ensured that the agent had sufficient time to reach the target before an episode concluded. Throughout the training phase, no obstacles or external disturbances were taken into consideration.

5.2. Testing Setup

We designed three different controllers for all experiments. Controller 1 utilized the default setting, while controller 2 excluded the axial target velocity cancellation operation.

On the other hand, controller 3 solely employed APF to influence position errors, without affecting velocity errors.

The testing environment consisted of an open space with a floor and a ceiling that was 5 m high. Additionally, there were several cylindrical obstacles, each standing at a height of 5 m, strategically placed around the center of the space. The positions of these obstacles were determined by uniformly selecting the distances of their axes from the center within the range of $[0, 6]$. Furthermore, the radii of the cylinders were uniformly chosen from the range of $[0.9, 1.1]$. It was ensured that the minimum distance between any two obstacles was at least d_{\min} . For each trial, three QUAUVs were utilized, which acted as dynamic obstacles and generated APF repulsive forces similar to the fixed obstacles in the environment.

They were initialized at $[8 \sin(\frac{2k\pi}{3}), 8 \cos(\frac{2k\pi}{3})]$, and the corresponding targets were at $[-8 \sin(\frac{2k\pi}{3}), -8 \cos(\frac{2k\pi}{3})]$, where $k = 0, 1, 2$. As the agents moved, proximity sensors mounted on them could detect nearby obstacles. The maximum sensing range d was 2 m, the position error clipping bound d_c was 0.5 m, while the repulsive force scaling factor λ was set as 0.18. The manually set velocity upper bound \bar{v} was set as 2 m/s, and through experiments, it was found that the natural velocity upper bound $\tilde{v} = 1.5$ m/s.

The agents faced the challenge of reaching the targets on the opposite side without any collisions. The success of navigation was determined by whether each agent entered the 0.1 m proximity of its corresponding target. We created four different environments with specific values for d_{\min} and N : (0.7, 15), (0.7, 12), (0.9, 12), and (0.9, 9). In each environment, we conducted 1000 trials for every controller setting. Based on these trials, we calculated the success rates and the average time taken for successful trials.

To evaluate path smoothness, we calculated the curvatures of projection of the actual paths on the horizontal plane according to

$$\kappa = \frac{|\dot{x}\ddot{y} - \dot{y}\ddot{x}|}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}}, \quad (26)$$

in which the time-derivatives of positions are approximated by the differences between two time steps. It should be emphasized that κ may be either positive or negative, depending on the turning direction. Path oscillations are reflected by the occurrence of sharp turns, which correspond to huge absolute curvatures. In other words, if a path contains multiple points with huge absolute curvatures, it is oscillatory. In this work, the curvature threshold for a sharp turn is defined as 1000.

5.3. Experiment Results

5.3.1. General Performance

Figure 7 is a vertical view of the agents' trajectories as they moved around in the four environments, where the circles represent cylindrical obstacles, the curves in orange, green, and blue illustrate the trajectories, and the stars at their ends are the targets. A smaller d_{\min} indicates a more challenging scenario, since the minimum width of the passageway between two obstacles was smaller, while a bigger N means there were more obstacles around the origin. It can be seen that the hierarchical control framework we have proposed is capable of navigating the agents to the corresponding targets smoothly without collisions or detours, even in the densest environment.

To validate the effectiveness of the proposed modifications, we conducted 1000 trials for the three controllers in various environments. The success rates and average reaching times are presented in Table 2. In the environment with low density (0.9, 9), all controllers achieved a high success rate. Among them, controller 2 demonstrated the fastest performance, with an average reaching time of only 22.801 s. This can be attributed to controller 2's tendency to accelerate more dramatically when the repulsive force contributes to the attractive force. However, as the density increased, controller 2 experienced a more significant decrease in success rate compared to controller 1. This is because the higher

acceleration increases the risk of overshooting, which will be discussed in the following section. Furthermore, controller 3, which did not consider any velocity errors, exhibited a severe decline in performance. It not only took more time to reach the targets but also struggled to handle dense environments. This was due to the delay between the assumed target and the actual target, modified by the repulsive APF forces. As a result, oscillatory behaviors were observed. In summary, the introduction of velocity errors effectively mitigated oscillations, while the cancellation of the target velocity radial component enhanced the agents' ability to address overshooting and the resulting oscillations. However, this came at the expense of slower agent movement.

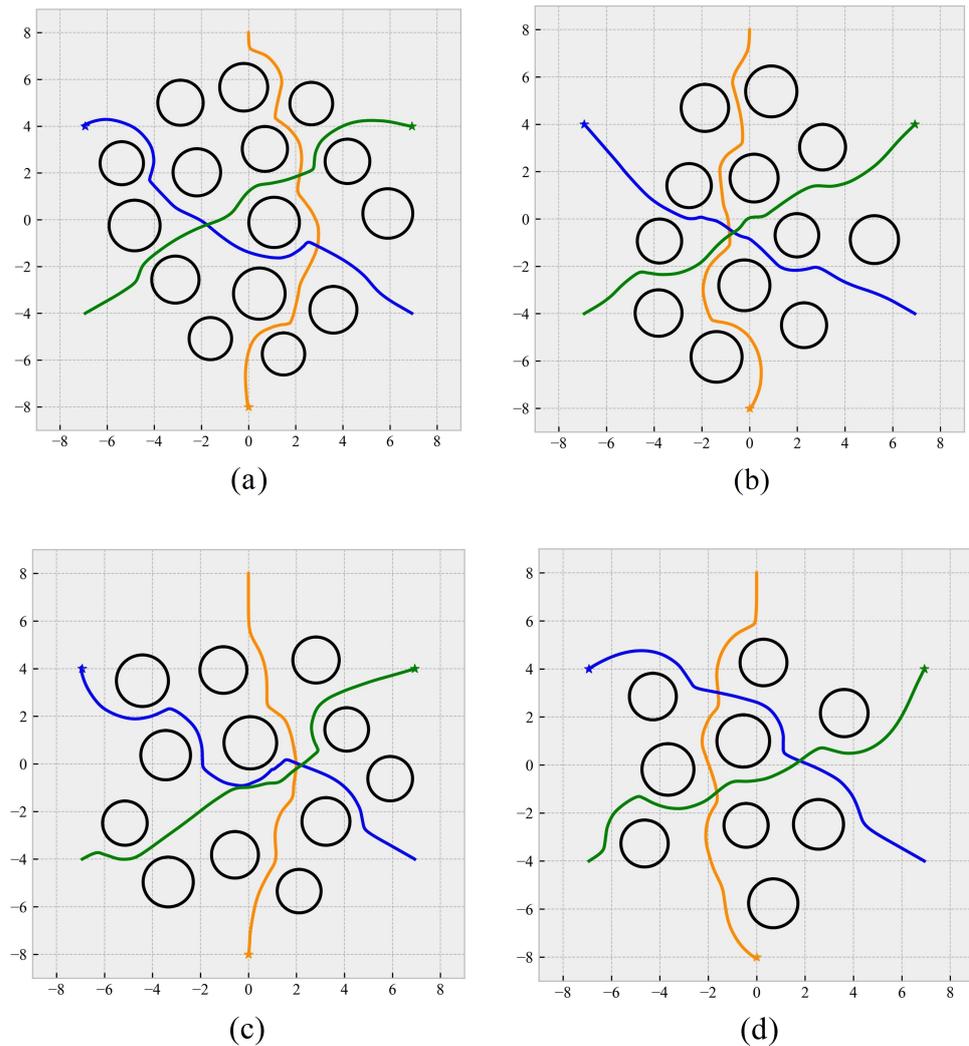


Figure 7. QUAV performance with the designed controllers in four environments, with the densest on the leftmost side and the sparsest on the rightmost side. (a) $d_{\min} = 0.7, N = 15$. (b) $d_{\min} = 0.7, N = 12$. (c) $d_{\min} = 0.9, N = 12$. (d) $d_{\min} = 0.9, N = 9$.

Table 2. Success rate (SCR) and average reaching time (ART) of different controllers.

Settings	Controller 1		Controller 2		Controller 3	
	SCR	ART (s)	SCR	ART (s)	SCR	ART (s)
(0.7, 15)	0.950	29.545	0.886	27.433	0.364	32.236
(0.7, 12)	0.968	27.194	0.945	25.223	0.675	29.012
(0.9, 12)	0.986	26.000	0.953	24.181	0.950	27.836
(0.9, 9)	0.989	24.529	0.975	22.801	0.968	25.984

The method is also applicable to an environment with more QAVs, as shown in Figure 8. The environment setting is similar to that discussed in the testing setup. In general, the agents are able to reach their targets without collisions. For example, in Figure 8b, where three pairs of agents move head to head, they can still reach the targets safely.

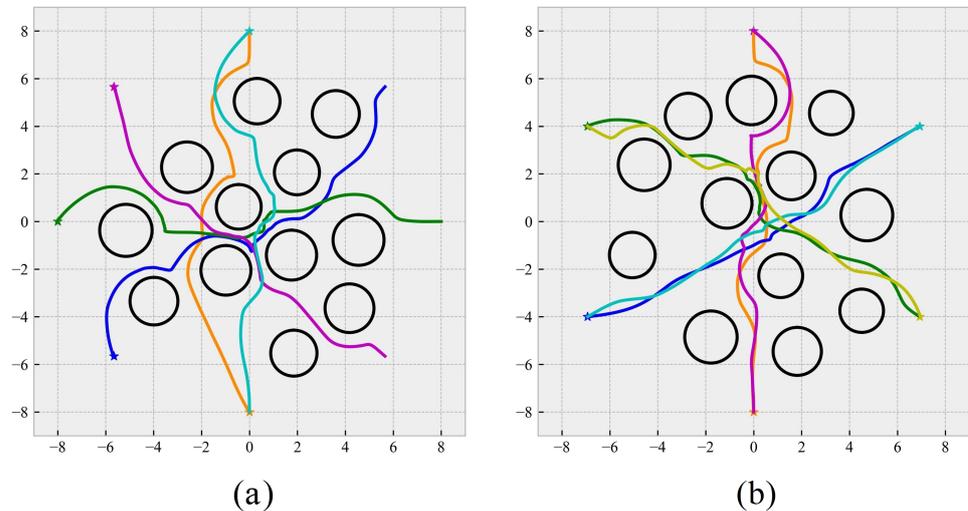


Figure 8. The method is applicable to cases with more QAVs. The environment settings are $d_{\min} = 0.7$, $N = 12$. (a) The case with 5 agents. (b) The case with 6 agents.

5.3.2. Oscillation Reduction and Overshooting Prevention

Table 3 lists the average number of points with an absolute curvature greater than 1000 for the three controllers over all successful trials. It can be seen that controller 3 demonstrates the worst performance in terms of path smoothness, which testifies that target velocity information plays an essential role in reducing path oscillations. On the other hand, controllers 1 and 2 are close, with the path of the former being slightly more oscillatory. It should be noted that overshooting is not equivalent to global path smoothness, since it only occurs at certain areas, such as the triangular trap. Therefore, the oscillations caused by overshooting cannot easily be reflected by path curvature.

Table 3. The average number of sharp turns ($|\kappa| > 1000$) of different controllers.

Settings	Controller 1	Controller 2	Controller 3
(0.7, 15)	79.427	73.855	122.403
(0.7, 12)	54.788	51.731	90.938
(0.9, 12)	41.925	33.658	75.417
(0.9, 9)	30.434	25.136	53.372

To give quantitative analysis to the effectiveness of radial target velocity cancellation operation as defined by Equation (24), another experiment setting was used. Concretely, three cylindrical obstacles with a radius of 1.5 m were randomly placed on a ring whose inner and outer radii were 2.3 m and 2.5 m, respectively. The three QAVs were initialized at $[0.6 \sin(\frac{2k\pi}{3}), 0.6 \cos(\frac{2k\pi}{3})]$, and the corresponding targets were at $[-8 \sin(\frac{2k\pi}{3}), -8 \cos(\frac{2k\pi}{3})]$, where $k = 0, 1, 2$. In this way, they were trapped by the encircled obstacles at first, and the task was to escape the triangular trap. Ideally, the maximum distance between any QAV and the corresponding target position is 8.6 m, otherwise it is caused by overshooting. In this experiment, overshooting cases are defined as cases where at least one QAV's distance to its target exceeded 10 m, as shown in Figure 9a. Furthermore, we define trapped cases where at least one QAV's distance to the origin is within 4 m, as shown in Figure 9b. Table 4 shows the results. It can be seen that controller 1 encountered the fewest overshooting cases, which shows that the velocity cancellation operation helps prevent-

ing overshooting. Moreover, since this operation simplifies the obstacle environment by neglecting obstacles behind it, trapped cases are largely avoided as well.

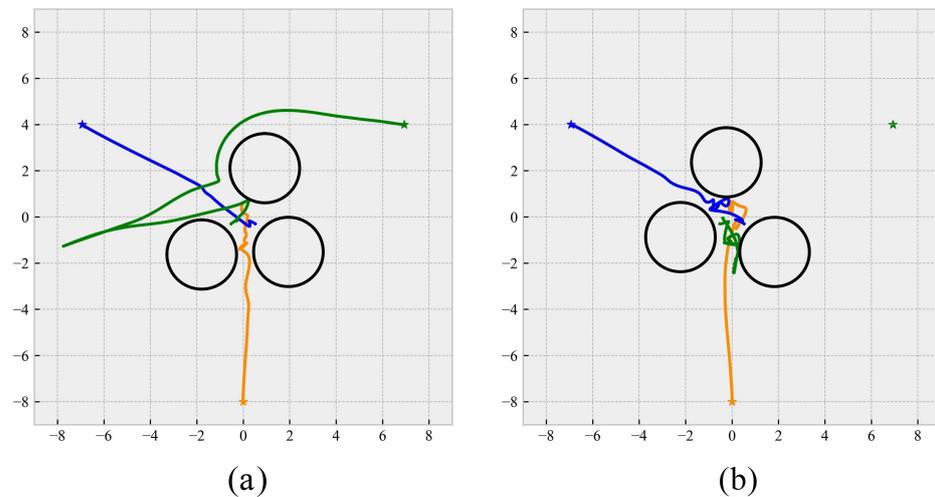


Figure 9. (a) Overshooting case. (b) Trapped case.

Table 4. The number of overshooting and trapped cases of different controllers.

	Controller 1	Controller 2	Controller 3
Overshooting Cases	23	145	481
Trapped Cases	24	199	641
SCR	0.969	0.732	0.237
ART (s)	13.735	13.958	18.566

An intuitive comparison of the three controllers can be found in Figure 10. In Figure 10b, the agent represented by the green trajectory experienced oscillation within the triangular area formed by obstacles 1, 2, and 3. The reason behind this oscillation was that the combined force exerted by obstacle 1 and obstacle 2 was directed towards the target and was not counteracted even though they did not pose a threat to the agent. Consequently, the force towards the goal became excessively strong, leading to the agent overshooting in this direction. Furthermore, disregarding velocity information entirely resulted in more frequent oscillations. The reason was that, in this case, the agent assumed the intermediate target to be fixed, while in reality the intermediate target was affected by the relative position between the agent and the obstacles. Oscillations occurred when the intermediate target updated before the agent reached the previous one. This can be observed in Figure 10c, where all agents' trajectories exhibit a zigzag pattern, especially when maneuvering between obstacles. This increases the risk of collision with nearby obstacles.

5.3.3. The Effect of a Clipping Bound

In the mentioned experiments, we utilized a position error clipping bound of $d_c = 0.5$ m. This choice was made based on previous experiments that indicated that it was the safest option. We varied the value of d_c for controller 1, ranging from 0.3 m to 1.0 m with an increment of 0.1 m. A total of 1000 trials were conducted with the environment parameters set at (0.7, 12). Additionally, we observed that, for d_c values of 0.3 m and 0.4 m, the agent's velocity was relatively low, resulting in a longer time to reach the target. To account for more trials that had the potential to reach the targets, we relaxed the time requirement for success. This adjustment allowed us to consider a broader range of trials. The results of these experiments are presented in Table 5. From the table, it can be observed that the average reaching time is inversely correlated with d_c . Furthermore, $d_c = 0.5$ yielded the highest success rate. This can be attributed to the fact that, when the agent moved too slowly, it lacked the flexibility to handle imminent collisions. Conversely,

when the agent moved too quickly, the outputs of the low-level flight controllers were not sufficiently strong to decelerate the agent in time, thereby increasing the likelihood of collisions.

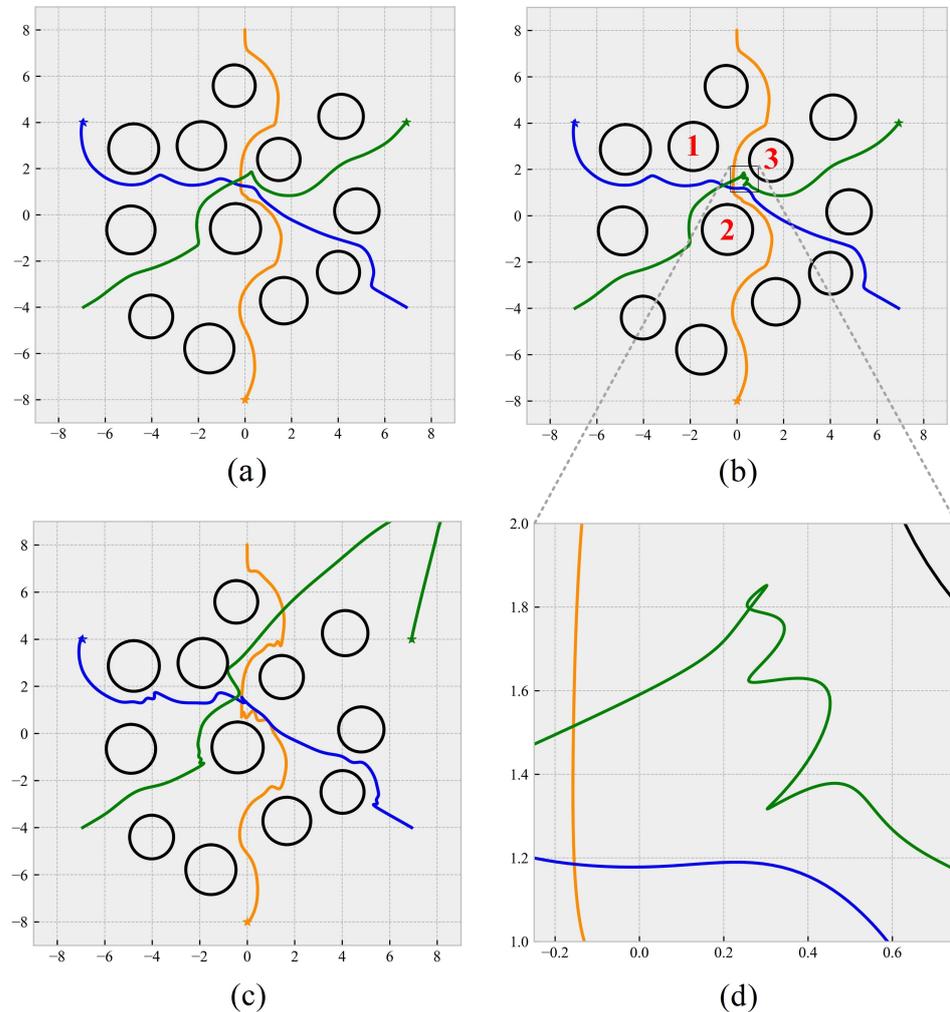


Figure 10. Performance comparison of different controllers in the same environment, where $d_{\min} = 0.7$ and $N = 12$. (a) Controller 1. All the designs mentioned previously were used. (b) Controller 2. Radial component of the target velocity was not canceled even when it contributed to the attractive force. (c) Controller 3. No velocity errors were introduced to the states. (d) A close-up view of controller 2’s path.

Table 5. Success rate (SCR) and average reaching time (ART) of different controllers.

d_c	0.3 *	0.4 *	0.5	0.6	0.7	0.8	0.9	1.0
SCR	0.517	0.930	0.968	0.941	0.851	0.714	0.585	0.568
ART (s)	43.437	34.983	27.194	22.340	19.606	18.705	19.146	17.710

* The time requirement for success was relaxed from 3000 steps to 5000 steps.

5.3.4. Comparison to Existing Methods

This section compares the proposed method to two other existing methods in terms of success rate, average reaching time, average absolute curvature, and the number of sharp turns. The first method to be compared uses A* as the path planner and the SE(3) control law proposed in [45] as the motion controller. This method discretizes the environment into cubes with side length 0.2 m, which are also known as voxels. It should be noted that the environment information of this algorithm is known in advance instead of being obtained

with sensors. The second method to be compared [47] is a hierarchical DRL method that uses a DQN network to generate waypoints based on the sensor information returned by 16 rangefinders. The waypoints then serve as the target points of the motion controller trained with PPO. The testing environment was a single-angle scenario, where the agent was initialized at $[0, 4.5, 2.5]$, and the target position was $[0, -4.5, 2.5]$. Twelve cylindrical obstacles with a radius of 0.4 m were randomly placed around the origin. The positions of these obstacles were determined by uniformly selecting the distances of their axes from the center within the range of $[0, 3]$ and $d_{\min} = 0.6$. The results are listed in Table 6, which also includes the performance of controller 3 in this environment. We can see that the proposed method outperformed the other two algorithms in terms of success rate and average reaching time, with the path slightly more oscillatory. Although path oscillation is an inherent problem with APF, it can be reduced to a large extent with the help of velocity information. After addressing this issue, the controller can navigate the QUAUV in a more safe and efficient manner.

Figure 11 illustrates the paths of the four methods in the same environment. It can be seen that the proposed method's path is smooth on a macro level and is far away from obstacles. The path of $A^* + SE(3)$ is mostly straight because the waypoints returned by A^* are exactly on the centers of the voxels. Figure 11c shows a similar path, in that there are only 16 possible directions to choose from when updating the waypoints. The path of controller 3, on the other hand, is the most oscillatory one.

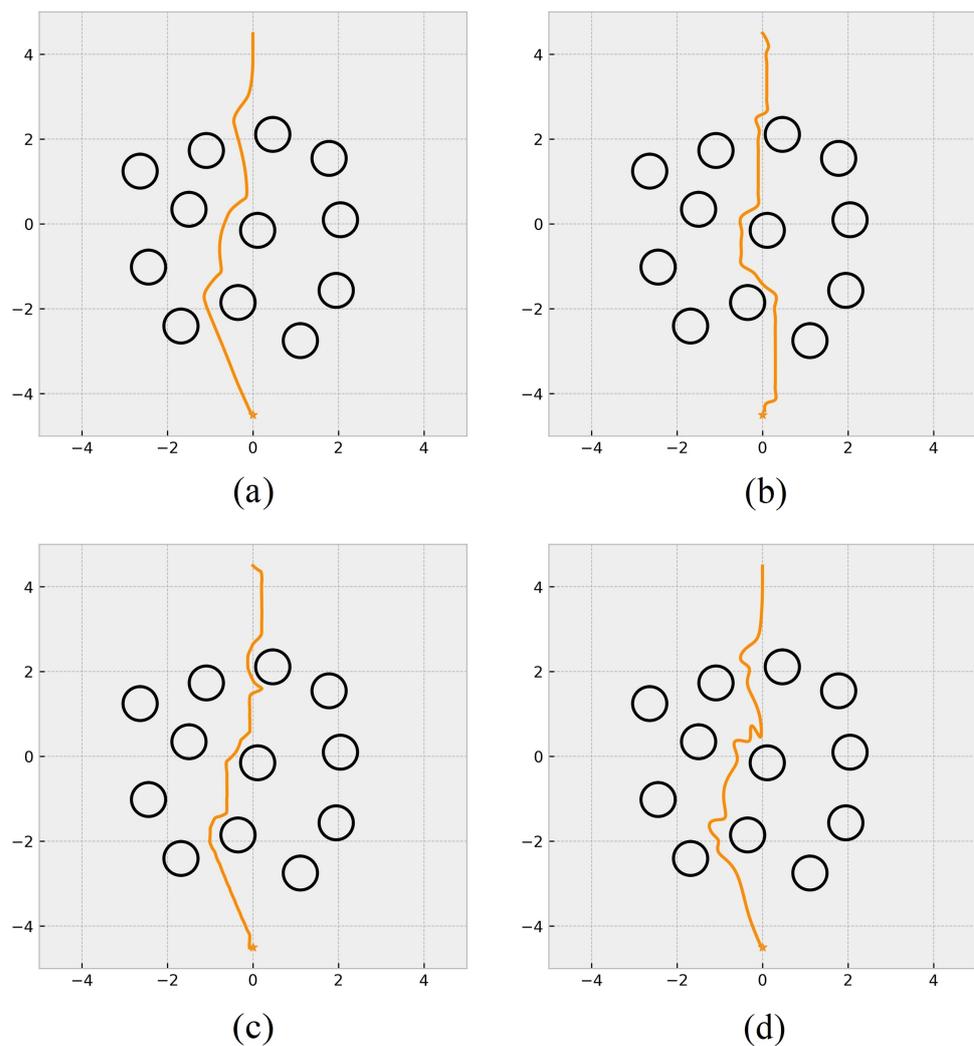


Figure 11. Performance comparison of the four methods in the same environment. (a) Proposed method. (b) $A^* + SE(3)$. (c) DQN + PPO. (d) Controller 3.

Table 6. Comparison of the proposed method to existing methods.

	APF + TD3	A* + SE(3)	DQN + PPO	Controller 3
SCR	0.983	0.969	0.885	0.761
ART (s)	15.312	19.458	36.694	18.249
$ \kappa $	7.121	4.794	6.610	11.426
sharp turns	0.651	0.480	0.431	1.112

6. Discussion

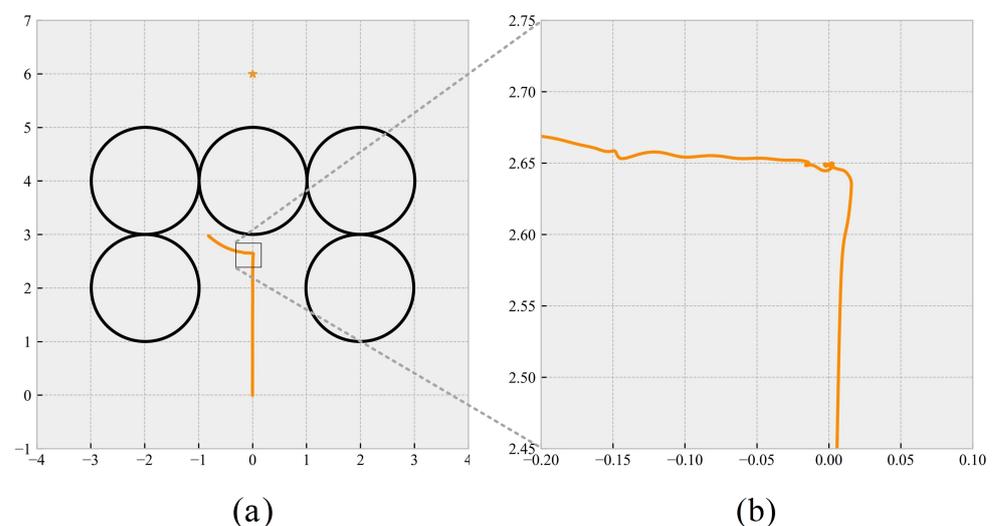
6.1. Limitations

Although the designed controller can achieve high success rates with moderate reaching time even in a dense environment, there is still room for improvement. We considered cases as failures when the agents were close to the targets but still outside the 0.1 m neighborhood. This issue is referred to as the GNRON problem, which is discussed in Section 2. Table 7 provides the number of GNRON cases for different controllers, where the success requirement is relaxed to entering the 0.3 m neighborhood of targets. It is evident that the success rates could have been higher if we had employed a strategy that actively addressed this problem. By introducing a damping term in the repulsive forces when the agents are in close proximity to the targets, this problem can be effectively resolved. Additionally, there are other limitations to the proposed method, such as the absence of explicit collision detection, leading to occasional collisions, and the hyperparameter d_c , which needs to be manually adjusted.

Another limitation of the proposed method is that it fails to escape local minima created by U-shaped obstacles, which is shown in Figure 12a. Moreover, when the QUAV's moving direction is almost collinear with the joint force, the path is usually not smooth, as shown in Figure 12b. One possible solution is to combine the proposed method with the vortex potential field method discussed in [36,37], since in this way the agent will move smoothly around the obstacle instead of heading directly towards it and spending a long time adjusting the path.

Table 7. The number of GNRON cases for different controllers.

Environments	(0.7, 15)	(0.7, 12)	(0.9, 12)	(0.9, 9)
Controller 1	19	12	10	8
Controller 2	20	10	11	3
Controller 3	6	4	4	1

**Figure 12.** (a) The QUAV failed to escape the local minimum caused by a U-shaped obstacle. (b) When the approaching angle is close to zero, the path is not smooth.

6.2. Applicability to Dynamic Environments

The proposed method has the potential to be transferred to dynamic environments since collision avoidance behavior is only determined by the repulsive forces updated at every time step. In fact, the preceding experiments have verified this ability because all the QUAVs are dynamic, generating repulsive forces to other QUAVs. As shown in Figure 13, when they move close to each other, the QUAVs are able to take actions to avoid collisions. This behavior resembles the cooperative collision avoidance mentioned in [36]. One significant difference between agents and obstacles is that the former ones usually actively take action to avoid collisions while the latter ones' behavior may be unpredictable. Therefore, avoiding collisions between moving QUAVs and moving obstacles is more challenging. One possible solution is to combine the current framework with reciprocal velocity obstacles [48] based on the current velocity information of the obstacles.

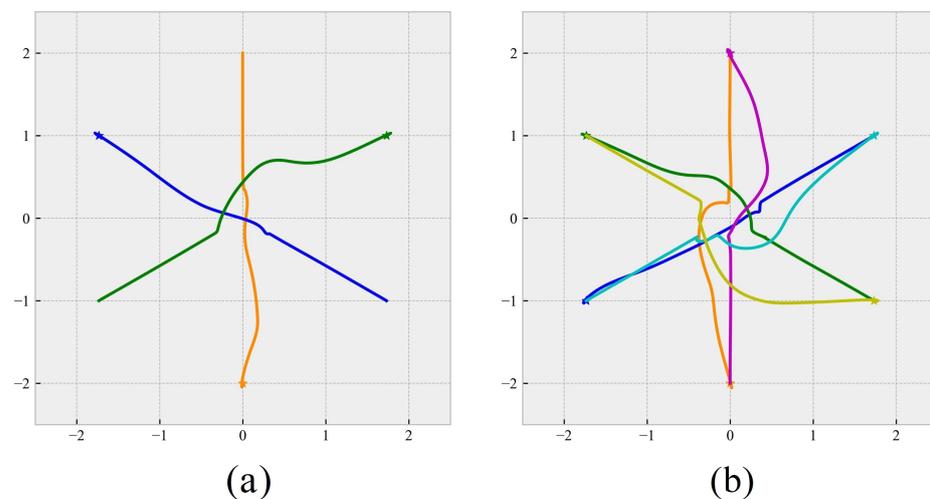


Figure 13. Cooperative collision avoidance. (a) The case with 3 agents. (b) The case with 6 agents.

6.3. Irregular Obstacles

In real applications, the shapes of obstacles may not be as simple as cylinders. Since this work assumes that the QUAV only detects nearby obstacles with sensors, it is impractical to generate a global potential field in advance. There are two simple ways to adapt this framework so as to make it applicable to environments with irregular obstacles. The first method, as shown in Figure 14a, is to replace the surface of the sensed part with smaller cylinders. In this way, the irregular obstacle's repulsive force equals the joint repulsive force of all cylindrical obstacles. The second method is to assume the part behind the sensed surface as solid and calculate its center of gravity. The distance to the obstacle is obtained as shown in Figure 14b. Some works deal with irregular obstacles in advanced ways. For example, Ge et al. [49] integrated the repulsive force of every part of the obstacle to calculate the ultimate repulsive force. This can model the obstacles more accurately, but it requires the QUAV to have prior global information of the environment. Guo et al. [50] used the closest point on the obstacle's surface to generate repulsive force, which is the same as our method when the obstacles are cylindrical or round. However, separate discussions are needed for obstacles with corners or edge points. This makes it difficult to achieve stable control using the Hessian matrix considering that its calculation will be even more complex under such circumstances.

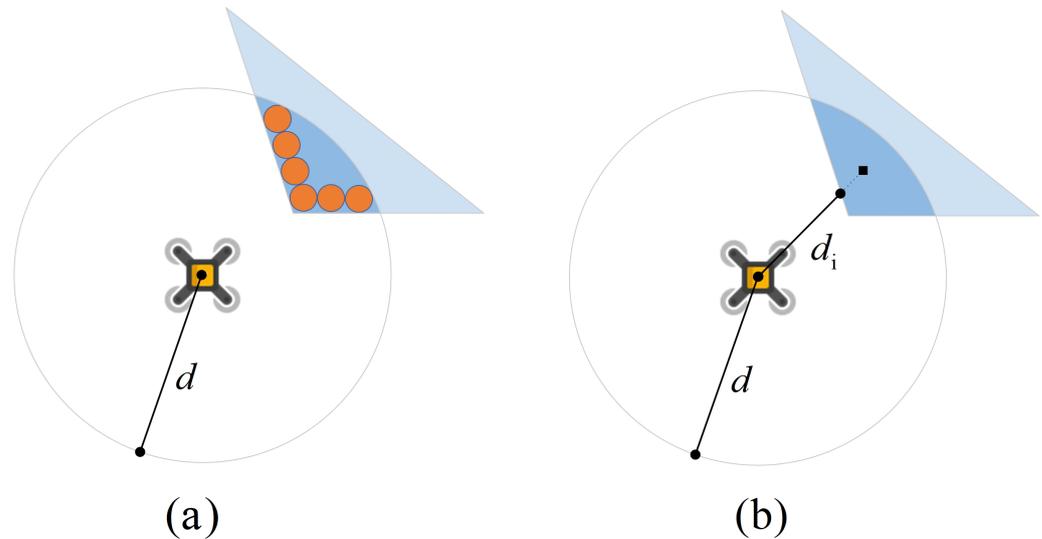


Figure 14. Two possible ways to model irregular obstacles, where the sensed parts are shown in dark blue. (a) The surface of the sensed part is replaced with smaller cylindrical obstacles, illustrated as orange circles. (b) Assuming the sensed part is a solid object.

7. Conclusions

In this study, we have proposed an efficient hierarchical controller that combines the use of an APF for path planning and obstacle avoidance with DRL motion controllers to generate rotor signals. Our approach differs from traditional APF methods in that the APF force is no longer directly applied to the QUAV but instead used to modify the position errors, effectively pushing the sensed target position. Additionally, we have introduced the calculation of the time-derivative of the APF force and incorporated a second-order term in the velocity error of the DRL controllers' inputs. This modification has successfully reduced oscillations that occur when an agent is surrounded by multiple obstacles. The oscillation technique not only ensures smoother paths but also significantly improves the robustness of the controller. Furthermore, we have demonstrated that canceling the projection of the target velocity on the attractive force when they are in the same direction is an effective method to prevent overshooting and enhance the success rate. Although this approach may slightly decrease the speed at which the target is reached, it is a reasonable compromise that prioritizes safety, especially in dense environments. We would like to emphasize the importance of the clipping factor d_c as a hyperparameter in our design. It is crucial to find an optimal value for d_c , as both excessively small and excessively large values can lead to significant performance degradation. Lastly, we discussed the limitations of the proposed method and provided some possible solutions. The control framework in this work can be applied to a real testing environment considering that it is based on real parameters of a QUAV. Potential challenges include the accumulated errors of the discrete system that generates outputs based on differences, a key point that needs to be addressed in future work.

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

Funding: This research was funded in part by the NNSFC&CAAC under Grant U2233209 and in part by the Natural Science Foundation of Sichuan, China, under Grant 2023NSFSC0484.

Data Availability Statement: Data are contained within the article.

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

References

1. Pugliese, L.D.P.; Guerriero, F.; Macrina, G. Using Drones for Parcels Delivery Process. *Procedia Manuf.* **2020**, *42*, 488–497. [[CrossRef](#)]
2. Shakhathreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634. [[CrossRef](#)]
3. Huang, Y.; Thomson, S.J.; Hoffmann, W.C.; Lan, Y.; Fritz, B.K. Development and Prospect of Unmanned Aerial Vehicle Technologies for Agricultural Production Management. *Int. J. Agric. Biol. Eng.* **2012**, *6*, 1–10.
4. Muchiri, G.N.; Kimathi, S. A Review of Applications and Potential Applications of UAV. In Proceedings of the Sustainable Research and Innovation Conference (SRI), Pretoria, South Africa, 20–24 June 2022; pp. 280–283.
5. Valsan, A.; Parvathy, B.; GH, V.D.; Unnikrishnan, R.S.; Reddy, P.K.; Vivek, A. Unmanned Aerial Vehicle for Search and Rescue Mission. In Proceedings of the 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 16–18 April 2020; pp. 684–687.
6. Silvagni, M.; Tonoli, A.; Zenerino, E.; Chiaberge, M. Multipurpose UAV for Search and Rescue Operations in Mountain Avalanche Events. *Geomat. Nat. Hazards Risk* **2017**, *8*, 18–33. [[CrossRef](#)]
7. Pinto, M.F.; Melo, A.G.; Marcato, A.L.; Urdiales, C. Case-based Reasoning Approach Applied to Surveillance System Using an Autonomous Unmanned Aerial Vehicle. In Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017; pp. 1324–1329.
8. Lv, M.; Wang, N. Distributed Control for Uncertain Multi-agent Systems with the Powers of Positive-odd Numbers: A Low-complexity Design Approach. *IEEE Trans. Autom. Control.* **2024**, *69*, 434–441. [[CrossRef](#)]
9. Wang, Y.; Dong, L.; Sun, C. Cooperative Control for Multi-player Pursuit-evasion Games with Reinforcement Learning. *Neurocomputing* **2020**, *412*, 101–114. [[CrossRef](#)]
10. Dijkstra, E.W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269–271. [[CrossRef](#)]
11. Kavraki, L.E.; Svestka, P.; Latombe, J.C.; Overmars, M.H. 1996. Probabilistic Roadmaps for Path Planning in High-dimensional Configuration Spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [[CrossRef](#)]
12. Elbanhawi, M.; Simic, M. Sampling-based Robot Motion Planning: A Review. *IEEE Access* **2014**, *2*, 56–77. [[CrossRef](#)]
13. Khatib, O. Real-time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [[CrossRef](#)]
14. Tang, L.; Dian, S.; Gu, G.; Zhou, K.; Wang, S.; Feng, X. A Novel Potential Field Method for Obstacle Avoidance and Path Planning of Mobile Robot. In Proceedings of the 2010 3rd International Conference on Computer Science and Information Technology (ICCSIT), Chengdu, China, 9–11 July 2010; pp. 633–637.
15. Han, H.; Cheng, J.; Xi, Z.; Lv, M. Symmetric Actor-critic Deep Reinforcement Learning for Cascade Quadrotor Flight Control. *Neurocomputing* **2023**, *559*, 126789. [[CrossRef](#)]
16. Han, H.; Xi, Z.; Cheng, J.; Lv, M. Obstacle Avoidance Based on Deep Reinforcement Learning and Artificial Potential Field. In Proceedings of the 9th International Conference on Control, Automation and Robotics (ICCAR), Beijing, China, 21–23 April 2023; pp. 215–220.
17. Han, H.; Xi, Z.; Lv, M.; Cheng, J. Acceleration of Formation Control Based on Hessian Matrix of Artificial Potential Field. In Proceedings of the 2023 42th Chinese Control Conference, Tianjin, China, 24–26 July 2023; pp. 5866–5871.
18. Han, H.; Cheng, J.; Xi, Z.; Yao, B. Cascade Flight Control of Quadrotors Based on Deep Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2022**, *7*, 11134–11141. [[CrossRef](#)]
19. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Man, Cybern.* **1968**, *4*, 100–107. [[CrossRef](#)]
20. Stentz, A. Optimal and Efficient Path Planning for Partially-known Environments. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation (ICRA), San Diego, CA, USA, 8–13 May 1994; pp. 3310–3317.
21. Farooq, M.U.; Ziyang, Z.; Ejaz, M. Quadrotor UAVs Flying Formation Reconfiguration with Collision Avoidance using Probabilistic Roadmap Algorithm. In Proceedings of the 2017 International Conference on Computer Systems, Electronics and Control, Dalian, China, 25–27 December 2017; pp. 866–870.
22. Ma'arif, A.; Rahmani, W.; Vera, M.A.M.; Nuryono, A.A.; Majdoubi, R.; Çakan, A. Artificial Potential Field Algorithm for Obstacle Avoidance in UAV Quadrotor for Dynamic Environment. In Proceedings of the 2021 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT), Online, 17–18 July 2021; pp. 184–189.
23. Koren, Y.; Borenstein, J. Potential Field Methods and Their Inherent limitations for Mobile Robot Navigation. In Proceedings of the 1991 International Conference on Robotics and Automation, Sacramento, CA, USA, 9–11 April 1991; pp. 1398–1404.
24. Kim, J.O.; Khosla, P. Real-time Obstacle Avoidance Using Harmonic Potential Functions. In Proceedings of the 1991 International Conference on Robotics and Automation, Sacramento, CA, USA, 9–11 April 1991; pp. 790–796.
25. Park, M.G.; Jeon, J.H.; Lee, M.C. Obstacle Avoidance for Mobile Robots Using Artificial Potential Field Approach with Simulated Annealing. In Proceedings of the 2001 IEEE International Symposium on Industrial Electronics, Pusan, Republic of Korea, 12–16 June 2001; pp. 1530–1535.
26. Wang, D.; Li, C.; Guo, N.; Song, Y.; Gao, T.; Liu, G. Local Path Planning of Mobile Robot Based on Artificial Potential Field. In Proceedings of the 2020 39th Chinese Control Conference, Shenyang, China, 27–29 July 2020; pp. 3677–3682.

27. Lai, D.; Dai, J. Research on Multi-UAV Path Planning and Obstacle Avoidance Based on Improved Artificial Potential Field Method. In Proceedings of the 2020 3rd International Conference on Mechatronics, Robotics and Automation (ICMRA), Shanghai, China, 16–18 October 2022; pp. 84–88.
28. Doria, N.S.F.; Freire, E.O.; Basilio, J.C. An Algorithm Inspired by the Deterministic Annealing Approach to Avoid Local Minima in Artificial Potential Fields. In Proceedings of the 2013 16th International Conference on Advanced Robotics, Montevideo, Uruguay, 25–29 November 2013; pp. 1–6.
29. Ge, S.S.; Cui, Y.J. New Potential Functions for Mobile Robot Path Planning. *IEEE Trans. Robot. Autom.* **2000**, *16*, 615–620. [[CrossRef](#)]
30. Ren, J.; McIsaac, K.A.; Patel, R.V. Modified Newton’s Method Applied to Potential Field-based Navigation for Mobile Robots. *IEEE Trans. Robot.* **2006**, *22*, 384–391.
31. Biswas, K.; Kar, I. On Reduction of Oscillations in Target Tracking by Artificial Potential Field Method. In Proceedings of the 2014 9th International Conference on Industrial and Information Systems (ICIIS), Gwalior, India, 15–17 December 2014; pp. 1–6.
32. Zhao, M.; Lv, X. Improved Manipulator Obstacle Avoidance Path Planning Based on Potential Field Method. *J. Robot.* **2020**, *2020*, 1701943. [[CrossRef](#)]
33. Zhang, W.; Xu, G.; Song, Y.; Wang, Y. An Obstacle Avoidance Strategy for Complex Obstacles Based on Artificial Potential Field Method. *J. Field Robot.* **2023**, *40*, 1231–1244. [[CrossRef](#)]
34. Zheng, S.; Luo, L.; Zhang, J. Non-oscillation Path Planning Based on Artificial Potential Field. In Proceedings of the IEEE International Conference on Control, Electronics and Computer Technology (ICCETC), Jilin, China, 28–30 April 2023; pp. 1225–1228.
35. Tran, H.N.; Shin, J.; Jee, K.; Moon, H. Oscillation Reduction for Artificial Potential Field Using Vector Projections for Robotic Manipulators. *J. Mech. Sci. Technol.* **2023**, *37*, 3273–3280. [[CrossRef](#)]
36. Martis, W.P.; Rao, S. Cooperative Collision Avoidance in Mobile Robots using Dynamic Vortex Potential Fields. In Proceedings of the International Conference on Automation, Robotics and Applications (ICARA), Abu Dhabi, United Arab Emirates, 10–12 February 2023; pp. 60–64.
37. Szczepanski, R. Safe Artificial Potential Field-Novels Local Path Planning Algorithm Maintaining Safe Distance from Obstacles. *IEEE Robot. Autom. Lett.* **2023**, *8*, 4823–4830. [[CrossRef](#)]
38. Meradi, D.; Benselama, Z.A.; Hedjar, R.; Gabour, N.E.H. Quaternion-based Nonlinear MPC for Quadrotor’s Trajectory Tracking and Obstacles Avoidance. In Proceedings of the International Conference on Advanced Electrical Engineering (ICAEE), Constantine, Algeria, 29–31 October 2022; pp. 1–6.
39. Valencia, D.; Kim, D. Quadrotor Obstacle Detection and Avoidance System Using a Monocular Camera. In Proceedings of the Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Singapore, 21–23 July 2018; pp. 78–81.
40. Gageik, N.; Benz, P.; Montenegro, S. Obstacle Detection and Collision Avoidance for a UAV with Complementary Low-cost Sensors. *IEEE Access* **2015**, *3*, 599–609. [[CrossRef](#)]
41. Yao, Q.; Zheng, Z.; Qi, L.; Yuan, H.; Guo, X.; Zhao, M.; Liu, Z.; Yang, T. Path Planning Method with Improved Artificial Potential Field—A Reinforcement Learning Perspective. *IEEE Access* **2020**, *8*, 135513–135523. [[CrossRef](#)]
42. Xing, T.; Wang, X.; Ding, K.; Ni, K.; Zhou, Q. Improved Artificial Potential Field Algorithm Assisted by Multisource Data for AUV Path Planning. *Sensors* **2023**, *23*, 6680. [[CrossRef](#)]
43. Panerati, J.; Zheng, H.; Zhou, S.; Xu, J.; Prorok, A.; Schoellig, A.P. Learning to Fly—A Gym Environment with Pybullet Physics for Reinforcement Learning of Multi-agent Quadcopter Control. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 28–30 September 2021; pp. 7512–7519.
44. Lv, M.; Ahn, C.K.; Zhang, B.; Fu, A. Fixed-time Anti-saturation Cooperative Control for Networked Fixed-wing Unmanned Aerial Vehicles Considering Actuator Failures. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 8812–8825. [[CrossRef](#)]
45. Goodarzi, F.; Lee, D.; Lee, T. Geometric Nonlinear PID Control of a Quadrotor UAV on SE(3). In Proceedings of the 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 3845–3850.
46. Bitcraze. *Crazyflie 2.0*. Available online: <https://www.bitcraze.io/products/old-products/crazyflie-2-0/> (accessed on 2 January 2024).
47. Xi, Z.; Han, H.; Zhang, Y.; Cheng, J. Autonomous Navigation of QUAVs Under 3D Environments Based on Hierarchical Reinforcement Learning. In Proceedings of the 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 24–26 July 2023; pp. 4101–4106.
48. Van den Berg, J.; Lin, M.; Manocha, D. Reciprocal Velocity Obstacles for Real-time Multi-agent Navigation. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA), Pasadena, CA, USA, 19–23 May 2008; pp. 1928–1935.
49. Ge, S.S.; Liu, X.; Goh, C.H.; Xu, L. Formation Tracking Control of Multiagents in Constrained Space. *IEEE Trans. Control Syst. Technol.* **2015**, *24*, 992–1003. [[CrossRef](#)]
50. Guo, Y.; Chen, G.; Zhao, T. Learning-based Collision-free Coordination for a Team of Uncertain Quadrotor UAVs. *Aerosp. Sci. Technol.* **2021**, *119*, 107127. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.