electronics

MDPI

Article

Reinforcement Learning-Based Energy-Saving Path Planning for UAVs in Turbulent Wind

Shaonan Chen, Yuhong Mo *, Xiaorui Wu, Jing Xiao and Quan Liu

Electric Power Science Research Institute of Guangxi Power Grid Co., Ltd., Nanning 530023, China; chenshaonan749963@126.com (S.C.); xiao_j.sy@gx.csg.cn (X.W.); wu_xr.sy@gx.csg.cn (J.X.); liuquan1140@163.com (Q.L.)

* Correspondence: moyuhong123@126.com

Abstract: The unmanned aerial vehicle (UAV) is prevalent in power inspection. However, due to a limited battery life, turbulent wind, and its motion, it brings some challenges. To address these problems, a reinforcement learning-based energy-saving path-planning algorithm (ESPP-RL) in a turbulent wind environment is proposed. The algorithm dynamically adjusts flight strategies for UAVs based on reinforcement learning to find the most energy-saving flight paths. Thus, the UAV can navigate and overcome real-world constraints in order to save energy. Firstly, an observation processing module is designed to combine battery energy consumption prediction with multi-target path planning. Then, the multi-target path-planning problem is decomposed into iterative, dynamically optimized single-target subproblems, which aim to derive the optimal discrete path solution for energy consumption prediction. Additionally, an adaptive path-planning reward function based on reinforcement learning is designed. Finally, a simulation scenario for a quadcopter UAV is set up in a 3-D turbulent wind environment. Several simulations show that the proposed algorithm can effectively resist the disturbance of turbulent wind and improve convergence.

Keywords: energy-saving path planning; reinforcement learning; unmanned aerial vehicle; turbulent wind

Citation: Chen, S.; Mo, Y.; Wu, X.; Xiao, J.; Liu, Q. Reinforcement Learning-Based Energy-Saving Path Planning for UAVs in Turbulent Wind. *Electronics* **2024**, *13*, 3190. https://doi.org/10.3390/electronics13163190

Academic Editor: Mahmut Reyhanoglu

Received: 10 July 2024
Revised: 8 August 2024
Accepted: 9 August 2024
Published: 12 August 2024

1. Introduction

1.1. Background

In recent years, the unmanned aerial vehicle (UAV) technology has gradually matured, demonstrating diverse potentials and gaining widely attention in engineering research and market applications [1,2]. The endurance time and path planning of UAVs is crucial for information acquisition and task execution of path planning. In practical applications, such as power inspection, aerial photography, and logistics, UAVs need to have sufficient flight range and endurance time to complete the task of path planning [3]. However, the endurance issue has always been a major factor restricting the development of UAV technology research and market applications. Therefore, how to effectively improve the endurance time of UAVs has become an urgent problem to be solved in the industry.

UAVs can be classified into several common types based on their mechanical structures, including multi-rotor drones, fixed-wing drones, and unmanned helicopters. Multi-rotor drones and unmanned helicopters are characterized by a flexible attitude, the capability to ascend and descend vertically, the ability to hover freely, low requirements for the environment, and a site for take-off and landing [4]. However, they have some limitations in terms of load capacity, flight speed, and endurance. Therefore, it becomes crucial for UAV energy-saving path planning to understand how the power consumption of drones changes with different flight states.

The traditional battery energy consumption management system of UAVs cannot be directly applied to drones. Due to different practical application scenarios, for example, the UAV has different motion modes and thus different energy consumption losses

during autonomous landing and horizontal flight. Furthermore, the influence of three-dimensional(3-D) turbulent wind environmental factors during the flight of the UAV makes the traditional battery energy consumption management system for the UAV. At present, researchers have established battery energy consumption management systems for fixed-wing UAVs and single-rotor UAVs or provided heuristic battery energy consumption management solutions for multi-rotor UAVs [5]. The power consumption model of the UAV has been extensively studied by domestic and foreign scholars [6], mainly divided into three categories: ① the data-driven heuristic battery energy consumption management model. In [7], a quadcopter battery energy consumption management model based on collected data-driven approaches is proposed. By collecting information such as flight speed data and heading angle, the algorithm infers the battery energy-loss model under flight conditions, and then designs an energy consumption management model. However, this algorithm fails to consider designing the battery energy consumption management model from the perspective of task optimization, such as path-planning tasks. Similarly, given the case of a UAV, a power consumption model involving its speed and working conditions is presented in [8] based on experimental results. However, the power consumption models presented in [8,9] are solely related to speed, overlooking the impact of other crucial factors on the power consumption of UAVs, such as the number of rotors and payload weight. Therefore, the lack of more accurate closed-form expressions limits their applications. In addition, by analyzing battery performance, some power consumption models for electric UAVs have been derived in [5,10,11], and the aforementioned shortcomings have been identified. Therefore, in order to theoretically investigate the power consumption of UAVs and improve the applicability of the obtained models, theoretical power consumption models based on kinematics and aircraft theory are considered. ② Battery energy consumption management model based on UAV kinematics theory. In [9], the authors propose a general fixed-wing energy-loss model based on speed and acceleration from the perspective of UAV kinematics theory, but this model is not suitable for rotor drones. In [12], a closed-form power consumption model is derived for a single-rotor UAV under constant-speed 1-D forward flight, which is used to study energy-efficient UAV communications. This model is then extended in [13,14] to derive analytical models for a single-rotor UAV performing 2-D forward flight. Furthermore, the model in [12] was validated in [15] by fitting measured data from UAVs collected from extensive experiments. Additionally, a power consumption model for a single-rotor UAV during vertical flight was derived in [14]. All these works provide valuable guidance for establishing UAV power consumption models. However, considering that multi-rotor UAVs are the most popular type of UAV today, the existing works [5,7,8,10] only provide heuristic power consumption models without rigorous mathematical derivations, which limits their application in many research areas. Therefore, it becomes necessary to derive a theoretical power consumption model for multi-rotor UAVs. Furthermore, previous studies [9,13,14] only considered the flight status of UAVs in 1-D or 2-D scenarios, overlooking the general 3-D scenario. This prompts the need to investigate corresponding power consumption characteristics in order to study the power consumption behavior of UAVs in 3-D scenarios. ③ Battery energy consumption management model for multi-rotor drones. The authors in [16] proposed a closed-form mathematical expression model for battery energy loss in 1-D or 2-D real-world flight scenarios, which can be extended to 3-D real-world scenarios. However, this algorithm neglects the research on the closed-loop learning framework between path planning and data collection (factors such as the UAV's motion model and 3-D turbulent wind), making it difficult to effectively guide the UAV in adaptive path planning under limited energy consumption. Evidently, the battery energy consumption management system for UAVs has evolved from 1-D to 3-D real-world scenarios, extending from a data-collection and data-driven energy management model to a generalizable closed-form mathematical expression for calculating battery energy loss, thereby effectively optimizing energy consumption.

With the advancement of artificial intelligence algorithms, the optimization of existing UAV power consumption model strategies has also become more intelligent. For instance,

Hung et al. [17] controlled the UAV's propulsion system using a fuzzy-based equivalent energy consumption minimization strategy and a control strategy based on the optimal operating curve of the engine. Lee et al. [18] and Khayyam et al. [19] have designed rule-based power management systems and neural network control structures, respectively. Bongermino et al. [20,21] proposed a real-time iterative algorithm based on dynamic programming. Although Lei Tao et al. [22] analyzed various management strategies such as state machines, fuzzy logic, dynamic programming, and minimum equivalent fuel consumption, their methods are still limited to rule-based energy management strategies, and the shortcomings of such methods have not been fully addressed. In [23], the authors adopted the multiagent deep deterministic policy gradient (MADDPG) algorithm and prioritized experience replay (PER) to optimize the energy consumption and cooperation strategy of the UAV-assisted multiaccess edge computing (MEC) system.

In summary, researchers have devoted significant effort to studying the management strategies for UAV battery energy consumption and have achieved remarkable results. However, both the complex 3-D turbulent wind environmental factors and the changes in the UAV's own flight status can affect the battery energy consumption management system. Traditional research has rarely combined path-planning targets with known acquisition parameters for autonomous learning.

Reinforcement learning (RL) is a recently emerged artificial intelligence (AI) technique that describes and solves the problem of an agent achieving specific goals through interactions with the environment by learning to maximize rewards. Therefore, this paper proposes a reinforcement learning-based energy-saving path-planning algorithm (ESPP-RL) for the UAV in turbulent wind. Based on RL, the ESPP-RL algorithm considers the 3-D turbulent wind environmental and the UAV's flight movement patterns (including horizontal flight, vertical descent, and vertical ascent), introducing local observation conditions to enable the policy network to efficiently process observation features of dynamic dimensions, thus making path planning adaptive.

1.2. Motivation and Contribution

In recent years, RL has been successfully applied in the fields of robot planning, control, and navigation. RL has been shown to be a powerful tool for solving path-planning problems. Through interaction with the environment, the agent can learn how to make optimal decisions in uncertain environments. However, most existing RL methods still face challenges when dealing with high-dimensional state spaces and dynamic environments. Moreover, ensuring the robustness of the algorithm under various wind intensities while maintaining real-time performance is also a current research hotspot. A reinforcement learning-based energy-saving path-planning algorithm (ESPP-RL) in a turbulent wind environment is proposed. The main contributions of this paper are as follows:

- Considering the complex factors of 3-D turbulent wind and the UAV's motion models (including horizontal flight, vertical descent, and vertical ascent), an optimal energy-saving path planning for the UAV on the basic of the framework of RL technology is proposed.
- Utilizing end-to-end training, combined with carefully crafted state features and reward functions, an efficient single-policy network is introduced, which not only achieves adaptive path planning but also optimizes dynamic battery management strategies to tackle challenges under partially observable conditions. The policy network adeptly handles dynamic and dimensional observational features, significantly enhancing the UAV's adaptability in path planning under limited battery capacity and improving energy-saving efficiency.
- Conducting sets of experiments on the built simulation platform, compared to other algorithms, the ESPP-RL algorithm demonstrates superior performance and notable robustness advantages under a complex 3-D turbulent wind environment.

2. System Model and Problem Formulation

2.1. UAV Model

Firstly, a dynamic analysis of autonomous drones is crucial. In this paper, the dynamics model of the four-rotor UAV is taken as the analysis object, which has six degrees of freedom. As shown in Figure 1, the geodetic coordinate system and AUV coordinate system are set up to describe the motion and force of the UAV. $O_B X_B Y_B Z_B$ is the geodetic coordinate system and $O_E X_E Y_E Z_E$ is the body coordinate system used to describe the motion of the drone for the 3-D environment.

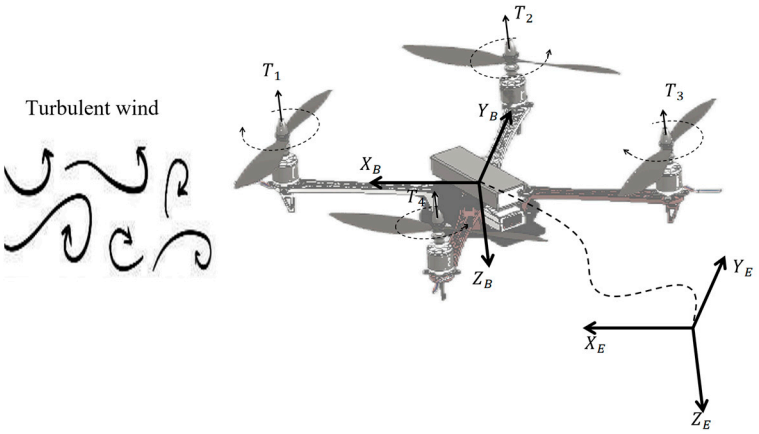


Figure 1. Coordinate systems for a quadrotor UAV.

It is assumed that for small roll and pitch rotation, the angular dynamics are considered to be consistent across frames. For the 3-D environment, the linear position of the UAV in the inertial coordinate system is $\mathbf{d} = [\,x\,,\,y\,,\,z]^{\mathrm{T}}$ and the linear velocity in the body coordinate system is $\mathbf{v} = [\,u\,,\,v\,,\,w]^{\mathrm{T}}$, the attitude vector is $\Theta = [\,\varphi\,,\,\theta\,,\,\psi]^{\mathrm{T}}$, and the angular speed vector is $\mathbf{\Omega}$. The kinetics can be simplified into [24–26]:

$$\dot{\mathbf{d}} = \mathbf{R}\mathbf{v} \tag{1}$$

$$\dot{\Theta} = \mathbf{\Omega} \tag{2}$$

$$\mathbf{f} = -T_h\mathbf{e_3} + \mathbf{R}^{-1}G\mathbf{e_3} + \mathbf{R}^{-1}\delta \tag{3}$$

$$G\dot{\mathbf{v}} = \mathbf{f}g - G\mathbf{\Omega} \times \mathbf{v} \tag{4}$$

$$\dot{\mathbf{\Omega}} = \mathbf{J}^{-1}(\boldsymbol{\tau} - \mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega}) \tag{5}$$

where G is the gravity of the vehicle and g is the acceleration of gravity. The transformation matrix $\mathbf{R}$ is responsible for converting the data from the volume coordinate system to the inertial coordinate system. The vector $\mathbf{f}$ contains all the linear forces acting on the quadrotor, director vector $\mathbf{e_3} = [0, 0, 1]^{\mathrm{T}}$, while $\delta = [\delta x, \delta y, \delta z]^{\mathrm{T}}$ describes all the external disturbances acting on the vehicle. The total thrust T_h is generated by four motors and can be expressed as $T_h = \sum_{j=1}^{4} T_h^j$.

In the hovering state of the four-rotor UAV, it is presumed that the weight of each rotor is uniform. Since the four-rotor UAV flies in a 3-D scene, its own gravity and the total thrust are balanced, so the thrust on each rotor is equal. In order to simplify the calculation, the rotor-related parameters (rotor solidity h, rotor disk area A_s) on each rotor are the same, and the weight of each rotor of a quadrotor UAV is the same, where C_T is the thrust coefficient and ρ is the air density. The main notations used in this section are summarized in Table 1.

Table 1. List of main notations.

Parameter	Meaning	Value
ρ	Air density in kg/m^3	1.168
h	Rotor solidity	0.045
A_s	Rotor disk area in m^2	0.214
C_T	Thrust coefficient	0.001195
k	Incremental correction faction to induced power	0.11
γ_v	Fuselage drag coefficient during vertical flight in kg/m	0.220168
γ_h	Fuselage drag coefficient in horizontal flight in kg/m	0.005256
λ	Profile drag coefficient	0.011
G	UAV weight in Newtons	20

The quadrotor UAV flies horizontally with a constant speed $\mathbf{V}_u$ in the 3-D environment, which is decomposed into three speeds on the world coordinate axis, $\mathbf{V}_x$, $\mathbf{V}_y$, and $\mathbf{V}_z$. When a quadrotor UAV is hovering, the energy consumption on each rotor is equal, so the total energy consumption during hovering can be expressed as the sum of the energy consumption of the four rotors. The energy consumption of the jth rotor P_{hov}^j and the total energy consumption during hovering can be formulated as [16]:

$$P_{hov}^j = \left(\frac{G}{4C_T}\right)^{3/2} \frac{\lambda h}{8\sqrt{\rho \mathrm{A}_s}} + (1+k)\left(\frac{G}{4}\right)^{3/2} \frac{1}{\sqrt{2\rho \mathrm{A}_s}} \tag{6}$$

$$\begin{aligned} \sum_j^4 P_{hov}^j &= 4 \times \left(\left(\frac{G}{4C_T}\right)^{3/2} \frac{\lambda h}{8\sqrt{\rho \mathrm{A}_s}} + (1+k)\left(\frac{G}{4}\right)^{3/2} \frac{1}{\sqrt{2\rho \mathrm{A}_s}} \right) \\ &= \underbrace{\left(\frac{G}{C_T}\right)^{3/2} \frac{\lambda h}{16\sqrt{\rho \mathrm{A}_s}}}_{P_{bl}} + \underbrace{(1+k) G^{3/2} \frac{1}{2\sqrt{2\rho \mathrm{A}_s}}}_{P_{in}} \end{aligned} \tag{7}$$

where k represents the incremental correction factor to induced power. The total energy consumption during hovering can be composed of two parts: one is the rotor profile power P_{bl} and the other is the induced power P_{in}.

When a quadrotor UAV flies horizontally with a constant speed, its energy consumption can be expressed as [16]:

$$\mathbf{\Delta} P_h(\mathrm{V}_h) = \frac{3}{4}\delta \sqrt{\frac{G\rho \mathrm{A}_s}{C_T}} h \|\mathrm{V}_h\|^2 + P_{in}\left[\left(\sqrt{1 + \frac{\|\mathrm{V}_h\|^4}{4v_0^4}} - \frac{\|\mathrm{V}_h\|^2}{2v_0^2}\right)^{1/2} - 1\right] + 4\rho\gamma_h \|\mathrm{V}_h\|^3 \tag{8}$$

When a quadcopter UAV is flying vertically, there are two situations: vertical ascent and vertical descent. In both cases, the problem of velocity direction and thrust direction is dealt with by the *sgn* function. Therefore, in vertical flight, due to the influence of fuselage resistance, the energy loss is [16]:

$$\begin{aligned} \mathbf{\Delta} P_v(\mathrm{V}_v) &= \frac{1}{2} G \|\mathrm{V}_v\| + 2sgn(\mathrm{V}_v)\rho\gamma_v \|\mathrm{V}_h\|^3 + \left(\frac{G}{2} + 2sgn(\mathrm{V}_v)\rho\gamma_v \|\mathrm{V}_v\|^2\right) \\ &\times \sqrt{\left(1 + \frac{2\gamma_v sgn(\mathrm{V}_v)}{\rho \mathrm{A}_s}\right)\|\mathrm{V}_v\|^2 + \frac{G}{2\rho \mathrm{A}_s}} + sgn(\|\mathrm{V}_v\| - 1)\frac{G}{2}\sqrt{\frac{G}{2\rho \mathrm{A}_s}} \end{aligned} \tag{9}$$

2.2. Turbulent Wind Model

In changeable and challenging 3-D space, the flight trajectory and attitude of autonomous drones are not only limited by their internal dynamics model but also by various factors in the surrounding environment. In particular, the wind field plays a crucial role in the course control and stability of the UAV. The following content will explore the theoretical basis of constructing a turbulent wind model in depth. In a turbulent wind scenario, the

motion of a quadrotor UAV is affected by wind resistance. The turbulent wind disturbance model is noted as follows [27]:

$$\delta_{tur} = -\frac{1}{2}\rho \boldsymbol{B}_{tur} \mathrm{A}_s (\dot{\mathbf{d}} - \boldsymbol{v}_{\omega - tur})^2 sign(\dot{\mathbf{d}} - \boldsymbol{v}_{\omega - tur}) \tag{10}$$

where $\boldsymbol{B}_{tur} = [x_{tur}, y_{tur}, z_{tur}]^{\mathrm{T}}$ represents the drag coefficient and $\boldsymbol{v}_{\omega - tur}$ is the turbulent wind speed in the volume coordinate system.

2.3. Reinforcement Learning Model

Reinforcement learning algorithms acquire rewards in the environment through continuous trial and error, guiding the training and optimization direction of the algorithms based on the reward values, ultimately enabling the RL algorithms to prioritize actions or paths that maximize the reward values. RL is a type of learning approach that acquires corresponding reward values by observing the environment and taking actions and ensures the convergence of the algorithm through maximizing reward expectations.

As shown in Figure 2, RL is generally described by a Markov process which consists of components such as state s, action a, policy π, and reward r, and achieves state transitions in a probabilistic manner based on the state transition function. At each time step t, the agent acquires the current state information s_t from the environment; based on the state s_t and policy π, the agent selects an action a_{t+1} to output within the action space. According to the state transition function $p(s_{t+1}|s_t, a)$, the environment then returns a reward value r and the state for the next time step s_{t+1}. Ultimately, the policy π aims to optimize itself by maximizing the expected reward.

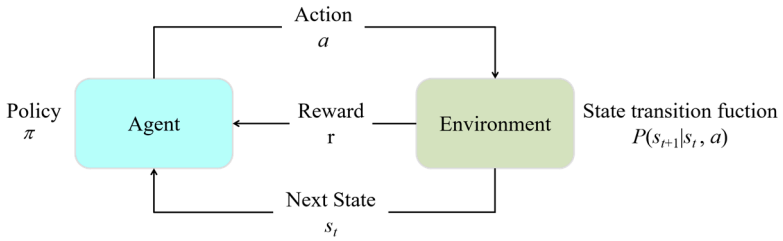


Figure 2. An agent interacting with its environment.

A Markov decision process (MDP) is the foundation of RL algorithms, which expresses the state transition probabilities and processes after making a decision [28]. The MDP is given by the tuple (S, A, p, r), including the state space S, action space A, state transition probability p, and reward value r. Among them, the state transition probability, denoted as $p : S \times S \times A$, represents the probability density of transitioning to the next state when an action is taken $a_t \in A$ in the current state $s_t \in S$. For every completed transfer of state, the environment generates an immediate reward value r based on the state and action, and the range of reward values can be expressed as $r : S \times A \to [r_{\min}, r_{\max}]$.

3. Algorithm

Based on a UAV motion model and 3-D turbulent wind model, this paper presents the development of a UAV's motion simulation platform and introduces a UAV path-planning scheme ESPP-RL. The specific technical content is as follows.

3.1. Turbulent Wind Environment

To increase the realism of the simulation environment and enhance the practical deployability of the algorithm on a UAV, this paper constructs a turbulent wind environment based on PyBullet. As a physics simulation library, PyBullet integrates rich physical and mathematical constraints, enabling complex real-time physical calculations and rendering. It has been widely applied in the fields of robot motion simulation and deep learning.

Therefore, PyBullet serves as the foundational tool for environment construction, catering to the dynamic environmental computation needs of UAVs during complex task execution.

A continuous 3-D environment space of dimensions $100 \times 100 \times 100$ is constructed within PyBullet, and subsequently, a 3-D turbulent wind environment is simulated within it. The model data of turbulent wind are derived utilizing formula (10), and subsequent interpolation processing is applied to this data, ensuring that a corresponding turbulent wind strength $I(u_I, v_I, w_I)$ is assigned to any position within the spatial domain. And u_I, v_I, w_I represent the linear velocities of turbulent wind in the *x*-, *y*-, and *z*-axes, respectively.

3.2. State Transition Function

For a quadcopter UAV, this paper sets the action space to four dimensions $a \in (\Omega_u, \Omega_v, \Omega_w, T_a)$, including the angular velocity outputs $\omega_u, \omega_v, \omega_w$ and thrust T_a in the *x*-, *y*-, and *z*-axes in the body coordinate system. The thrust component on each axis is expressed as:

$$[T_u, T_v, T_w]^{\mathrm{T}} = [\Omega_u, \Omega_v, \Omega_w]^{\mathrm{T}} * T_a \tag{11}$$

$$m[a_u, a_v, a_w]^{\mathrm{T}} = [T_u, T_v, T_w]^{\mathrm{T}} - [G, G, G]^{\mathrm{T}} \tag{12}$$

where $[a_u, a_v, a_w]$ represents the acceleration of the UAV affected by thrust $[T_u, T_v, T_w]$ in the horizontal, vertical, and vertical axis directions. In the context of environmental observation, the state space of the environment is assumed to be partially observable, and the state space acquired by the UAV at a given time step is defined as:

$$S = [\mathbf{d}, \mathbf{v}, \Theta, \Omega, p_{tar}, I] \tag{13}$$

which includes the linear position component $\mathbf{d}$, linear velocity component $\mathbf{v}$, attitude angle component Θ, angular velocity component Ω, and the target's position information of the UAV p_{tar}. It also includes the current turbulence wind strength $I(u_I, v_I, w_I)$ at the location. The state transition is influenced by the UAV state transition equation, the algorithm's action output, and the turbulence wind interference. Upon the execution of an action, the subsequent position $\mathbf{d}' = [\, x' \,, y' \,, z']^{\mathrm{T}}$ of the UAV is determined by a confluence of factors: the UAV's current velocity $[\mathrm{v}_u, \mathrm{v}_v, \mathrm{v}_w]^{\mathrm{T}}$, the acceleration generated by the algorithmic output, and the turbulence wind strength.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \mathrm{v}_u + a_u \\ \mathrm{v}_v + a_v \\ \mathrm{v}_w + a_w \end{bmatrix} + \begin{bmatrix} u_I \\ v_I \\ w_I \end{bmatrix} \tag{14}$$

3.3. Reward Function

In reinforcement learning, the purpose of algorithm optimization is to maximize the reward expectation and then affect the gradient updating direction of the neural network. Therefore, the design of a reward function is very important to the RL algorithm, which directly determines the effect of environment-guided algorithm training. To this end, this paper carefully designed the reward function, including the following four items: distance reward rew_{dis}, collision reward $rew_{collision}$, energy consumption reward rew_{energy}, and goal reward rew_{target}.

In order to improve the efficiency of the algorithm to explore the environment, distance reward is designed to guide the algorithm to approach the target point region. The distance reward rew_{dis} can be expressed by the formula:

$$rew_{dis} = \min DIS(P_{UAV}, P_{tar_i})/size_{env}, i = 1, 2, \ldots n \tag{15}$$

where $DIS(P_{UAV}, P_{tar_i})$ represents the Euclidean distance of the UAV from the target and $size_{env}$ corresponds to the environmental dimensions. The collision reward is a reward for the auxiliary algorithm to perceive the environmental boundary and obstacles, and the value of the obstacle reward changes according to the perception of the agent, where

$r_{perception}$ is the obstacle perception radius of the UAV and r_{fail} is the collision radius. The collision reward $rew_{collision}$ can be expressed by the formula:

$$rew_{collision} = \begin{cases} 0(DIS(P_{UAV}, P_{\text{obstacle}}) > r_{perception} \\ -1/DIS(P_{UAV}, P_{\text{obstacle}}), r_{fail} < DIS(P_{UAV}, P_{\text{obstacle}}) < r_{perception} \\ -100, DIS(P_{UAV}, P_{\text{obstacle}}) < r_{fail} \end{cases} \quad (16)$$

In order to make the UAV complete the task more efficiently, this paper designs the energy consumption reward rew_{energy}, which is used to evaluate the number of running steps and action quality of the algorithm. $\mathbf{\Delta}_h(P_{UAV}, P_{tar})$ is the vertical update component of the UAV and the target position after the execution of each step and $\mathbf{\Delta}_v(P_{UAV}, P_{tar})$ corresponds to the horizontal update component $\psi = 1.2$. The energy consumption reward $rew_{collision}$ can be expressed by the formula:

$$rew_{energy} = -0.1 + \psi\mathbf{\Delta}_h(P_{UAV}, P_{tar}) + \mathbf{\Delta}_v(P_{UAV}, P_{tar_i}) \quad (17)$$

When the UAV reaches the target point, a larger positive reward will be given, for example, $rew_{target} = 100$, and the episode will end. Therefore, the comprehensive reward function is expressed as:

$$rew_{total} = rew_{dis} + rew_{collision} + rew_{energy} + rew_{target} \quad (18)$$

3.4. Reinforcement Learning Algorithm

A significant feature of UAV systems is energy limitation, which requires the path planning of the UAV to have a higher flexibility of algorithms. How to self-adapt the battery management, select the movement path, and adapt to the change-of-wind field can effectively improve the intelligence and adaptability of the algorithm. The traditional path-planning algorithm has limited capability and difficulty in dealing with a UAV multi-target path planning problem in a dynamic environment. Therefore, a reinforcement learning-based energy-saving path-planning algorithm in a turbulent wind environment is designed, The framework of the proposed ESPP-RL algorithm is shown in Figure 3.

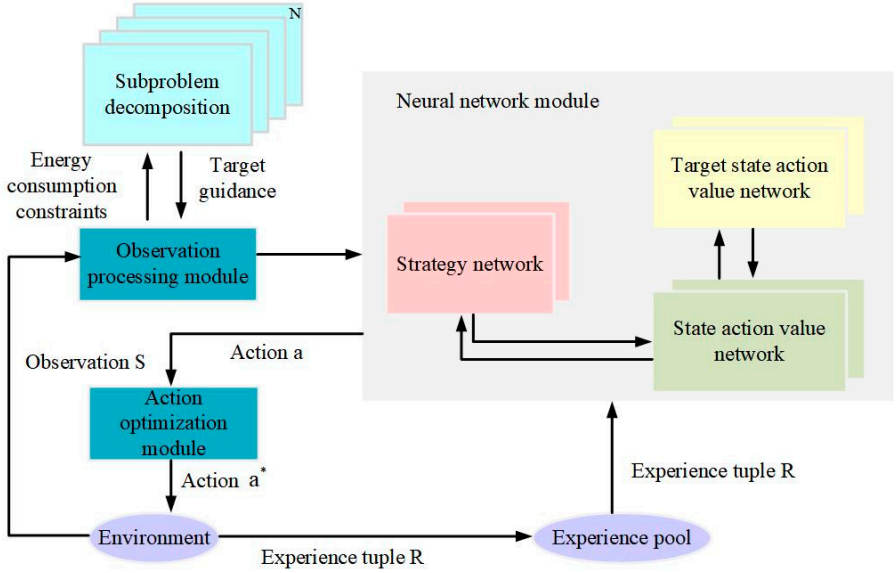


Figure 3. The framework of the ESPP-RL algorithm.

In general, the ESPP-RL algorithm includes an observation processing module, a neural network module, and an action optimizer. This section will explain each module of the ESPP-RL algorithm. Algorithm 1 summarizes the training process.

Firstly, the power consumption of the UAV in 3-D space is expressed as $P_{total}(V_{total}) = P_{hov} + \Delta P_h(V_h) + \Delta P_v(V_v)$, where the total power is decomposed into three components: hover power P_{hov}, vertical descent and ascent power P_h, and horizontal flight power P_v. The energy consumed by the UAV moving to the target point is:

$$E = P_{hov}(\Delta t_h + \Delta t_v) + \Delta P_h(V_h)\Delta t_h + \Delta P_v(V_v)\Delta t_v \tag{19}$$

Under multi-target path planning, the position of the UAV changes dynamically and the corresponding expectation of total energy consumption can be expressed as:

$$E_{total} = \min \sum_i P_{hov}^i(\Delta t_h^i + \Delta t_v^i) + \Delta P_h^i(V_h^i)\Delta t_h^i + \Delta P_v^i(V_v^i)\Delta t_v^i \tag{20}$$

$$E_{total} = \min \sum_i \left\{ \begin{array}{l} P_{hov}^i(\Delta t_h^i + \Delta t_v^i) + \left\{ \frac{3}{4}\delta \sqrt{\frac{G\rho A_s}{C_T}} h \left\| V_h^i \right\|^2 + P_{in} \left[\left(\sqrt{1 + \frac{\left\| V_h^i \right\|^4}{4v_0^4}} - \frac{\left\| V_h^i \right\|^2}{2v_0^2} \right)^{1/2} - 1 \right] + 4\gamma_{r\perp} \left\| V_h^i \right\|^3 \right\} \Delta t_h^i \\ + \left\{ \begin{array}{l} \frac{1}{2} G \left\| V_v^i \right\| + 2\gamma_\perp sgn\left(V_v^i\right) \left\| V_v^i \right\|^3 + \left(\frac{G}{2} + 2\gamma_\perp sgn\left(V_v^i\right) \left\| V_v^i \right\|^2 \right) \\ \times \sqrt{\left(1 + \frac{2\gamma_\perp sgn\left(V_v^i\right)}{\rho A_s} \right) \left\| V_v^i \right\|^2 + \frac{G}{2\rho A_s}} + sgn\left(\left\| V_v^i \right\| - 1 \right) \frac{G}{2} \sqrt{\frac{G}{2\rho A_s}} \end{array} \right\} \Delta t_v^i \end{array} \right\} \tag{21}$$

According to the energy consumption expectation E_{total} and the position state transfer of the UAV, the observation processing module decomposed the multi-target path-planning problem into a dynamic single-objective path-planning subproblem, calculated and stored the solution of the subproblem, and iteratively optimized to obtain the discrete path optimal solution of the energy consumption expectation E_{total}. At the same time, the observation processing module splices the optimal solution of the discrete path with the environment observation and inputs it into the neural network module.

Algorithm 1: ESPP-RL Algorithm

Initialize the multi-target path-planning environment for the UAV
Initialize the state action value network Q_{θ_1}, Q_{θ_2}, target action state value network $Q_{\theta_1'} Q_{\theta_2'}$, policy network π_ϕ, target policy network $\pi_{\phi'}$, and random parameters θ_1, θ_2, ϕ
Initialize the replay buffer **B**
Initial the network parameters $\theta_1' \leftarrow \theta_1, \theta_2' \leftarrow \theta_2, \phi' \leftarrow \phi$
Obtain information about the observation processing module
for t = 1 to T **do**
    Select action with exploration noise $a \sim \pi_{\phi'}(s) + \varepsilon$, $\varepsilon \sim clip(N(0, \sigma), -c, c)$ and observe reward r andnew state s'
    Store transition tuple (s, a, r, s') in B
    Sample mini-batch of N transitions (s, a, r, s') from B
    $\widetilde{a} \leftarrow \pi_{\phi'}(s') + \varepsilon, \varepsilon \sim clip(N(0, \widetilde{\sigma}), -c, c)$
    $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta_i'}(s', \widetilde{a})$
    Update critics $\theta_i \leftarrow argmin_{\theta_i} N^{-1} \sum \left(y - Q_{\theta_i}(s, a)\right)^2$
    if t mod d == True **then**
        Update ϕ by the deterministic policy gradient:
        $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)\Big|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$
        Update target networks
        $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$
        $\theta_i' \leftarrow \tau\theta_i + (1 - \tau)\theta_i'$
    end if
end for
Input the action to the PID controller

The information of the observation processing module is inputted into the neural network module, which includes the state action value network $Q_{\theta_i}(s_t, a_t), i = 1, 2$, the target action state value network $Q_{\theta_i'}(s_t, a_t), i = 1, 2$, the policy network π_ϕ, and the target policy network $\pi_{\phi'}$, with the corresponding network parameters being $\theta_1, \theta_2, \theta_1', \theta_2', \phi, \phi'$, respectively. The updated observation space input target strategy network $\pi_{\phi'}$ with maximum reward expectation, the target strategy network, is expressed as:

$$\pi_{\phi'} = \underset{\pi_{\phi'}}{\operatorname{argmax}} \sum_t \mathrm{E}_{(s_t, a_t) \sim \rho_{\pi_{\phi'}}} [r(s_t, a_t)] \tag{22}$$

The network $\pi_{\phi'}$ selects the action output based on the observation $a \sim \pi_{\phi'}(s) + \varepsilon$, where the action noise is represented by ε, in order to increase the environmental exploration rate $\varepsilon \sim clip(N(0, \sigma), -c, c)$. The algorithm interacts with the environment to feedback the corresponding reward value obtained by the action $r(s_t, a_t)$. The state action value network $Q_{\theta_i}(s_t, a_t), i = 1, 2$ in the neural network module evaluates the action output effect of the current policy network according to the reward value $r(s_t, a_t)$. It is defined as:

$$Q_{\theta_i}(s_t, a_t) \triangleq r(s_t, a_t) + \gamma \mathrm{E}_{s_{t+1} \sim p}[V(s_{t+1})], i = 1, 2 \tag{23}$$

$$V(s_t) \triangleq \mathrm{E}_{a_t \sim \pi_{\phi'}}[Q_{\theta_i}(s_t, a_t)] \tag{24}$$

where γ represents the discount factor, which is used to represent the impact of future n steps on the value function of the current action a_{t+n}, And $V(s_t)$ is the state value function, which evaluates the expected value of the policy network action output in the current state. Further, this method introduces the target state action value network $Q_{\theta_i'}(s_t, a_t)$, $i = 1, 2$, designs the objective function to stabilize the algorithm training, and reduces the overestimation problem caused by the influence of the state value action network on the policy network. The specific objective function y_{tar} is expressed as:

$$y_{tar} = r(s_t, a_t) + \gamma \left(\min_{i=1,2} Q_{\theta_i'}(s_{t+1}, \pi_\phi(s_{t+1})) \right) \tag{25}$$

The state action value network parameters $\theta_i, i = 1, 2$ are updated according to the objective function y_{tar}:

$$\theta_i \leftarrow \underset{\theta_i}{\operatorname{argmin}} N^{-1} \sum (y_{tar} - Q_{\theta_i}(s_t, a_t))^2 \tag{26}$$

At the same time, the strategy parameters and the target action value network parameters in the algorithm are updated in a delayed way to smooth the training process. Policy parameters ϕ are updated by gradient optimization:

$$\phi \leftarrow \nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a) \Big|_{a = \pi_\phi(s)} \nabla_\phi \pi_\phi(s) \tag{27}$$

Finally, the target action value network parameters $\theta_i, i = 1, 2$ and target policy parameter ϕ' are updated, and η represents the soft update weight.

$$\theta_i' \leftarrow \eta \theta_i + (1 - \tau) \theta_i' \tag{28}$$

$$\phi' \leftarrow \eta \phi + (1 - \tau) \phi' \tag{29}$$

In addition, the action output of the neural network module will pass through an action optimizer. The action optimizer is composed of a PID controller, and the action output of the algorithm will be converted into the actual output parameters after the PID controller, in order to smooth the action output and ensure the relative stability of the UAV attitude.

4. Experiment

4.1. Experimental Setup

In this section, we focus on optimizing the path-planning performance of a four-axis UAV in a 3-D turbulent wind environment and optimizing the energy consumption performance while ensuring that the path planning is completed. The turbulent wind model is imported into a 3-D continuous space of dimensions $100 \times 100 \times 100$ utilizing the PyBullet physics simulation library. By integrating the motion space model, state transition model, and reward function specific to the four-axis UAV, a comprehensive RL simulation environment is constructed. The algorithms are implemented on Ubuntu 18.04, with the hardware configuration including an Intel i9-11900k processor, NVIDIA GTX3080 graphics card, and software based on Python 3.8, Ray 1.8.0, and TensorFlow 2.3.1. And soft actor-critic (SAC) and twin delayed deep deterministic (TD3) algorithms are implemented within the framework as comparison algorithms. The algorithm parameters are shown in Table 2.

Table 2. Algorithm parameters.

Algorithm	SAC	TD3	ESPP-RL
Learning rate	3×10^{-4}	1×10^{-3}	1×10^{-3}
Total training steps (T)	1×10^{-6}	1×10^{-6}	1×10^{-6}
Batch_size	1×10^{-5}	2×10^{-6}	1×10^{-5}
Update weight (η)	0.005	0.005	0.005
Policy noise	1	1	1
Discount factor (γ)	0.99	0.99	0.99
Delayed update (d)	None	2	2
PID parameters k_p, k_i, k_d	None	None	$4 \times 10^{-2}, 5 \times 10^{-7}, 1 \times 10^{-4}$

The performance of various algorithms in path planning is quantified through the application of four key metrics, which serve as the basis for evaluating the results

1. Steps: The UAV selects actions at fixed intervals according to the policy and then moves to the next position according to the state transition function. The number of steps reflects the number of interactions and the time consumed.
2. Path length: This paper records the cumulative position offset of the agent in the environment and as the path length. The path length reflects the algorithm's path-planning ability and the ability to resist wind-field interference.
3. Target reach number: The target reach number can accurately reflect the adaptability of the agent to the environment and the quality of the decision. A high goal reach number means that the agent can effectively identify and utilize information in the environment to reach the goal quickly and with minimal cost. Secondly, this index is also an important basis for evaluating the stability of the algorithm.
4. Reward value: The reward value is a core indicator to measure the performance of the agent. The reward value reflects the agent's performance in the environment, i.e., the total reward it receives after performing a series of actions.

4.2. Analysis of Experimental Results

Sufficient training and experiments have been conducted on the SAC, TD3, and ESPP-RL algorithm. Since the training is an oscillating process, in order to better visualize the training results, appropriate smoothing processing is carried out. The training process is recorded in Figure 4, where Figure 4a represents the average number of actions of the algorithm in the training process, Figure 4b corresponds to the reward value of the UAV in a round, Figure 4c reflects the average distance of the UAV in each round, and, finally, Figure 4d represents the average number of target points that the algorithm can reach in the environment.

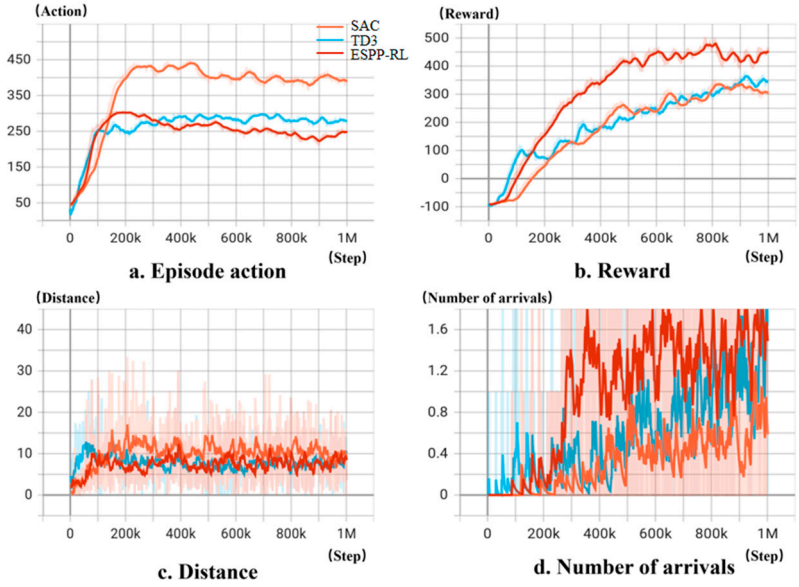


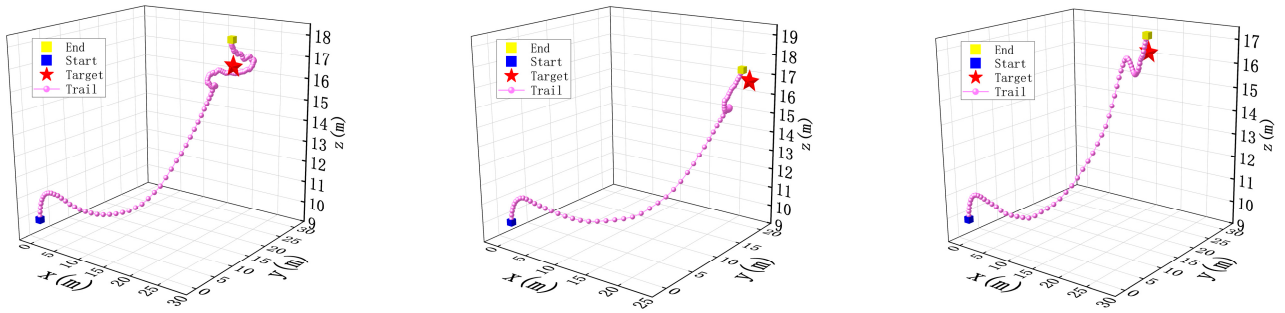
Figure 4. The training process using algorithms.

In the initial stage of algorithm training, the agent randomly selects actions to explore the environment, and when it touches the environmental boundary or obstacles, it will result in the end of the turn and obtain a large negative reward, which makes the average number of steps and reward value of the algorithm at the beginning of the training stage low. With the influence of the distance reward and random exploration, the UAV accidentally explores the target point and obtains a large target reward, and these exploration processes of the environment are combined into experiences and stored in the experience playback pool. The algorithm regularly extracts experience from the experience playback pool for training and updates its own network parameters. It can be seen that the ESPP-RL algorithm benefits from the observation processing module, with a faster training speed and higher cumulative reward value.

4.2.1. Single-Target Path-Planning Environment

To facilitate observation of the algorithm's action strategy output performance, the environment is refined into a single-target path-planning scenario. The algorithm does not need to assign tasks to multiple objective points and mainly considers the path-planning effect when the target is determined. Specifically, a static target point and multiple obstacles are configured within the environment, requiring the UAV to navigate from a designated starting point, bypass the obstacles, and ultimately reach the target location. The experimental results show that all algorithms can avoid obstacles to reach the target point in the single-target path-planning task.

Based on the algorithmic execution outcomes presented in Figure 5 and Table 3, within the context of single-target path-planning tasks, all algorithms successfully avoided obstacles and reached the target. Notably, the TD3 and ESPP-RL algorithms exhibited comparable results, whereas the SAC algorithm demonstrated lower reward values and required more steps, which is indicative of higher energy consumption. More specifically, despite requiring a comparable number of steps as TD3, the ESPP-RL algorithm achieved higher reward values, attributed to its PID-regulated action output, which facilitated optimal maneuvers, resulting in reduced vertical deviations and consequently higher rewards. Conversely, as depicted in Figure 5a, while the SAC algorithm initially demonstrated comparable performance to other algorithms during the initial stages of path planning, it struggled to reach the target's decision boundary towards the end due to disturbances caused by turbulent wind, hindering its ability to precisely navigate within the target's vicinity.



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

Figure 5. The results of SAC, TD3, and ESPP-RL in a single-target path-planning environment. (**a**) SAC; (**b**) TD3; (**c**) ESPP-RL.

Table 3. The reward and steps of algorithms under a single-target path-planning environment task.

	SAC	TD3	ESPP-RL
Reward	123.96	156.86	160.32
Steps	135	66	68

4.2.2. Multi-Target Path-Planning Environment

To evaluate the decision-making capability and path optimization efficiency of the UAV, the number of target points is varied within the multi-target path-planning environment. The training results are shown in Figure 6 and Table 4. In single-target path planning, emphasis is placed on the algorithm's resilience against disturbances in turbulent wind. Conversely, in multi-target scenarios, the algorithm dynamically selects path nodes based on the locations of the targets, necessitating intricate decision-making.

Table 4. The reward and steps of algorithms under a multi-target path-planning environment task.

		SAC	TD3	ESPP-RL
Three-target	Reward	326.85	397.66	421.14
	Steps	313	248	222
Four-target	Reward	407.31	487.98	528.40
	Steps	458	339	326
Five-target	Reward	461.73	587.59	615.77
	Steps	494	370	353

As depicted in Table 4, the ESPP-RL algorithm surpasses other methods in terms of both step count and reward value when tasked with 3, 4, and 5 target points. This superiority stems from the algorithm's imposition of energy consumption constraints, which are decomposed via Equation (17) to minimize overall energy expenditure during path planning, thereby yielding superior traversal solutions. In contrast, the SAC algorithm's action output is significantly perturbed by turbulent wind, hindering its ability to swiftly reach target points, resulting in a higher cumulative step count and lower reward value. While the TD3 algorithm exhibits comparable performance to ESPP-RL in single-target path planning, it notably underperforms in multi-target tasks due to its focus on optimizing individual target paths, neglecting intertarget relationships, leading to locally optimal solutions in the broader context.

The proposed algorithm can output actions by combining the environment information and PID controller to ensure more accurate actions. Therefore, the algorithm converges faster under turbulent wind interference and maintains a high path-planning success rate, which reflects the strong adaptability of the proposed algorithm. The proposed algorithm has a smoother path. When approaching a moving target, its strong robustness

enables it to accurately reach the target location, reducing the number of steps required and the cumulative distance. The experimental results demonstrate that in a multi-target path-planning environment, the ESPP-RL algorithm exhibits outstanding performance. It efficiently switches between multiple targets, planning both short and safe paths, resulting in a reduced accumulated distance. In summary, the ESPP-RL algorithm, with its precise action control, fast convergence speed, high success rate in path planning, smooth path characteristics, and robustness, demonstrates remarkable superiority in multi-target scenarios.

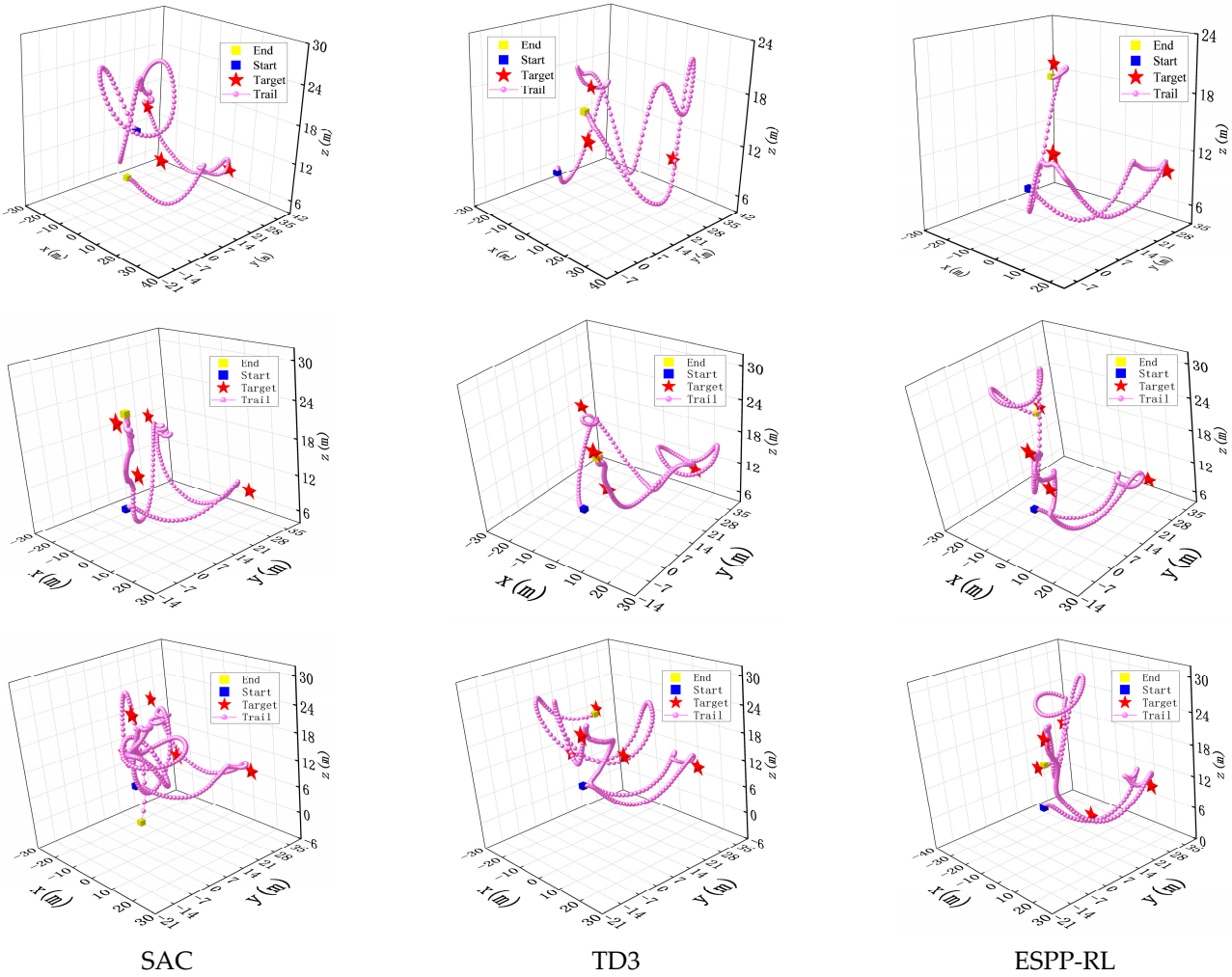


Figure 6. The results of SAC, TD3, and ESPP-RL in a multi-target path-planning environment. On the left is the SAC algorithm and in the middle is the TD3 algorithm. On the right is the ESPP-RL algorithm. The number of target points from top to bottom is 3, 4, and 5.

5. Conclusions

This paper proposes a reinforcement learning-based energy-saving path-planning algorithm for a UAV in complex 3-D turbulent wind. Based on the RL framework, the algorithm takes into account various flight models of the UAV, including vertical descent, vertical ascent, and horizontal flight, while paying particular attention to battery energy limitations. Through end-to-end training methods, the algorithm designs state features and reward functions for the entire mission process, utilizing a single policy network to achieve adaptive path planning and dynamic battery management. An optimization strategy suitable for locally observable conditions is also introduced, enabling the policy network to efficiently process dynamic dimensional observation features, thus improving the adaptability of path planning for drones with limited energy consumption. Experimental simulation results demonstrate that the proposed algorithm outperforms other drone

algorithms in terms of robustness, especially in complex 3-D turbulent wind, ensuring that drones can complete tasks in a time-efficient and collision-free manner, further bridging the gap between drone theoretical research and practical applications.

Future research will include exploring multi-UAVs' collaboration in complex 3-D turbulent wind to keep the UAVs stable, achieving better energy consumption, realizing autonomous obstacle avoidance and path planning. Additionally, we will strive to elevate the versatility of our algorithms to accommodate a wider range of UAV types, ensuring their applicability across diverse platforms and missions, and verifying the proposed control strategies with real UAVs.

Author Contributions: Conceptualization, S.C.; methodology, S.C. and Y.M.; software, Y.M.; validation, S.C., Y.M. and X.W.; formal analysis, X.W.; investigation, Q.L.; data curation, J.X.; writing—original draft preparation, S.C. and X.W.; writing—review and editing, X.W. and J.X.; visualization, Q.L.; supervision, S.C. and Y.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Technology Project of China Southern Power Grid Company Limited, specifically under the grant titled "Development and Application of Distribution Network Inspection Drone Based on Electric Field Coupled Wireless Charging Technology", grant number GXKJXM20222172.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: Shaonan Chen, Yuhong Mo, Xiaorui Wu, Jing Xiao and Quan Liu were employed by Electric Power Science Research Institute of Guangxi Power Grid Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Zeng, Y.; Zhang, R.; Lim, T.J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Commun. Mag.* **2016**, *54*, 36–42. [CrossRef]
2. Agrawal, N.; Bansal, A.; Singh, K.; Li, C.-P.; Mumtaz, S. Finite block length analysis of RIS-assisted UAV-based multiuser IoT communication system with non-linear EH. *IEEE Trans. Commun.* **2022**, *70*, 3542–3557. [CrossRef]
3. Zhang, Y.; Li, J.; Zhang, L.; Zhao, N.; Tang, W.; Wang, R.; Xiong, K. Energy consumption optimal design of power grid inspection trajectory for UAV mobile edge computing node. In Proceedings of the 2021 6th Asia Conference on Power and Electrical Engineering (ACPEE), Chongqing, China, 8–11 April 2021; pp. 1316–1321.
4. Aboudonia, A.; Rashad, R.; El-Badawy, A. Composite hierarchical anti-disturbance control of a quadrotor UAV in the presence of matched and mismatched disturbances. *J. Intell. Robot. Syst.* **2018**, *90*, 201–216. [CrossRef]
5. Tseng, C.-M.; Chau, C.-K. Personalized prediction of vehicle energy consumption based on participatory sensing. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3103–3113. [CrossRef]
6. Aharon, I.; Kuperman, A. Topological Overview of Powertrains for Battery-Powered Vehicles with Range Extenders. *IEEE Trans. Power Electron.* **2011**, *26*, 868–876. [CrossRef]
7. Ahmed, S.; Mohamed, A.; Harras, K.; Kholief, M.; Mesbah, S. Energy efficient path planning techniques for UAV-based systems with space discretization. In Proceedings of the 2016 IEEE Wireless Communications and Networking Conference, Doha, Qatar, 3–6 April 2016; pp. 1–6.
8. Di Franco, C.; Buttazzo, G. Energy-aware coverage path planning of UAVs. In Proceedings of the 2015 IEEE International Conference on Autonomous Robot Systems and Competitions, Vila Real, Portugal, 8–10 April 2015; pp. 111–117.
9. Zeng, Y.; Zhang, R. Energy-efficient UAV communication with trajectory optimization. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 3747–3760. [CrossRef]
10. Chan, C.; Kam, T. A procedure for power consumption estimation of multi-rotor unmanned aerial vehicle. *Proc. J. Phys. Conf. Ser.* **2020**, *1509*, 012015. [CrossRef]
11. Abeywickrama, H.V.; Jayawickrama, B.A.; He, Y.; Dutkiewicz, E. Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance. *IEEE Access* **2018**, *6*, 58383–58394. [CrossRef]
12. Zeng, Y.; Xu, J.; Zhang, R. Energy minimization for wireless communication with rotary-wing UAV. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2329–2345. [CrossRef]
13. Yan, H.; Chen, Y.; Yang, S.-H. New energy consumption model for rotary-wing UAV propulsion. *IEEE Wirel. Commun. Lett.* **2021**, *10*, 2009–2012. [CrossRef]
14. Yang, Z.; Xu, W.; Shikh-Bahaei, M. Energy efficient UAV communication with energy harvesting. *IEEE Trans. Veh. Technol.* **2020**, *69*, 1913–1927. [CrossRef]

15. Gao, N. Energy model for UAV communications: Experimental validation and model generalization. *China Commun.* **2021**, *18*, 253–264. [CrossRef]
16. Gong, H.; Huang, B.; Jia, B.; Dai, H. Modelling Power Consumptions for Multi-rotor UAVs. *arXiv* **2022**, arXiv:2209.04128v1.
17. Hung, J.Y.; Gonzalez, L.F. On parallel hybrid-electric propulsion system for unmanned aerial vehicles. *Prog. Aerosp. Sci.* **2012**, *8*, 1–17. [CrossRef]
18. Lee, B.; Kwon, S.; Park, P. Active power management system for an unmanned aerial vehicle powered by solar cells, a fuel cell, and batteries. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 3167–3177. [CrossRef]
19. Khayyam, H.; Bab-Hadiashar, A. Adaptive intelligent energy management system of plug in hybrid electric vehicle. *Energy* **2014**, *69*, 319–335. [CrossRef]
20. Bongermino, E.; Mastrorocco, F.; Tomaselli, M.; Monopoli, V.G.; Naso, D. Model and energy management system for a parallel hybrid electric unmanned aerial vehicle. In Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017.
21. Bongermino, E.; Tomaselli, M.; Monopoli, V.G. Hybrid aeronautical propulsion: Control and energy management. *IFAC Pap. Online* **2017**, *50*, 169–174. [CrossRef]
22. Lei, T.; Min, Z.; Fu, H. Dynamic balance energy management strategy for hybrid Power Supply of fuel cell UAV. *Acta Aeronaut. Sin.* **2020**, *41*, 324048. (In Chinese)
23. Yan, M.; Zhang, L.; Jiang, W.; Chan, C.A.; Gygax, A.F.; Nirmalathas, A. Energy Consumption Modeling and Optimization of UAV-Assisted MEC Networks Using Deep Reinforcement Learning. *IEEE Sens. J.* **2024**, *24*, 13629–13639. [CrossRef]
24. Khaghani, M.; Skaloud, J. VDM-based UAV attitude determination in absence of IMU data. In Proceedings of the European Navigation Conference, ENC 2018, Gothenburg, Sweden, 14–17 May 2018; pp. 84–90.
25. Mahony, R.; Hamel, T.; Pflimlin, J.-M. Nonlinear complementary filters on the special orthogonal group. *IEEE Trans. Autom. Control* **2008**, *53*, 1203–1217. [CrossRef]
26. Lyu, P.; Lai, J.; Liu, J.; Liu, H.H.T.; Zhang, Q. A thrust model aided fault diagnosis method for the altitude estimation of a quadrotor. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 1008–1019. [CrossRef]
27. Miranda-Moya, A.; Castañeda, H.; Wang, H. Fixed-Time Extended Observer-Based Adaptive Sliding Mode Control for a Quadrotor UAV under Severe Turbulent Wind. *Drones* **2023**, *7*, 700. [CrossRef]
28. Altman, E. *Constrained Markov Decision Processes: Stochastic Modeling*, 1st ed.; Routledge: New York, NY, USA, 1999.