aerospace

MDPI

Article

Model-Reference Reinforcement Learning for Safe Aerial Recovery of Unmanned Aerial Vehicles

Bocheng Zhao [1], Mingying Huo [1,*], Ze Yu [1], Naiming Qi [1] and Jianfeng Wang [2]

[1] School of Astronautics, Harbin Institute of Technology, Harbin 150001, China; 22b918135@stu.hit.edu.cn (B.Z.); yuze@hit.edu.cn (Z.Y.); qinmok@163.com (N.Q.)
[2] Avic Aerodynamics Research Institute, Shenyang 110034, China; wangjianfeng_avic@163.com
* Correspondence: huomingying@hit.edu.cn

Abstract: In this study, we propose an aerial rendezvous method to facilitate the recovery of unmanned aerial vehicles (UAVs) using carrier aircrafts, which is an important capability for the future use of UAVs. The main contribution of this study is the development of a promising method for online generation of feasible rendezvous trajectories for UAVs. First, the wake vortex of a carrier aircraft is analyzed using the finite element method, and a method for establishing a safety constraint model is proposed. Subsequently, a model-reference reinforcementearning algorithm is proposed based on the potential function method, which can ensure the convergence and stability of training. A combined reward function is designed to solve the UAV trajectory generation problem under non-convex constraints. The simulation results show that, compared with the traditional artificial potential field method under different working conditions, the success rate of this method under non-convex constraints is close to 100%, with high accuracy, convergence, and stability, and has greater application potential in the aerial recovery scenario, providing a solution to the trajectory generation problem of UAVs under non-convex constraints.

Keywords: aerial UAV recovery; wake vortices; safety constraint model; model-reference reinforcementearning

check for updates

Citation: Zhao, B.; Huo, M.; Yu, Z.; Qi, N.; Wang, J. Model-Reference Reinforcement Learning for Safe Aerial Recovery of Unmanned Aerial Vehicles. *Aerospace* **2024**, *11*, 27. https://doi.org/10.3390/aerospace11010027

Academic Editor: Gokhan Inalhan

Received: 10 November 2023
Revised: 24 December 2023
Accepted: 26 December 2023
Published: 27 December 2023

Copyright: © 2023 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

Aerial recovery is a new technology that can quickly and efficiently capture unmanned aerial vehicle (UAV) in the air when there are no suitable platforms foranding onand or sea [1,2]. Swarm UAVs areow-cost and reusable devices that can be quickly deployed and have recently attracted considerable research interest [3]. A carrier aircraft transports a swarm of UAVs to the edge of a combat zone and releases them. Swarm UAVs then return to the carrier aircraft after completing their tasks. Aerial recovery can improve combat performance, reuse rate, and flexibility of UAVs. However, this involves docking the probe of the UAVs with a cable-drogue assembly towed by the carrier aircraft and then pulling it into the cabin with a winch. The idea of "aerial recovery" has gained interest since theaunch of the Defense Advanced Research Projects Agency (DARPA)'s "Gremlins" project (2016) [4]. On 29 October 2021, DARPA successfully recovered an X-61 "Gremlin" drone in mid-air for the first time [5,6]. Studies have shown that wake vortices have a significant impact on the flight safety of aircrafts [7]. If a following aircraft breaks into the wake area of aeading aircraft, the aerodynamic force of the following aircraft is significantly disturbed [8].

In this case, the main challenges of UAV aerial recovery can be summarized as follows: (1) describing the unsafe area for rendezvous and docking missions while accounting for the effects of wake vortices and turbulence from the carrier aircraft; (2) obtaining closer proximity to the carrier while avoiding unsafe areas, which will reduce theength of the recovery rope, thereby reducing the difficulty of dragging the drone back into the cabin

and increasing efficiency; (3) the development of an online real-time control algorithm that ensures safety and accessibility.

Current research methods for wake vortices can be roughly divided into experimental measurement methods, theoretical analyses, and numerical simulation methods based on computational fluid dynamics. The experimental methods of wake simulation include artificial simulation of experimental environment devices represented by on-site observation methods, such as wind tunnels [9]. However, these methods are time consuming, expensive, andack flexibility. In addition, some characteristics of the wake vortex flow field cannot be accurately resolved and captured through experiments owing to theimitations of the data collection and measurement methods. The theoretical analysis methods are primarily used for rapid modeling and simulation of wake vortices [10]. They cannot satisfy the computational requirements of complex shapes and high precision. Numerical simulations are based on the theory of computational fluid dynamics. With the rapid development of high-performance computer technology, these methods have been widely used in aircraft flow fields, such asarge eddy simulations (LES) and simulations based on the Reynolds-averaged Navier–Stokes (RANS) theory [11,12]. Numerical simulation methods are widely recognized for their ability to simulate wake flows in a cost-effective, convenient, and scalable manner.

Some studies have been conducted on the aerial recovery of UAVs [4,13]. Different theoretical approaches have been explored to solve the guidance and control problem [14]. These can be divided into three categories: optimization, hybrid, and analytical methods [15,16]. Examples include direct collocation [17], indirect (variational) methods [18], rapidly exploring random trees (RRTs) [19], sequential convex optimization [20], waypoint following using a PD or H-infinity controller [21], artificial potential functions (APFs) [22], second-order cone programming [23], model predictive control (MPC) [24], and adaptive control techniques [25]. However, many of these methods are offline, time-consuming, and unsuitable for non-convex optimization problems, whereas deep reinforcementearning (RL), which enables autonomous real-time, collision-free rendezvous, and proximity operations, has the potential to overcome these disadvantages [26]. One method of using deep RL is to obtain solutions for systems with complex dynamics and constraints [27]. It has the key benefit of performing well for problems that do not have a suitable nonlinear programming formulation, can generate real-time and dynamic solutions for online applications [28], and has been used in applications in real-world UAV trajectory generation and control, achieving the highest performance far beyond that of humans [29]. This can also be beneficial when the underlying problem is stochastic [30]. However, solving the potential reward sparseness when RL is used to solve UAV navigation problems is difficult [31]. In the early exploration stage of training, UAVs without expert experience do not always receive rewards after each action, rendering it difficult for the algorithm to converge [32]. Existing methods primarily include exploiting intrinsic curiosity [33,34] or information gain [35]; however, these methods mayead to performance degradation. Another method is to imitate expert experience [36]; however, the upperimit of the algorithm performance may be affected by experts, and expert data may be difficult to obtain.

In this study, we considered the aerial recovery of UAV as a reach-avoidance problem to establish a safety constraint model by analyzing the flow field around the carrier aircraft. We present an online trajectory generation method for UAVs in a three-dimensional (3D) space. The contributions of this study can be summarized as follows:

(1) The equipotential airflowines of the flow field around a carrier aircraft were depicted by finite element analysis, and a method for establishing safe constraints was proposed.

(2) A new potential function-based reinforcementearning control method (PRL) was proposed, utilizing the advantages of a model-reference RL algorithm. It can be used for trajectory generation under non-convex constraints, satisfying the safety constraints while guaranteeing convergence. This may be more efficient than RL algorithmsearned from scratch.

(3) A reward function based on a potential function was designed to ensure security while reducing the impact of sparse rewards. Simultaneously, as the training of reinforcementearning is completed in an offline environment, the call model can meet the requirements of online and real-time response in the actual application process. In addition, the convergence and stability of PRL algorithm are proved and the feasibility of the algorithm is verified through simulation experiments.

2. Problem Formulation

When performing an air-recovery mission, it is assumed that the carrier aircraft is flying at the intended cruise speed. The UAV moves to the recovery drogue, which is connected to the carrier through a cable and remains relatively stationary with the carrier. During this process, the UAV must avoid the flow field in an unsafe area around the carrier aircraft. When rendezvous occurs and the relative speed is similar, recovery drives dock with the UAV and finally return to the cabin. In this study, both UAV and drogue are regarded as mass points. A schematic diagram is shown in Figure 1.

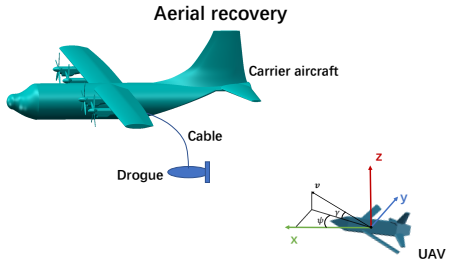


Figure 1. Schematic diagram of UAV aerial recovery.

2.1. Dynamic Model for The UAV

In this study, we considered a simplified dynamic model for a UAV [37], which can be written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{v} \\ \dot{\gamma} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} v\cos(\psi)\cos(\gamma) \\ v\sin(\psi)\cos(\gamma) \\ v\sin(\gamma) \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{1}$$

where (x, y, z) is the position of UAV in space, v is the ground speed, ψ is the yaw angle, γ is the flight path angle, and $\boldsymbol{u}_t = [u_1, u_2, u_3]^\top$ is the control of UAV.

As the drogue is considered to be relatively stationary with the carrier, we use $\boldsymbol{p}_{Dt} = [x_D, y_D, z_D]^\top$ to express the state vector of its position $\boldsymbol{p}_{Ct} = [x_C, y_C, z_C]^\top$ to express the position of the carrier and use $\boldsymbol{p}_{Ut} = [x_U, y_U, z_U]^\top$ to express the position of the UAV. $\boldsymbol{v}_{Ut} = [\dot{x}_U, \dot{y}_U, \dot{z}_U]^\top$, $\boldsymbol{v}_{Dt} = [\dot{x}_D, \dot{y}_D, \dot{z}_D]^\top$ is the velocity of UAV and the velocity of the drogue at time step t, respectively. The speed deviation can be expressed as $\boldsymbol{e}_v = \boldsymbol{v}_{Dt} - \boldsymbol{v}_{Ut} = [e_{vx}, e_{vy}, e_{vz}]^\top$, the flight path angle error between the drogue and the UAV is defined as $e_\gamma = \gamma_U - \gamma_D$, and the yaw angle error between them is defined as $e_\psi = \psi_U - \psi_D$. The position of the UAV relative to that of the drogue is defined as $\boldsymbol{d}_{Dt} = \boldsymbol{p}_{Dt} - \boldsymbol{p}_{Ut} = [e_x, e_y, e_z]^\top$.

2.2. Hazard Area Concept

The wake vortex created by the movement through the carrier can cause a variety of responses, as the air circulation can causearge rolling torques on the aircraft, which can be dangerous. Air swirling in the wake vortex has a tangential velocity that depends on its relative distance from the vortex center. A vortex has a core in which the tangential velocity increasesinearly with the radius. Outside the vortex core, the tangential velocity begins to decrease due to an inverse relationship with the radius. In order to simplify this problem,

an area can be defined around the wake vortex, which is called the Hazard area. Outside this area, tangential velocityess than V_0 pose no danger to aircraft. It should be noted that the tolerance of tangential velocity is different for an aircraft with different aerodynamic parameters. The tangential velocity can be expressed as

$$\begin{aligned} V_\omega &= \omega y_\omega, y_\omega \le \alpha \\ V_\omega &= \frac{\Gamma}{2\pi y_\omega}, y_\omega > \alpha \end{aligned} \tag{2}$$

where α is the core radius, ω is the angular velocity, y_ω is theocation of the wing relative to the vortexocation, and Γ is the initial vortex strength. It is related to the weight, speed, and altitude of the aircraft generating the vortex and can be expressed as

$$\Gamma_0 = \frac{4M_C}{\rho v_C b_C \pi} M_c \tag{3}$$

where M_C, v_C, b_C denote the mass, speed, and wingspan of the carrier aircraft inevel flight, respectively. This means that the initial strength of the vortex can be treated as a constant. Theocal variation in the aircraft's angle of attack caused by the eddy's current velocity field causes a rolling moment around the centerline of the wing and affects the flight of the aircraft [7], particularly for the outer region of the wake vortex, which is relevant for sizing the hazard zone because the core region must be avoided in any case [38].

The NASA Ames Research Center utilizes a numerical integration technique known as the strip theory, which uses the Biot–Savartaw to determine the induced velocity normal to the surface [39]. It determines the final rolling moment coefficient induced by the eddy currents to find aircraft danger, indicating that the numerical simulation method is feasible for establishing an unsafe area of the aircraft wake vortex. The rolling moment coefficient can be expressed as

$$C_\omega = \frac{2}{TS_C b_U} \int_0^{\frac{b_U}{2}} y_\omega C_{l_\alpha} \frac{V_\omega}{V_\infty} Tc \mathrm{d}y_\omega \tag{4}$$

where T is the dynamic pressure, S_C is the wing area of the carrier aircraft, b_U is the span of the wake-penetrating aircraft, C_{l_α} is the wingift curve slope, V_∞ is the freestream velocity, and c is the cordength.

The simplified danger zone can help overcome the wake vortex problem. As the distance between the aircraft and the wake vortex increases, the impact on the aircraft decreases. The spatial characteristics of this phenomenon indicated the formation of a dangerous space around the wake vortex. Avoiding these areas enables safe and undisturbed flight operations [40]; thus, it can be assumed that outside the hazardous area, vortices pose no danger to the aircraft [38]. The dangerous area is based on the vortex-induced roll coefficient, which is caused by the tangential velocity. Comparing the vortex-induced roll factor to the maximum coefficient of roll supplied by the airplane control system can determine the safety of a UAV flight [7]. In this study, the non-safe area was determined using numerical analysis. When the tangential velocity of the wake vortex wasess than $V_\omega = V_0$, the vortex-induced roll coefficient was smaller than the roll coefficient provided by the aircraft control system, as described in detail in Section 3.

3. Establishment of Safety Constraint Model

3.1. Finite Element Analysis

We consider a carrier aircraft with a wingspan of 40 m as an example to perform finite element analysis using ANSYS FLUENT 2021 R2. The transport aircraft model used in the experiment is shown in Figure 2. An area of block-oriented inversion encryption (BOI) was set near the aircraft to refine the grid.

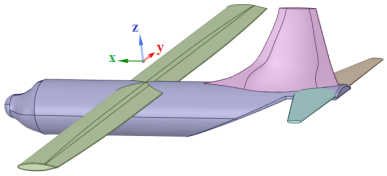
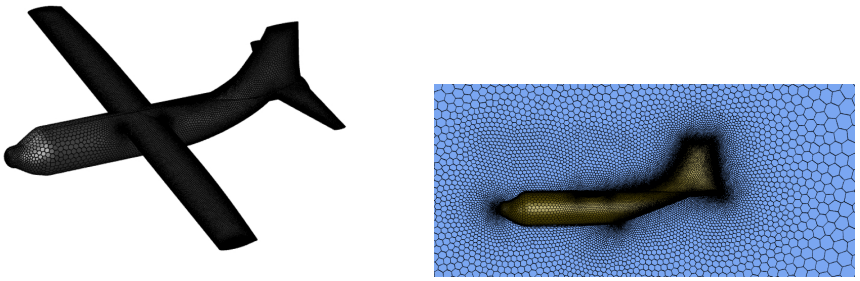


Figure 2. Model of Carrier Aircraft.

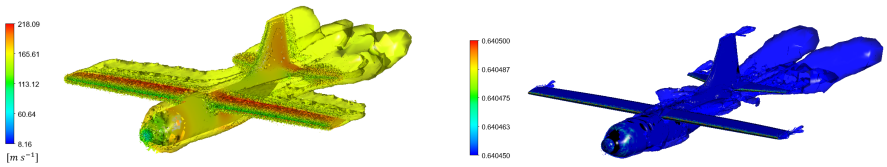
A watertight geometric process was applied to divide the mesh. The minimum size of the surface grid, the maximum size, the grid of the encrypted area, and the growth rate were set to 20 mm, 10,000 mm, 500 mm, and 1.1, respectively. To capture the surface accurately, we used curvature and proximity methods to identify the curvature and microstructure. *far* is set as the pressure far-field boundary condition, *sym* is set as the symmetrical surface boundary condition, and the other boundaries are set as *wall* boundaries. Finally, the volume grid is divided, and a polyhedral grid with a maximum grid size of 10,000 mm is selected. Simultaneously, a boundary–layer grid is applied to the aircraft's surface. The final grid division results are shown in Figure 3.



(**a**) Aircraft meshing (**b**) Flow field meshing

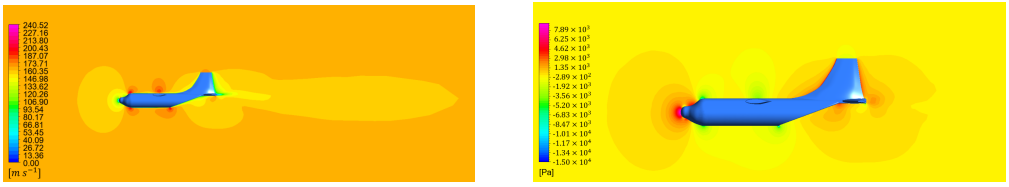
Figure 3. Meshing results.

In this study, the density-based steady-state solver was selected as the solution, the turbulence model was selected as the reliable k-e model, the energy equation was activated, the air density was set as an ideal gas, the viscosity was set as Sutherland, and the boundary condition was the pressure far field. This work analyzes the trailing vortex condition of the carrier aircraft at cruising altitude and cruising speed, and carries out aerial recovery of the fixed wing UAV under these conditions. The flight surface speed is 580 km/h and the flight altitude is 6 km. After the calculation, the Mach number was 0.51, the temperature was 249.18 K, and the far-field pressure was 47,217.5 Pa. The numerical solution method used in this study is a coupled algorithm in which the discrete format is second-order upwind. Theift and drag were monitored during the calculation until convergence. The flow field around a carrier aircraft is shown in Figure 4. Figure 5a shows the velocity contours and Figure 5b shows the pressure contours on the symmetry plane. Figure 6 shows the wake vortex cloud diagrams of different flow field sections, and the vector diagram of wake vortex tangential velocity in different flow field sections is shown in Figure 7.



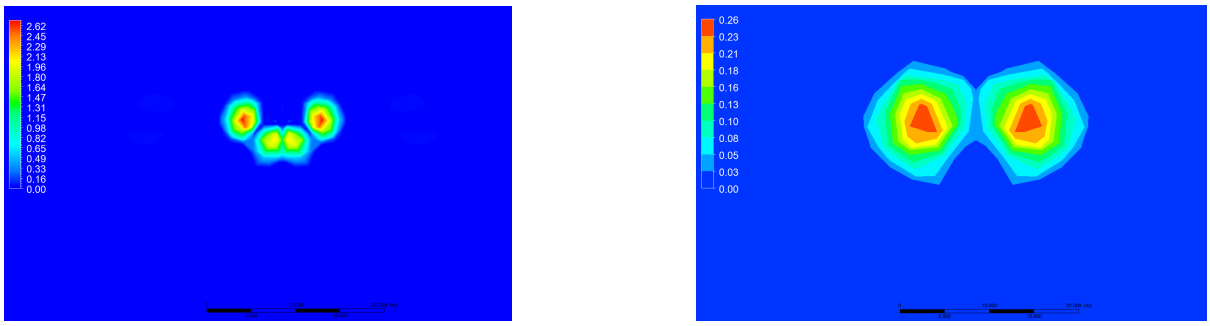
(**a**) Vortex core region of velocity (**b**) Vortex core region of swirling strength

Figure 4. Vortex core region. Velocity is defined using the vorticity method in (**a**) and the velocity swirling strength is defined using the swirling method in (**b**).



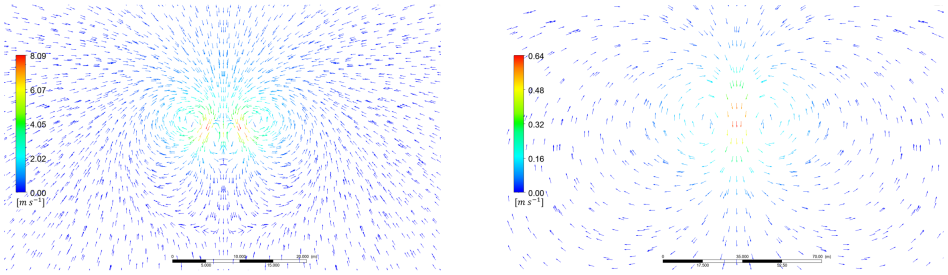
(**a**) Velocity contours on the symmetrical plane (**b**) Pressure contours on the symmetrical plane

Figure 5. Velocity and pressure distribution cloud map on the side view.



(**a**) Cloud map of wake vorticity at 30m behind the wing (**b**) Cloud map of wake vorticity at 50m behind the wing

Figure 6. Cloud map of wake vorticity.



(**a**) (**b**)

Figure 7. Tangential velocity vector diagram of vortex with different flow field sections. (**a**) Velocity vector diagram of flow field section at 30 m behind the wing; (**b**) velocity vector diagram of flow field section at 50 m behind the wing.

3.2. Establishment of the Safety Constraint Model

Because the ideal docking state of aerial recovery is performed on the symmetry plane (*xoz* plane) of the carrier aircraft, we used a multi-segment second-order Bezier curve with n segments to establish safety constraints in this plane. Figures 6 and 7 illustrate that the wake vorticity and the corresponding tangential velocity vector of the vortex center decrease as the wake vortex gradually moves away from the carrier aircraft. In addition, owing to the influence of the separation flow created by theow-pressure areas around the carrier, there are changes in the flow velocity (as shown in Figure 5), where the tangential velocity vector isess than $V_\omega = V_0$, and the corresponding points in the flow field are projected onto the symmetry plane to obtain the contourines at the points farthest from the aircraft in all directions. We establised a polar coordinate system with the center of the wing as the origin and the speed direction of the carrier machine in the x direction, and set n to 5 to create safety constraints, as shown in Figure 8, where points $N_a(l_a, \beta_a)$ are the control points of the second-order Bezier curve and a is an integer from 0 to $2n$. Notably, the polar coordinate system is only established in the symmetry plane to facilitate the expression of safety constraints, and the conversion with the Cartesian coordinate system will not be expanded in detail.

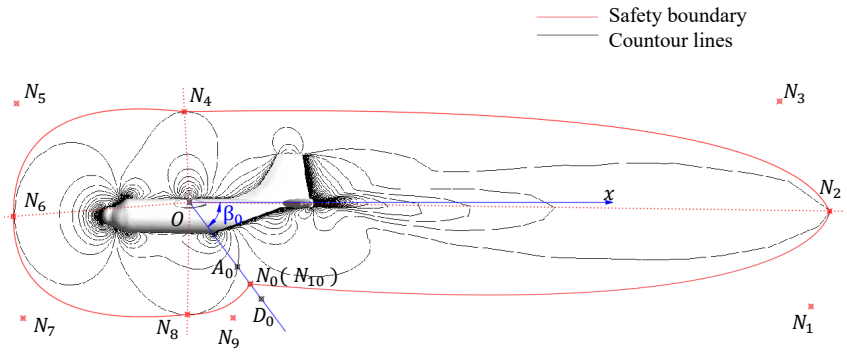


Figure 8. Safety constrains for Aerial Recovery, where N_0 (N_{10}), N_2, N_4, N_6, N_8 are the points farthest from the carrier aircraft in this area, respectively; A_0 is the intersection point of the ray in this direction and the contourine.

We define a positive integer h that satisfies $h \leq n$, and points $M_j(l_j, \beta_j)$ on the boundary of the constraints can be expressed as

$$\begin{aligned} l_j &= \sum_{i=0}^{2} l_{2h-i} \binom{2}{i} b^{2-i}(1-b)^i \\ \beta_j &= \sum_{i=0}^{2} \beta_{2h-i} \binom{2}{i} b^{2-i}(1-b)^i \end{aligned} \tag{5}$$

where $b \in [0, 1)$, $\beta_{2h-2} < \beta_j < \beta_{2h}$.

The coordinates of the recovery target point (the point where the recovery drogue isocated) are determined as $D_0(l_{D_0}, \beta_0)$, the coordinates of the point on the equipotentialine in this direction are determined as $A_0(l_{A_0}, \beta_0)$, and the parameters in Figure 8 are set to satisfy the following conditions:

$$\begin{aligned} \beta_0 &= \arctan \frac{2m_D g}{C\rho S_D {v_d}^2} - \frac{\pi}{2} \\ \frac{l_0}{l_{A_0}} &= k_1 \\ \frac{l_{D_0}}{l_0} &= k_2 \end{aligned} \tag{6}$$

where m_D is the mass of the recovery drogue, g is gravity, C is the air resistance coefficient, ρ is the air density, S_D is the windward area of the drogue, v_d is the relative velocity of the drogue and air, and k_1 and k_2 are the expansion coefficients, usually greater than 1.

4. PRL Algorithms

In the UAV navigation problem, there are situations in which rewards are sparse, and only when the UAV completes the target task can it obtain a nonzero reward [32]. The distance between the two aircrafts always increases further in the early stages of training, owing to the high speed of the carrier aircraft. Hence, it is difficult for the UAV to obtain rewards,eading to a poor convergence of the algorithm. This difficulty is particularly evident in this problem, where at the early stage of exploration, owing to aack of expert experience and understanding of the environment, the agent will randomly choose actions, and it will take aong time to reach the goal or fail to converge at all. To solve the above problems, we propose a potential function reinforcementearning method thatearns the potential function by interacting with the environment, ensuring convergence without expert experience while avoiding the problems of being unable to reach and falling into theocal optimum.

First, the corresponding potential field function is constructed based on the hazard area established previously and the position of the carrier aircraft. By designing the corresponding reward function, the output value of the action network in the reinforcementearning algorithm is used to adjust the size of the potential function to make the UAV move toward the target position, and the purpose of avoiding the hazard area is achieved at the same time. The specific details of the algorithm will be introduced in detail in this section.

4.1. Construction of Potential Function

The APF method uses potential field gradients comprising attractive and repulsive potentials to derive the necessary control inputs to achieve the desired or target positions [22,41]. Considering this idea combined with the safety boundary model established in Section 3.2, we determine a potential field function for aerial recovery. The attractive potential τ_a can be expressed as

$$\tau_a = \frac{1}{2}\boldsymbol{d}_{Dt}^{\top}\boldsymbol{W_a}\boldsymbol{d}_{Dt} \tag{7}$$

where $\boldsymbol{W_a} \subset \mathbb{R}^{3\times3}$is a symmetric positive-definite attractive potential shaping matrix.

The repulsive potential τ_b can be expressed as

$$\tau_b = \frac{\boldsymbol{d}_{Dt}^{\top}\boldsymbol{W_b}\boldsymbol{d}_{Dt}}{2\exp[\boldsymbol{d}_{St}^{\top}\boldsymbol{W_c}\boldsymbol{d}_{St} - 1]} \tag{8}$$

where $\boldsymbol{W_b}, \boldsymbol{W_c} \subset \mathbb{R}^{3\times3}$ is a positive-definite attractive potential shaping matrix and $\boldsymbol{d}_{St} = \boldsymbol{p}_{Ut} - \boldsymbol{p}_{St}$ is the position of the UAV relative to the target safety boundary constraint inertial position, $\boldsymbol{p}_{St}$.

From a geometric perspective, shaping matrix $\boldsymbol{W}$ can be regarded as theength, width, and height of a potential function. The total potential function τ_{sum} is defined as the sum of the attractive and repulsive potential functions as follows:

$$\tau_{\text{sum}} = \tau_a + \tau_b \tag{9}$$

4.2. Markov Decision Process

Generally, the RL problem is formalized using a discrete-time Markov decision process (MDP): $= (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \delta)$. $\boldsymbol{s}_t \in \mathcal{S} \subset \mathbb{R}^{\dim(\boldsymbol{s})}$ is the state vector at time t and $\mathcal{S}$ denotes the state space. Then, the agent takes an action $\boldsymbol{a}_t \in \mathcal{A} \subset \mathbb{R}^{\dim(\boldsymbol{a})}$ according to a stochastic policy/controller $\pi(\boldsymbol{a}_t \mid \boldsymbol{s}_t)$. The state transition probability $\mathcal{P}$ defines the transition probability. In the problem studied herein, the state of the next step can be updated by inputting the state and action of the current step into the model. We assumed that the input and state data can be sampled from the system (1) at discrete time steps due to the fact that RL trains the policy network from data samples. Based on MDP, an actor–critic algorithm was proposed as theearning control framework. The detailed derivations are discussed below.

4.3. State and Action Settings

Let $\boldsymbol{s}_t = [e_x, e_y, e_z, e_{vx}, e_{vy}, e_{vz}]^{\top} \subset \mathbb{R}^6$ be the state of the system at step t and the actions of RL as the coefficients of the potential function established in (9). The action of system at step t is defined as $\boldsymbol{a}_t = [\boldsymbol{a}_t^a, \boldsymbol{a}_t^b]^{\top}$. Because a higher precision control is desired at the end positions, a continuous feedback control rate is chosen and defined as

$$\boldsymbol{u}_t = \boldsymbol{K}_a(\boldsymbol{u}_t + \boldsymbol{a}_t^a\nabla\tau_a + \boldsymbol{a}_t^b\nabla\tau_b) \tag{10}$$

where $\boldsymbol{K}_a \subset \mathbb{R}^{3\times3}$ is the positive gain matrix and $\nabla\tau_a$ can be expressed as

$$\nabla\tau_a = \boldsymbol{d}_{Dt}\boldsymbol{W_a} \tag{11}$$

$\nabla\tau_b$ can be expressed as

$$\nabla\tau_b = \exp[1 - \boldsymbol{d}_{St}^{\top}\boldsymbol{W_c}\boldsymbol{d}_{St}](\boldsymbol{W_b}\boldsymbol{d}_{Dt} - (\boldsymbol{d}_{Dt}^{\top}\boldsymbol{W_b}\boldsymbol{d}_{Dt})\boldsymbol{W_c}\boldsymbol{d}_{St}) \tag{12}$$

4.4. Setup of the RL

For standard RL, the objective is to maximize an expected accumulated return described by a value function $V_\pi(s_t)$ with

$$V_\pi(\boldsymbol{s}_t) = \sum_{t}^{\infty}\sum_{\boldsymbol{a}_t}\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)\sum_{\boldsymbol{s}_{t+1}}\mathcal{P}(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{a}_t)(R_t + \delta V_\pi(\boldsymbol{s}_{t+1})) \tag{13}$$

where $\mathcal{P}(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{a}_t)$ is theikelihood of the next state $\boldsymbol{s}_{t+1}$, $R_t = R(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is the reward function, $\delta \in [0, 1)$ is a constant discount factor, and $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$ is called control policy in RL. Policy $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$ is the probability of choosing an action $\boldsymbol{a}_t \in \mathcal{A}$ at a state $\boldsymbol{s}_t \in \mathcal{S}$. In this paper, a Gaussian policy is used, which is

$$\pi(\boldsymbol{a}|\boldsymbol{s}) = \mathcal{N}(\boldsymbol{a}(\boldsymbol{s}), \sigma) \tag{14}$$

where $\mathcal{N}(\cdot,\cdot)$ stands for a Gaussian distribution with $\boldsymbol{a}(\boldsymbol{s})$ as the mean and σ as the covariance matrix which controls the exploration during theearning process. The Q-function used to evaluate the action can be expressed as

$$Q_\pi(\boldsymbol{s}_t, \boldsymbol{a}_t) = R_t + \delta\mathbb{E}_{\boldsymbol{s}_{t+1}}[V_\pi(\boldsymbol{s}_{t+1})] \tag{15}$$

where $\mathbb{E}_{\boldsymbol{s}_{t+1}}[\cdot] = \sum_{\boldsymbol{s}_{t+1}} P_{t+1|t}[\cdot]$ is an expectation operator.

By introducing the entropy of policy $\mathcal{H}(\pi(\boldsymbol{a}_{t+1}|\boldsymbol{s}_{t+1})) = -\sum_{\boldsymbol{a}_t}\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)\ln(\pi(\boldsymbol{a}_t|\boldsymbol{s}_t))$ and the temperature parameter η [42], the modified Q-function can be expressed as

$$Q_\pi(\boldsymbol{s}_t, \boldsymbol{a}_t) = R_t + \delta\mathbb{E}_{\boldsymbol{s}_{t+1}}[V_\pi(\boldsymbol{s}_{t+1}) + \eta\mathcal{H}(\pi(\boldsymbol{a}_{t+1}|\boldsymbol{s}_{t+1}))] \tag{16}$$

There are two steps that needed to be executed repeatedly in theearning process: policy evaluation and policy improvement. In the policy evaluation, the Q-value in Equation (16) is computed by applying a Bellman operation $Q_\pi(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathcal{T}^\pi Q_\pi(\boldsymbol{s}_t, \boldsymbol{a}_t)$ where

$$\mathcal{T}^\pi Q_\pi(\boldsymbol{s}_t, \boldsymbol{a}_t) = R_t + \delta\mathbb{E}_{\boldsymbol{s}_{t+1}}\{\mathbb{E}_\pi[Q_\pi(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1}) - \eta\ln(\pi(\boldsymbol{a}_{t+1}|\boldsymbol{s}_{t+1}))]\} \tag{17}$$

and the policy is improved by

$$\pi_{\text{new}} = \arg\min_{\pi'\in\Pi}\mathcal{D}_{KL}(\pi'(\cdot|\boldsymbol{s}_t)\|Z^{\pi_{\text{old}}}e^{\frac{1}{\eta}Q_{\pi_{\text{old}}}(\boldsymbol{s}_t,\cdot)}) \tag{18}$$

where Π is a policy set and π_{old} is theast updated policy. $Q_{\pi_{\text{old}}}$ is the Q-value of π_{old}, $\mathcal{D}_{KL}$ is the Kullback–Leiber (KL) divergence [43], and $Z^{\pi_{\text{old}}}$ is the normalization factor. Through mathematical deduction [42], Equation (18) can be written as

$$\pi^* = \arg\min_{\pi\in\Pi}\mathbb{E}_\pi[\eta\ln\pi(\boldsymbol{a}_t|\boldsymbol{s}_t) - Q(\boldsymbol{s}_t, \boldsymbol{a}_t)] \tag{19}$$

4.5. Rewards

In the actual process, the velocity direction of the carrier aircraft can be selected arbitrarily in the *xoy* plane. In this subsection, we assume that the carrier moves along the x-direction to facilitate the description of the content of the reward function.

The essence of the reward function in reinforcementearning is to communicate the goal to the agent and motivate the agent to select a strategy to achieve the goal. The goal of the agent in the action is to maximize the reward value. For UAV trajectory generation, reinforcementearning algorithms are trained using discrete reward functions. Because the

agent is initially trained withittle or no knowledge of the environment, it chooses actions at random and may reach the goal over aong period [44]. However, excessively dense reward values may violate safety constraints or reduce accuracy. To address these problems, we designed a reward function for safe-RL. The reward R_t is defined as

$$R_t = C_{\mathrm{R}}[R_{\mathrm{dis}}, R_{\mathrm{sync}}, R_{\mathrm{done}}, R_{\mathrm{safe}}]^{\top} \tag{20}$$

where

$$\begin{aligned} R_{\mathrm{dis}} &= -\frac{\mathrm{d}\|\boldsymbol{d}_t\|}{\mathrm{d}t} \\ R_{\mathrm{sync}} &= \begin{cases} -\dfrac{\mathrm{d}\|\boldsymbol{v}_{Ut} - \boldsymbol{v}_{Dt}\|}{\mathrm{d}t}, e_t \le e_{\mathrm{threshold}} \\ 0, e_t > e_{\mathrm{threshold}} \end{cases} \end{aligned} \tag{21}$$

are performance indicators related to distance and speed, respectively, $C_{\mathrm{R}} = [C_{\mathrm{dis}}, C_{\mathrm{sync}}, C_{\mathrm{done}}, C_{\mathrm{safe}}]$ is the designed parameter, which is the coefficient of the rewards. $e_t = \|\boldsymbol{d}_t\|$ is the distance between UAV and the drogue at step t, and $e_{\mathrm{threshold}}$ is the threshold that can be adjusted, after which the two agents start to synchronize their speed. R_{done} can be expressed as follows:

$$R_{\mathrm{done}} = \begin{cases} R_d, d_t < e_d \\ 0, d_t \ge e_d \end{cases} \tag{22}$$

where R_d is aarge positive number, but its value cannot be too high; otherwise, it may cause the algorithm to fall into theocal optimum. e_d is the error range that the recovery mechanism can cover, which means that entering this range can be considered a successful recovery. Distance l_U, which is the distance of the carrier aircraft relative to the safe boundary constraints in the symmetry plane, can be derived from Equation (5), which is expressed as

$$l_U = \|e_x, e_z\| - \sum_{i=0}^{2} l_{2h-i} \binom{2}{i} b_U^{2-i} (1 - b_U)^i \tag{23}$$

where b_U satisfies

$$\begin{aligned} \beta_U &= \arctan \frac{e_z}{e_x}, \beta_{2h-2} < \beta_U < \beta_{2h} \\ \sum_{i=0}^{2} \beta_{2h-i} \binom{2}{i} b_U^{2-i} (1 - b_U)^i &= \beta_U, b_U \in [0, 1) \end{aligned} \tag{24}$$

Safety is judged when the relative distance in the y direction isess than that in the safe-tyine. R_{safe} is the penalty incurred when the UAV violates safety constraints. Because the distance from the UAV to the carrier aircraft can be expressed as $l_C = \|([x_U - x_C, z_U - z_C]\|$, we use the difference between l_C and l_U to describe the relative position between UAV and the safe boundary. R_{safe} can be expressed as

$$R_{\mathrm{safe}} = \begin{cases} -\dfrac{c_1}{\exp\left[c_2(l_C - l_U)^2 - 1\right]}, e_y \le e_{\mathrm{safe}} \\ 0, e_y > e_{\mathrm{safe}} \end{cases} \tag{25}$$

where $c_1, c_2 \in \mathbb{R}_+$ and e_{safe} are the safe distances in the y-direction.

4.6. Algorithm Design and Implementation

In PRL, the deep neural network for Q_θ and π_χ is called "critic" and "actor", respectively. The entire algorithm training process is shown in Figure 9, and the training process will be performed offline. At each time step $t + 1$, we collect the state $\boldsymbol{s}_t$, action $\boldsymbol{a}_t$, and reward R_t from theast step and the current state $\boldsymbol{s}_{t+1}$, which will be stored as a tuple $(\boldsymbol{s}_t, \boldsymbol{a}_t, R_t, \boldsymbol{s}_{t+1})$ at the replay buffer $\mathcal{D}$ [45]. At each step of policy evaluation or improvement, we randomly select a batch of historical data $\mathcal{B}$ from the replay buffer $\mathcal{D}$ to train the parameters θ of $Q_\theta(\cdot)$ and parameters χ of π_χ.

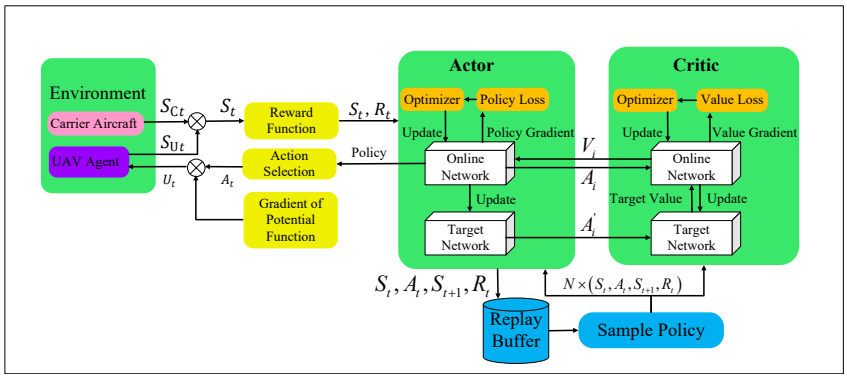


Figure 9. Implementation of the PRL control algorithm.

At the policy evaluation step, the parameters θ are obtained by minimizing the Bellman residual. Furthermore, we introduce a copy of the actor and critic networks to calculate target values, which can slowly change the goal value to improve the stability ofearning.

$$J_Q(\theta) = \mathbb{E}_{(\boldsymbol{s}_t, \boldsymbol{a}_t) \sim \mathcal{D}}[\frac{1}{2}(Q_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) - \hat{Q}(\boldsymbol{s}_t, \boldsymbol{a}_t))^2] \tag{26}$$

with

$$\hat{Q}(\boldsymbol{s}_t, \boldsymbol{a}_t) = R_t + \delta Q_{\overline{\theta}} - \delta\eta \ln(\pi_\chi) \tag{27}$$

in which $\overline{\theta}$ is the parameter of the target critic network.

Parameter χ is updated by adaptive momentum-estimation technique, and Equation (19) can be rewritten as

$$J_\pi(\chi) = \mathbb{E}_{(\boldsymbol{s}_t, \boldsymbol{a}_t) \sim \mathcal{D}}[-\eta \ln \pi_\chi - Q_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t)] \tag{28}$$

Th equation used to update the temperature parameter η is given as

$$J_\eta = \mathbb{E}_\pi[-\eta \ln \pi(\boldsymbol{s}_t|\boldsymbol{a}_t) - \eta\overline{\mathcal{H}}] \tag{29}$$

in which $\overline{\mathcal{H}}$ is target entropy. If training is done, optimal parameters will be output by Algorithm 1, where $\lambda_Q, \lambda_\pi, \lambda_\alpha > 0$ areearning rates and $\tau > 0$ is constant scalar.

Algorithm 1 PRL control algorithm

Initialize parameters θ , η and χ
  for all each iteration **do**
    for all each environment step **do**
      $\boldsymbol{a}_t \sim \pi_\chi(\boldsymbol{a}_t|\boldsymbol{s}_t)$
      $\boldsymbol{s}_{t+1} \sim P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$
      $\mathcal{D} \leftarrow \mathcal{D} \bigcup \{(\boldsymbol{s}_t, \boldsymbol{a}_t, R_t, \boldsymbol{s}_{t+1})\}$
    end for
    for all each gradient step **do**
      Sample a batch of data $\mathcal{B}$ from $\mathcal{D}$
      $\theta \leftarrow \theta - \lambda_Q \hat{\nabla}_\theta J_Q(\theta)$
      $\chi \leftarrow \chi - \lambda_\pi \hat{\nabla}_\chi J_\pi(\chi)$
      $\eta \leftarrow \chi - \lambda_\pi \hat{\nabla}_\chi J_\pi(\chi)$
      $\overline{\theta} \leftarrow \tau\theta + (1-\tau)\overline{\theta}$
    end for
  end for
  until training is done
  Output optimal parameters χ^*, θ^*

5. Performance Analysis

5.1. Converge Analysis

According to (21) and (25), R_{dis}, R_{safe}, and R_{sync} are all non-positive. In addition, the reward function R_{done} is bounded. Therefore, the total return R_t is bounded, that is $R_t \in [R_{\min}, R_{\text{done}}]$, where $R_{\min}$ is theowest bound of reward. For the convergence analysis of the PRL algorithm, the following Theorems 1 and 2 can be given.

Theorem 1. *Let $\mathcal{T}^{\pi}$ be the Bellman backup operator under policy π, sequence $Q^{i+1}(\boldsymbol{s}, \boldsymbol{a}) = \mathcal{T}^{\pi} Q^{i}(\boldsymbol{s}, \boldsymbol{a})$ will coverage to Q-function Q^{π} while $i \to \infty$.*

Proof. The specific proof process is detailed in Appendix A. □

Theorem 2. *According to (18),et $Q_{\pi_{old}}$ be the old policy π, it exists $Q_{\pi_{old}}(\boldsymbol{s}, \boldsymbol{a}) \leq Q_{\pi_{new}}(\boldsymbol{s}, \boldsymbol{a})$,$\forall \boldsymbol{s} \in \mathcal{S}$ and $\forall \boldsymbol{a} \in \mathcal{A}$.*

Proof. The specific proof process is detailed in Appendix B. □

According to Theorems 1 and 2, the following theorem can be proposed, in which superscript k represents the result of the k-th iteration.

Theorem 3. *Suppose π_k is the strategy obtained at the k-th iteration, if the strategy evaluation and improvement steps are applied repeatedly, there exists $\pi_k \to \pi_{optimal}$ as $k \to \infty$such that $Q_{\pi_{optimal}}(\boldsymbol{s}, \boldsymbol{a}) \leq Q_{\pi_{new}}(\boldsymbol{s}, \boldsymbol{a})$, $\forall \boldsymbol{s} \in \mathcal{S}$ and $\forall \boldsymbol{a} \in \mathcal{A}$, where $\pi_{optimal}$ is the optimal policy.*

Proof. The specific proof process is detailed in Appendix C. □

5.2. Stability Analysis

According to the following theorem, there is stability of the PRL algorithm in the closed-loop tracking system.

Theorem 4. *To prove that the closed-loop tracking system is stable, the Q-functionearned in the PRL training process needs to satisfy.*

$$\xi_1 R_t \leq Q_{\pi}(\boldsymbol{s}, \boldsymbol{a}) \leq \xi_2 R_t \tag{30}$$

$$\mathbb{E}_{\boldsymbol{s}_t \sim \zeta_{\pi}}[\mathbb{E}_{\boldsymbol{s}_{t+1} \sim \mathcal{P}_{\pi}}[Q_{\pi}(\boldsymbol{s}+1, \boldsymbol{a}+1)] - Q_{\pi}(\boldsymbol{s}, \boldsymbol{a})] \leq -v\mathbb{E}_{\boldsymbol{s}_t \sim \zeta_{\pi}}[R_t] \tag{31}$$

$$\left\|\mathbb{E}_{\boldsymbol{s}_t \sim \zeta_{\pi}}[\boldsymbol{s}_t]\right\| \leq \epsilon^t \frac{\xi_1}{\xi_2} \left\|\mathbb{E}_{\boldsymbol{s}_0 \sim \zeta_{\pi}}[\boldsymbol{s}_0]\right\| \tag{32}$$

where $\xi_1 > 0$, $\xi_2 > 0$, $v > 0$, and $\epsilon \in (0, 1)$. $\zeta_{\pi}(\boldsymbol{s}_t)$ can be expressed as

$$\zeta_{\pi}(\boldsymbol{s}_t) \triangleq \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} \mathcal{P}(\boldsymbol{s}_t \mid \pi, t) \tag{33}$$

Proof. The specific proof process is detailed in Appendix D. □

6. Simulations and Comparisons

The numerical simulation results demonstrate the effectiveness of the proposed method in solving the rendezvous problem in UAV aerial recovery. First, the convergence and accuracy of the PRL algorithm were verified in a task scenario in which the recovery drogue moved along a straightine. Then, the algorithm was extended to a more complex rendezvous situation, in which the recovery drogue moved along a circle, and the performance of the proposed method was further investigated. The APF method, which enables real-time on-board execution [46,47], was used to solve this rendezvous problem with non-convex constraints for comparison [15,48] to demonstrate the performance of the

PRL. Recycling in our simulation was considered successful if the following conditions were met:

$$\begin{aligned} \|\boldsymbol{d}_{Dt}\| &< 0.3 \\ \|\boldsymbol{e}_v\| &< 0.1 \end{aligned} \tag{34}$$

Note that $l_U - l_C > 0$ should be satisfied at every step during the entire approach; otherwise, it is regarded as a failure.

Reinforcementearning training was conducted in an offline environment. In the actual simulation process, a real-time response can be achieved after the network parameters areoaded. In the modeling of safety constraints, the coordinates of the safety boundary control points N_a established in the symmetric plane relative to the carrier after being converted to the the Cartesian coordinate system are given as $N_0(-10,-10)$, $N_1(-80,-15)$, $N_2(-120,-3)$, $N_3(80,15)$, $N_4(2,15)$, $N_5(30,15)$, $N_6(30,-4)$, $N_7(30,-15)$, $N_8(2,-15)$, $N_9(-8,-13)$, and $D_0(-15,-15)$. Table 1 summarizes the hyperparameters used in reinforcementearning. In combination with Algorithm 1, a training architecture of reinforcementearning was established.

Table 1. Hyperparameters of PRL.

Hyperparameters	Value
Sample batchsize $\|\mathcal{B}\|$	512
Learning rate μ_π	1×10^{-3}
Learning rate μ_V	2×10^{-3}
Discount factorδ	0.99
Neural network structure of policy	(256, 256)
Neural network structure of Value	(256, 256)
Time step (ms)	10
ϵ	0.2
C_R	[50.0, 5.0, 20.0, 20.0]

6.1. Rendezvous on Straight-Line Path

In the first case, the carrier plane is assumed to move along the x-direction at a constant speed of 161 m/s and the UAV approaches from above with an initial position deviation $\boldsymbol{d}_{D0} = [-1000, 1000, -1000]^\top$ and initial speed $\boldsymbol{e}_{v0} = [80, 0, 0]^\top$. Figure 10 shows the global trajectories of the different methods with the same initial state. Figure 11 shows the converged solutions for e_x, e_y, e_z, $\|\boldsymbol{v}_{Ut}\|$, e_γ, e_ψ. The red curves represent the solution obtained by the PRL, the blue curves represent the solution obtained by the APF, and the black dashedine represents the desired state curve.

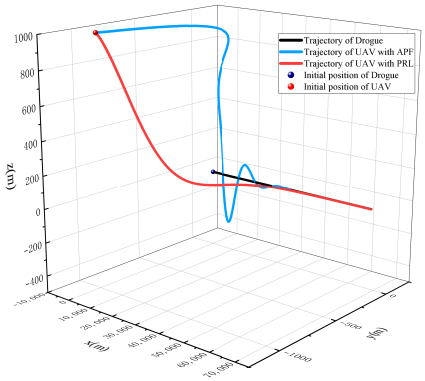
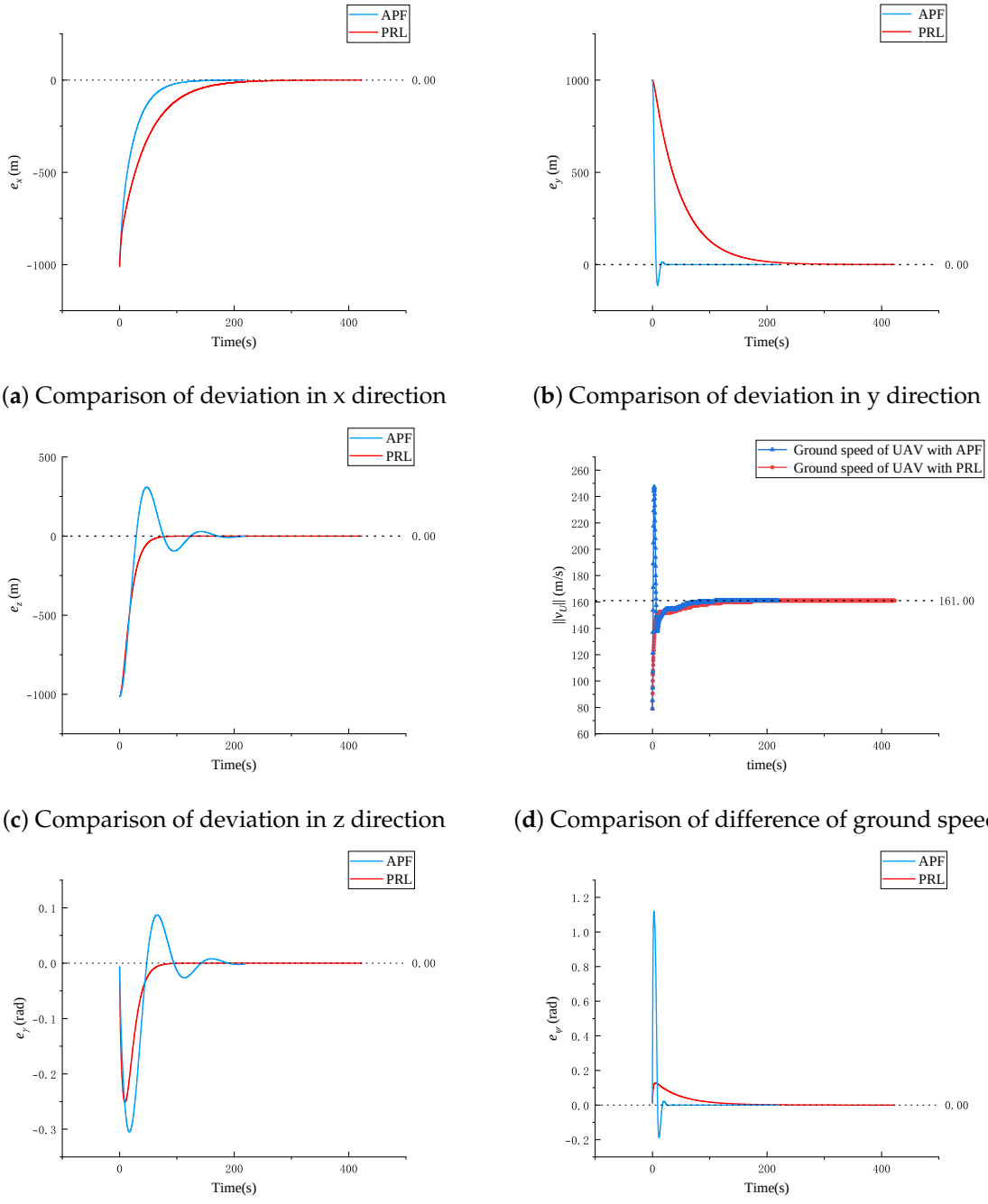


Figure 10. Global trajectory of aerial recovery of Case 1.



(**a**) Comparison of deviation in x direction

(**b**) Comparison of deviation in y direction

(**c**) Comparison of deviation in z direction

(**d**) Comparison of difference of ground speed

(**e**) Comparison of flight path angle error

(**f**) Comparison of yaw angle error

Figure 11. Converged solution results of Case 1.

As shown in Figure 11a, the position deviation in the x-direction always decreases throughout the process, and the APF converges faster in this direction. According to Figure 11b,c, in the y and z directions, the position deviation of the PRL in the direction steadily decreases, whereas the position deviation of the APF oscillatingly decreases. After a period, the relative position converges to a small value, and rendezvous occurs. Figure 11d shows that the overshoot of PRL was significantly smaller than that of APF as the ground speed approached the target value. Figure 11e,f shows that the velocity direction deviation of PRL quickly converged to the target value, whereas APF slowly converged. Combined with the global trajectory shown in Figure 10, both algorithms can generate trajectories that bring the drone closer to the target. Table 2 presents a comparison of the numerical results,

which shows the average of 100 experiments of the two algorithms in random initial state, where the probability of reaching the allowable deviation range under the condition of satisfying the safety constraints is P_{suc}. Based on the table, both methods can bring the UAV close to the target point, and the ending accuracy of Safe-RL is slightly higher.

Table 2. Comparison of average final state of rendezvousing on straight-line path.

Method	PRL	APF
$\|\boldsymbol{d}_{Dt}\|$, m/s	1.47×10^{-1}	1.93×10^{-1}
v_U, m/s	161	161
e_γ, rad	-3.5×10^{-15}	-9.98×10^{-4}
e_ψ, rad	5.7×10^{-40}	-8.07×10^{-4}
P_{suc}, Success rate	1.00	0.43

6.2. Rendezvous on Circular Path

We consider a recovery drogue moving along a circular path in another intersection case, which is more complex than the straight-line intersection problem discussed in the previous subsection, to further investigate the performance of the proposed method. In this case, the speed of recovery of the drogue was 161 m/s, and the path radius was 72,000 m. The UAV approaches from above with an initial position deviation $\boldsymbol{d}_{D0} = [-2500, 2500, -1000]^\top$ and initial speed $\boldsymbol{e}_{v0} = [80, 40, 0]^\top$. Figure 12 shows the global trajectories of the different methods with the same initial state. In these cases, the PRL method provides a smoother trajectory than the APF. Figure 13 shows the converged solutions for e_x, e_y, e_z, $\|\boldsymbol{v}_{Ut}\|$, e_γ, e_ψ. The red curves represent the solution obtained by the PRL, the blue curves are the solution from APF, the black dashedine represents the desired state curve, and the purple dashedine indicates the upper bound of the control.

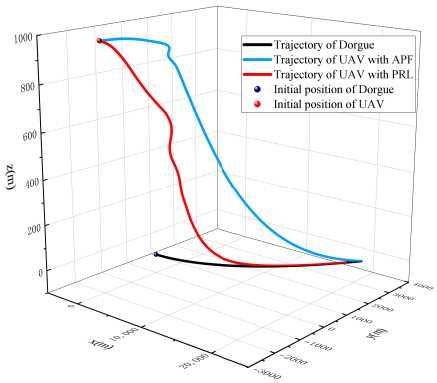
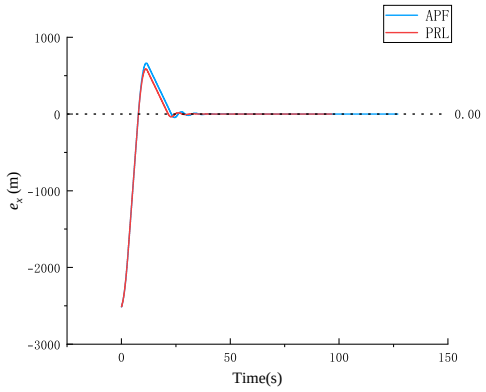


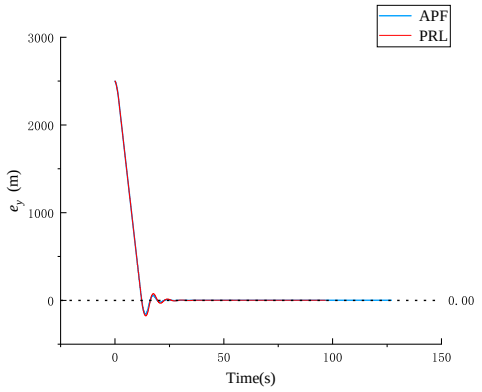
Figure 12. Global trajectory of aerial recovery of Case 2.

Based on the same parameter training used in Section 6.1, the PRL algorithm is implemented and the results are given as follows. Compared with the APF, similar to the solution observed in the previous example, the converged solutions of the two do not completely overlap; however, they follow a similar trend: the speed of the UAV reaches its upper bound at the beginning of the flight, as shown in Figure 11d, and hardly touches theower bound. As shown in Figure 11a–c, during the entire process, the position deviation continued to converge, and the PRL converged faster in the z direction. Figure 11e,f shows that at the end of the mission, the speed of the UAV obtained by the PRL algorithm is aligned with the heading direction of the recovery drogue, whereas the results of the APF algorithm still have a certain deviation, although the deviation isess than 0.01 rad. The 3D intersection trajectory shown in Figure 12 indicates that the recovery drogue completed a circular arc before the intersection occurred. Similarly, the PRL method provides a smoother trajectory than the APF. Table 3 presents the comparison of the numerical results; it shows the average of 100 experiments of the two algorithms in the same random initial state. The recovery success rate of the PRL was

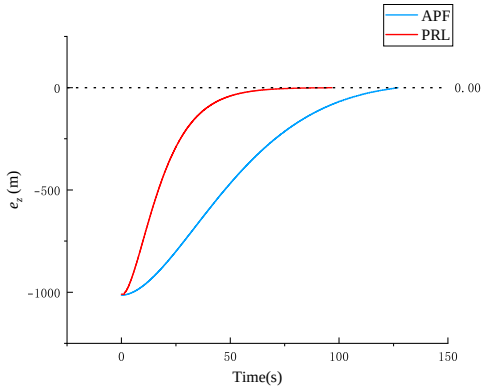
significantly higher than that of the APF. Section 6.3 analyzes the safety of the recovery process.



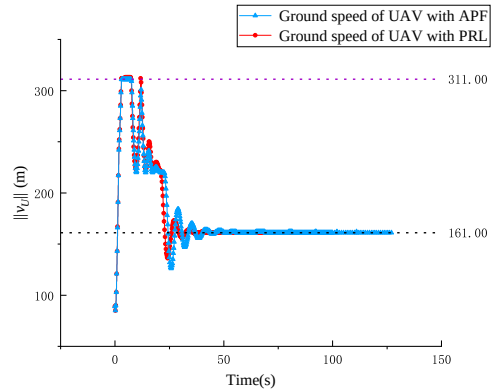
(**a**) Comparison of deviation in x direction



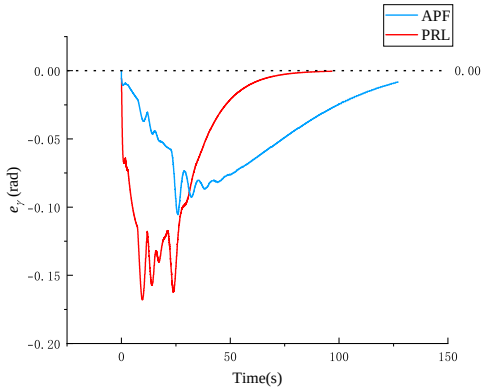
(**b**) Comparison of deviation in y direction



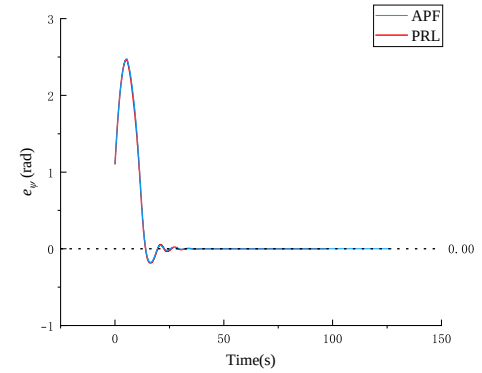
(**c**) Comparison of deviation in z direction



(**d**) Comparison of difference of ground speed



(**e**) Comparison of flight path angle error



(**f**) Comparison of yaw angle error

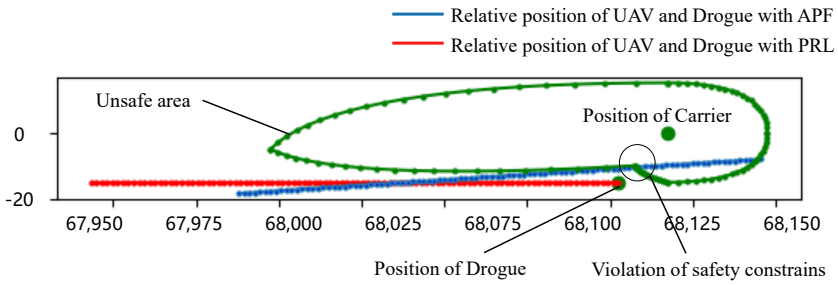
Figure 13. Converged solution results of Case 2.

Table 3. Comparison of average final state of rendezvous on circular path.

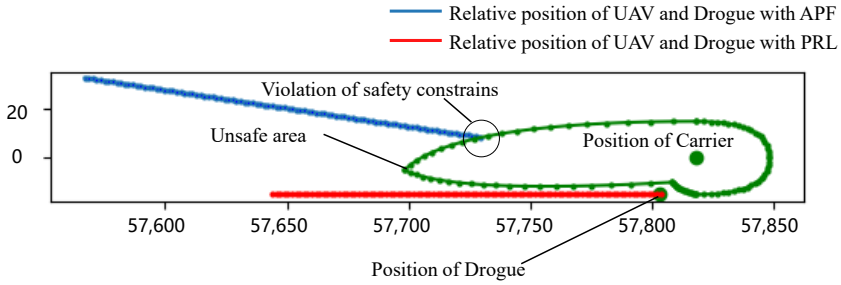
Method	PRL	APF
$\|\boldsymbol{d}_{Dt}\|$, m/s	1.97×10^{-1}	2.22×10^{-1}
v_U, m/s	161	161
e_γ, rad	1.00×10^{-5}	-1.79×10^{-3}
e_ψ, rad	-3.91×10^{-4}	-8.07×10^{-3}
P_{suc}, Success rate	0.99	0.33

6.3. Security Analysis

The comparisons of the success rates of PRL and APF are shown in Tables 2 and 3. The success rate is the rate at which the UAV does not violate safety constraints and satisfies reachability. The results shown in the table prove the ability of PRL to solve the intersection problem with security constraints. Figure 14a shows typical cases of an approaching process with safety constraint violations of APF. Under the same initial state, the APF violates the safe constrains and breaks into the unsafe area, which frequently occurs in our test of 100 randomly initial states. This is because of the overshoot of the APF whose trajectory is not as smooth as PRL, which also causes it to touch the safety boundary at some point. Only when the trajectory is relatively flat can the target position be reached while ensuring safety. However, after training, owing to the penalty of security constraints, PRL has a significantly higher probability of satisfying reachability without violating the security constraints.



(**a**) Typical cases of safety constraint violations for rendezvous from above



(**b**) Typical cases of safety constraint violations for rendezvous from below

Figure 14. Typical cases of the terminal trajectory of PRL and APF.

The results of the comprehensive simulation experiments can be used to easily compare the advantages of the proposed safe-RL algorithm. First, the convergence of PRL hasess overshoot and is more stable than APF. Second, PRL has a smaller terminal deviation. Finally, under the condition of satisfying the safety constraints, the task success rate of PRL is higher. In summary, the online feasible trajectory generation method has significant application potential for aerial recovery tasks.

7. Conclusions

To address a keyimitation in the current use of UAVs, this study considers rendezvous and docking problems based on aerial recovery. The main contributions of this study are as follows. First, a method of establishing safety constraints based on finite element analysis was proposed to model the non-safety region. Second, a reinforcementearning method based on a potential function was proposed to realize UAV online trajectory generation, while the trajectory satisfies the requirements of non-convex safety constraints. Finally, a reward function was designed such that the UAV can overcome the problem of reward sparsity even without expert experience. The simulation experiment results show that the performance of this algorithm is better than APF. The proposed algorithm has a smoother trajectory, higher efficiency, and can get closer to the targetocation. In terms of more important safety, in the two working conditions of the experiment, the success rates of this

algorithm in avoiding hazard areas reached 100% and 99%, respectively, while APF was only 43% and 33%. Therefore, the proposed algorithm can be generalized to the rendezvous and docking problem as a baseline technology, demonstrating that autonomous real-time, collision-free rendezvous and aircraft approaches are feasible.

Author Contributions: Conceptualization, B.Z. and M.H.; methodology, B.Z. and M.H.; software, B.Z. and M.H.; validation, B.Z., M.H. and N.Q.; formal analysis, J.W. and B.Z.; investigation, B.Z., Z.Y., and J.W.; resources, N.Q.; data curation, B.Z. and Z.Y.; writing—original draft preparation, B.Z. and M.H.; writing—review and editing, B.Z., Z.Y., and N.Q.; visualization M.H. and J.W.; supervision, N.Q. and M.H.; project administration, N.Q.; funding acquisition, N.Q. and M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (grant number 12272104) and in part by the National Natural Science Foundation of China (grant number U22B2013).

Data Availability Statement: Data are contained within the article.

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

Appendix A. Proof of Theorem 1

Proof. First, define an entropy-enhanced reward $\hat{R}_t = R_t - \delta\mathbb{E}_{s_{t+1}}\{\mathbb{E}_\pi[\eta \ln(\pi(\boldsymbol{a}_{t+1}|\boldsymbol{s}_{t+1}))]\}$ which is bounded, the Bellman operation can be expressed as

$$\mathcal{T}^\pi Q_\pi(\boldsymbol{s},\boldsymbol{a}) = \hat{R}_t + \delta\mathbb{E}_{\boldsymbol{s}_{t+1}}[Q_\pi(\boldsymbol{a}_{t+1}|\boldsymbol{s}_{t+1})] \tag{A1}$$

in terms of (16), $Q_\pi(\boldsymbol{s},\boldsymbol{a}) = \hat{R}_t + \delta\sum_{t+1}^{\infty}\sum_{a_{t+1}}\pi(\boldsymbol{a}_{t+1}|\boldsymbol{s}_{t+1})\sum_{\boldsymbol{s}_{t+1}}\mathcal{P}_{t+1|t}\hat{R_{t+1}}$, so

$$\|Q_\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)\|_\infty \le \overline{R}/(1-\delta) \tag{A2}$$

where $\|Q_\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)\|_\infty = \max_{s,a}|Q_\pi(\boldsymbol{s}_t,\boldsymbol{a}_t)|$. Thus, Q_π is bounded in ∞-norm. There exists

$$\begin{aligned}\left\|\mathcal{T}^\pi {Q_\pi}^1 - \mathcal{T}^\pi {Q_\pi}^2\right\|_\infty &= \left\|\hat{R}_t + \delta\mathbb{E}_{\boldsymbol{s}_{t+1}}[{Q_\pi}^1(\boldsymbol{a}_{t+1}|\boldsymbol{s}_{t+1})] - \hat{R}_t - \delta\mathbb{E}_{\boldsymbol{s}_{t+1}}[{Q_\pi}^2(\boldsymbol{a}_{t+1}|\boldsymbol{s}_{t+1})]\right\|_\infty \\ &\le \delta\left\|{Q_\pi}^1 - {Q_\pi}^2\right\|_\infty\end{aligned} \tag{A3}$$

where ${Q_\pi}^2$ and ${Q_\pi}^1$ are the Q-value atast and current iteration, respectively. Thus, $\lim_{i\to\infty} Q^{i+1}(\boldsymbol{s}_t,\boldsymbol{a}_t) \to Q_\pi$. □

Appendix B. Proof of Theorem 2

Proof. According to (18),

$$\mathbb{E}_{\pi_{\text{new}}}[\eta\ln(\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)) - Q_{\pi_{\text{old}}}(\boldsymbol{s}_t,\boldsymbol{a}_t)] \ge \mathbb{E}_{\pi_{\text{old}}}[\eta\ln(\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)) - Q_{\pi_{\text{old}}}(\boldsymbol{s}_t,\boldsymbol{a}_t)] \tag{A4}$$

combined with (A3), it can be derived that

$$\begin{aligned}Q_{\pi_{\text{old}}}(\boldsymbol{s}_t,\boldsymbol{a}_t) \le R_t + \delta\mathbb{E}_{\boldsymbol{s}_{t+1}}[\mathbb{E}_{\pi_{\text{new}}}[R_{t+1} + \delta\mathbb{E}_{\boldsymbol{s}_{t+2}}[\mathbb{E}_{\pi_{\text{new}}}[Q_{\pi_{\text{old}}}(\boldsymbol{s}_{t+2},\boldsymbol{a}_{t+2}) - \\ \delta\ln\pi_{\text{new}}(\boldsymbol{a}_{t+2}|\boldsymbol{s}_{t+2})]] - \delta\ln\pi_{\text{new}}(\boldsymbol{a}_{t+1}|\boldsymbol{s}_{t+1})]]\ldots \\ \le Q_{\pi_{\text{new}}}(\boldsymbol{s}_t,\boldsymbol{a}_t)\end{aligned} \tag{A5}$$

□

Appendix C. Proof of Theorem 3

Proof. According to (2), $Q_\pi(\boldsymbol{s}, \boldsymbol{a})$ is monotonically non-decreasing as the number of policy network iteration steps increases, which means it will converge to its upper bound $Q^*_\pi(\boldsymbol{s}, \boldsymbol{a})$ with $Q^*_\pi(\boldsymbol{s}, \boldsymbol{a}) \geq Q_\pi(\boldsymbol{s}, \boldsymbol{a})\ \forall \boldsymbol{s} \in \mathcal{S}$and $\forall \boldsymbol{a} \in \mathcal{A}$. □

Appendix D. Proof of Theorem 4

Proof. The probability distribution of MDP under a strategy function $p_\pi(\boldsymbol{s}_t) = \lim_{t\to\infty} \mathcal{P}(\boldsymbol{s}_t \mid \pi, t)$. (32) can be rewritten as

$$\begin{aligned} \int_{\mathcal{S}} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \mathcal{P}(\boldsymbol{s}_t \mid \pi, t) \times \Big(\int_{\mathcal{S}} \mathcal{P}_\pi(\boldsymbol{s}_{t+1} | \boldsymbol{s}_t) \times Q_{t+1} \mathrm{d}\boldsymbol{s}_{t+1} - Q_t \Big) \mathrm{d}\boldsymbol{s}_t \\ = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} (\mathbb{E}_{\mathcal{P}(\boldsymbol{s}_{t+1}|\pi,t+1)}[Q_{t+1}] - \mathbb{E}_{\mathcal{P}(\boldsymbol{s}_t|\pi,t)}[Q_t]) \end{aligned} \tag{A6}$$

Due to the variable ϵ always exists

$$(\frac{1}{\epsilon} - 1)\xi_2 - \frac{v}{\epsilon} = 0 \tag{A7}$$

Combined with (32), it comes to

$$\frac{1}{\epsilon^{t+1}} \mathbb{E}_{\mathcal{P}}(\boldsymbol{s}_{t+1} \mid \pi, t+1)[Q_{t+1}] - \frac{1}{\epsilon^t} \mathbb{E}_{\mathcal{P}}(\boldsymbol{s}_t \mid \pi, t)[Q_t] \leq \frac{1}{\epsilon^t}((\frac{1}{\epsilon} - 1)\xi_2 - \frac{v}{\epsilon})R_t \tag{A8}$$

Thus,

$$\frac{1}{\epsilon^t} \mathbb{E}_{\mathcal{P}}(\boldsymbol{s}_t \mid \pi, t)[Q_t] - \mathbb{E}_{\mathcal{P}}(\boldsymbol{s}_0 \mid \pi, 0)[Q_0] \leq 0 \tag{A9}$$

From the above conclusions, we can get

$$\left\| \mathbb{E}_{\boldsymbol{s}_t \sim \zeta_\pi}[\boldsymbol{s}_t] \right\| \leq \epsilon^t \frac{\xi_2}{\xi_1} \left\| \mathbb{E}_{\boldsymbol{s}_0 \sim \zeta_\pi}[\boldsymbol{s}_0] \right\| \tag{A10}$$

The results show that the PRL algorithm can guarantee the exponential stability. □

References

1. Husseini, T. Gremlins are coming: Darpa enters phase III of its UAV programme. *Army Technology*, 3 July 2018.
2. Nichols, J.W.; Sun, L.; Beard, R.W.; McLain, T. Aerial rendezvous of small unmanned aircraft using a passive towed cable system. *J. Guid. Control. Dyn.* **2014**, *37*, 1131–1142. [CrossRef]
3. Hochstetler, R.D.; Bosma, J.; Chachad, G.; Blanken, M.L. Lighter-than-air (LTA) "airstation"—Unmanned aircraft system (UAS) carrier concept. In Proceedings of the 16th AIAA Aviation Technology, Integration, and Operations Conference, Washington, DC, USA, 13–17 July 2016; p. 4223.
4. Wang, Y.; Wang, H.; Liu, B.; Liu, Y.; Wu, J.; Lu, Z. A visual navigation framework for the aerial recovery of UAVs. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 5019713. [CrossRef]
5. Darpa NABS Gremlin Drone in Midair for First Time. 2021. Available online: https://www.defensenews.com/unmanned (accessed on 1 July 2023).
6. Gremlins Program Demonstrates Airborne Recovery. 2021. Available online: https://www.darpa.mil/news-events/2021-11-05 (accessed on 1 July 2023).
7. Economon, T. Effects of wake vortices on commercial aircraft. In Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 7–10 January 2008; p. 1428.
8. Wei, Z.; Li, X.; Liu, F. Research on aircraft wake vortex evolution and wake encounter in upper airspace. *Int. J. Aeronaut. Space Sci.* **2022**, *23*, 406–418. [CrossRef]
9. Ruhland, J.; Heckmeier, F.M.; Breitsamter, C. Experimental and numerical analysis of wake vortex evolution behind transport aircraft with oscillating flaps. *Aerosp. Sci. Technol.* **2021**, *119*, 107163. [CrossRef]
10. Visscher, I.D.; Lonfils, T.; Winckelmans, G. Fast-time modeling of ground effects on wake vortex transport and decay. *J. Aircr.* **2013**, *50*, 1514–1525. [CrossRef]
11. Ahmad, N.N. Numerical simulation of the aircraft wake vortex flowfield. In Proceedings of the 5th AIAA Atmospheric and Space Environments Conference, San Diego, CA, USA, 24–27 June 2013; p. 2552.

12. Misaka, T.; Holzäpfel, F.; Gerz, T. Large-eddy simulation of aircraft wake evolution from roll-up until vortex decay. *AIAA J.* **2015**, *53*, 2646–2670. [CrossRef]
13. Liu, Y.; Qi, N.; Yao, W.; Zhao, J.; Xu, S. Cooperative path planning for aerial recovery of a UAV swarm using genetic algorithm and homotopic approach. *Appl. Sci.* **2020**, *10*, 4154. [CrossRef]
14. Luo, D.; Xie, R.; Duan, H. A guidanceaw for UAV autonomous aerial refueling based on the iterative computation method. *Chin. J. Aeronaut.* **2014**, *27*, 875–883. [CrossRef]
15. Zappulla, R.; Park, H.; Virgili-Llop, J.; Romano, M. Real-time autonomous spacecraft proximity maneuvers and docking using an adaptive artificial potential field approach. *IEEE Trans. Control. Syst. Technol.* **2018**, *27*, 2598–2605. [CrossRef]
16. Shao, X.; Xia, Y.; Mei, Z.; Zhang, W. Model-guided reinforcementearning enclosing for UAVS with collision-free and reinforced tracking capability. *Aerosp. Sci. Technol.* **2023**, *142*, 108609. [CrossRef]
17. Kim, S.-H.; Padilla, G.E.G.; Kim, K.-J.; Yu, K.-H. Flight path planning for a solar powered UAV in wind fields using direct collocation. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *56*, 1094–1105. [CrossRef]
18. Bonalli, R.; Hérissé, B.; Trélat, E. Optimal control of endoatmosphericaunch vehicle systems: Geometric and computational issues. *IEEE Trans. Autom. Control.* **2019**, *65*, 2418–2433. [CrossRef]
19. Shi, B.; Zhang, Y.; Mu, L.; Huang, J.; Xin, J.; Yi, Y.; Jiao, S.; Xie, G.; Liu, H. UAV trajectory generation based on integration of RRT and minimum snap algorithms. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 4227–4232.
20. Wang, Z.; Lu, Y. Improved sequential convex programming algorithms for entry trajectory optimization. *J. Spacecr. Rocket.* **2020**, *57*, 1373–1386. [CrossRef]
21. Romano, M.; Friedman, D.A.; Shay, T.J. Laboratory experimentation of autonomous spacecraft approach and docking to a collaborative target. *J. Spacecr. Rocket.* **2017**, *44*, 164–173. [CrossRef]
22. Fields, A.R. Continuous Control Artificial Potential Function Methods and Optimal Control. Master's Thesis, Air Force Institute of Technology, Wright-Patterson, OH, USA, 2014.
23. Lu, P.; Liu, X. Autonomous trajectory planning for rendezvous and proximity operations by conic optimization. *J. Guid. Control. Dyn.* **2013**, *36*, 375–389. [CrossRef]
24. Virgili-Llop, J.; Zagaris, C.; Park, H.; Zappulla, R.; Romano, M. Experimental evaluation of model predictive control and inverse dynamics control for spacecraft proximity and docking maneuvers. *CEAS Space J.* **2018**, *10*, 37–49. [CrossRef]
25. Sun, L.; Huo, W.; Jiao, Z. Adaptive backstepping control of spacecraft rendezvous and proximity operations with input saturation and full-state constraint. *IEEE Trans. Ind. Electron.* **2016**, *64*, 480–492. [CrossRef]
26. Faust, A.; Oslund, K.; Ramirez, O.; Francis, A.; Tapia, L.; Fiser, M.; Davidson, J. PRM-RL: Long-range robotic navigation tasks by combining reinforcement earning and sampling-based planning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 5113–5120.
27. Williams, K.R.; Schlossman, R.; Whitten, D.; Ingram, J.; Musuvathy, S.; Pagan, J.; Williams, K.A.; Green, S.; Patel, A.; Mazumdar, A.; et al. Trajectory planning with deep reinforcementearning in high-level action spaces. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *59*, 2513–2529. [CrossRef]
28. Dhuheir, M.; Baccour, E.; Erbad, A.; Al-Obaidi, S.S.; Hamdi, M. Deep reinforcement earning for trajectory path planning and distributed inference in resource-constrained UAV swarms. *IEEE Internet Things J.* **2022**, *10*, 8185–8201. [CrossRef]
29. Song, Y.; Romero, A.; Müller, M.; Koltun, V.; Scaramuzza, D. Reaching theimit in autonomous racing: Optimal control versus reinforcementearning. *Sci. Robot.* **2023**, *8*, eadg1462 . [CrossRef]
30. Bellemare, M.G.; Candido, S.; Castro, P.S.; Gong, J.; Machado, M.C.; Moitra, S.; Ponda, S.S.; Wang, Z. Autonomous navigation of stratospheric balloons using reinforcementearning. *Nature* **2020**, *588*, 77–82. [CrossRef]
31. Zhang, H.; Zongxia, J.; Shang, Y.; Xiaochao, L.; Pengyuan, Q.; Shuai, W. Ground maneuver for front-wheel drive aircraft via deep reinforcementearning. *Chin. J. Aeronaut.* **2021**, *34*, 166–176. [CrossRef]
32. Wang, C.; Wang, J.; Wang, J.; Zhang, X. Deep-reinforcement-learning-based autonomous uav navigation with sparse rewards. *IEEE Internet Things J.* **2020**, *7*, 6180–6190. [CrossRef]
33. Burda, Y.; Edwards, H.; Pathak, D.; Storkey, A.; Darrell, T.; Efros, A.A. Large-scale study of curiosity-drivenearning. *arXiv* **2018**, arXiv:1808.04355.
34. Pathak, D.; Agrawal, P.; Efros, A.A.; Darrell, T. Curiosity-driven exploration by self-supervised prediction. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–1 August 2017; pp. 2778–2787.
35. Houthooft, R.; Chen, X.; Duan, Y.; Schulman, J.; Turck, F.D.; Abbeel, P. VIME: Variational information maximizing exploration. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 1–9. [CrossRef]
36. Ng, A.Y.; Harada, D.; Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In Proceedings of the ICML, Bled, Slovenia, 27–30 June 1999; Volume 99, pp. 278–287.
37. Yan, C.; Xiang, X.; Wang, C.; Li, F.; Wang, X.; Xu, X.; Shen, L. Pascal: Population-specific curriculum-based madrl for collision-free flocking with arge-scale fixed-wing UAV swarms. *Aerosp. Sci. Technol.* **2023**, *133*, 108091. [CrossRef]
38. Schwarz, C.W.; Hahn, K.-U. Full-flight simulator study for wake vortex hazard area investigation. *Aerosp. Sci. Technol.* **2006**, *10*, 136–143. [CrossRef]
39. Rossow, V.J. Validation of vortex-lattice method foroads on wings in ift-generated wakes. *J. Aircr.* **1995**, *32*, 1254–1262. [CrossRef]

40. Schwarz, C.; Hahn, K.-U. Gefährdung beim einfliegen von wirbelschleppen. In Proceedings of the Deutscher Luft- und Raumfahrtkongress 2003, Jahrbuch 2003, Munich, Germany, 17–20 November 2003.
41. Munoz, J.; Boyarko, G.; Fitz-Coy, N. Rapid path-planning options for autonomous proximity operations of spacecraft. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Toronto, ON, Canada, 2–5 August 2010; p. 7667.
42. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcementearning with a stochastic actor. In Proceedings of the International Conference on Machine Learning, PMLR, Vienna, Austria, 25–31 July 2018; pp. 1861–1870.
43. Zhang, Q.; Pan, W.; Reppa, V. Model-reference reinforcementearning for collision-free tracking control of autonomous surface vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 8770–8781. [CrossRef]
44. Qi, C.; Wu, C.; Lei, L.; Li, X.; Cong, P. UAV path planning based on the improved ppo algorithm. In Proceedings of the 2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering (ARACE), Qingdao, China, 26–28 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 193–199.
45. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcementearning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
46. Munoz, J.D. Rapid Path-Planning Algorithms for Autonomous Proximity Operations of Satellites. Ph.D. Thesis, University of Florida, Gainesville, FL, USA, 2011.
47. Bevilacqua, R.; Lehmann, T.; Romano, M.Development and experimentation of LQR/APF guidance and control for autonomous proximity maneuvers of multiple spacecraft. *Acta Astronaut.* **2011**, *68*, 1260–1275. [CrossRef]
48. Lopez, I.; Mclnnes, C.R. Autonomous rendezvous using artificial potential function guidance. *J. Guid. Control. Dyn.* **1995**, *18*, 237–241. [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.