

Article

Automatic Landing Control for Fixed-Wing UAV in Longitudinal Channel Based on Deep Reinforcement Learning

Jinghang Li ¹, Shuting Xu ^{1,*} , Yu Wu ^{2,3}  and Zhe Zhang ⁴¹ College of Engineering, Beijing Forestry University, Beijing 100083, China; lijinghang@bjfu.edu.cn² College of Aerospace Engineering, Chongqing University, Chongqing 400044, China³ School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798, Singapore⁴ School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China

* Correspondence: xushuting@buaa.edu.cn

Abstract: The objective is to address the control problem associated with the landing process of unmanned aerial vehicles (UAVs), with a particular focus on fixed-wing UAVs. The Proportional–Integral–Derivative (PID) controller is a widely used control method, which requires the tuning of its parameters to account for the specific characteristics of the landing environment and the potential for external disturbances. In contrast, neural networks can be modeled to operate under given inputs, allowing for a more precise control strategy. In light of these considerations, a control system based on reinforcement learning is put forth, which is integrated with the conventional PID guidance law to facilitate the autonomous landing of fixed-wing UAVs and the automated tuning of PID parameters through the use of a Deep Q-learning Network (DQN). A traditional PID control system is constructed based on a fixed-wing UAV dynamics model, with the flight state being discretized. The landing problem is transformed into a Markov Decision Process (MDP), and the reward function is designed in accordance with the landing conditions and the UAV's attitude, respectively. The state vectors are fed into the neural network framework, and the optimized PID parameters are output by the reinforcement learning algorithm. The optimal policy is obtained through the training of the network, which enables the automatic adjustment of parameters and the optimization of the traditional PID control system. Furthermore, the efficacy of the control algorithms in actual scenarios is validated through the simulation of UAV state vector perturbations and ideal gliding curves. The results demonstrate that the controller modified by the DQN network exhibits a markedly superior convergence effect and maneuverability compared to the unmodified traditional controller.

Keywords: fixed-wing UAV; automatic landing control; parameter tuning; deep reinforcement learning; Deep Q-learning Network (DQN)



Citation: Li, J.; Xu, S.; Wu, Y.; Zhang, Z. Automatic Landing Control for Fixed-Wing UAV in Longitudinal Channel Based on Deep Reinforcement Learning. *Drones* **2024**, *8*, 568. <https://doi.org/10.3390/drones8100568>

Academic Editor: Agostino De Marco

Received: 14 September 2024

Revised: 6 October 2024

Accepted: 8 October 2024

Published: 10 October 2024



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

1. Introduction

The rapid development of science and technology has led to a growing prevalence of Unmanned Aerial Vehicles (UAVs) in a number of fields, including military operations, agriculture, police, cartography, and surveillance. UAVs typically exhibit limitations in size and endurance, and the manual recovery of UAVs in unknown and complex landing environments is challenging, which impedes continuous operation [1–3]. It can be reasonably deduced, therefore, that the deployment of a ground operation platform with the objective of facilitating the automatic landing and subsequent operation of UAVs represents a superior solution to the existing problem [4,5]. Therefore, the automatic landing and subsequent recovery of the UAV represents a superior solution to the issue of UAV endurance during operation. This approach allows for the optimal utilization of the UAV in actual operational scenarios, while reducing the necessity for manual equipment inspection and maintenance.

For the automatic landing control of UAVs, many landing control methods have been proposed. One of the most classic and widely used is PID control, which has a simple, stable, and reliable structure, and can generally meet the needs of fixed-wing UAV control [6]. In actual situations, in order to make the controller achieve the desired needs, tuning of the PID parameters is often required, but because of its control input surface, adjusting the input parameters is very complicated [7]. Especially in nonlinear flight control, the results obtained by considering external perturbations are often different from expectations [8], and parameter tuning is cumbersome in practice, and there may be problems such as local optimal solutions [9]. In addition, in solving the dead zone, there are improvements in adaptive and robust controls based on the conventional PID control, and the rudder problem also has some advantages. In [10], real-time learning was applied to train the neural network using the gradient descent of an error function to adaptively update weights. The design of the adaptive control system optimized the landing performance. In [11], a robust visual servoing control approach was proposed to address the aircraft landing problem, which has significant improvement in ameliorating the effects of time-varying uncertainties, including parameter uncertainties and external disturbances. Zhen [12] proposed a robust preview control scheme and an autoregressive (AR) model prediction scheme that were originally applied to the automatic carrier landing system (ACLS). In ref. [13], a comprehensive examination of UAV disturbances during landing, encompassing atmospheric turbulence and low-altitude noise, was conducted. The devised secure and resilient automatic landing controllers were demonstrated to be capable of converging on the residuals within a fixed time frame, exhibiting a degree of adaptability.

When using conventional control situations, the mathematical calculation of parameters in nonlinear cases is very cumbersome. With rapid developments in the field of artificial intelligence, the integration of deep learning methods of intelligent control applications increased; this method does not need to manually debug and calculate the input surface parameters one by one, which is convenient for solving problems, especially in nonlinear situations [14,15]. Tang et al. [16] solved the problem that the current DRL-based solutions could not be generalized in unseen environments in landing control. Qing et al. [17] proposed an adaptive Radial Basis Function Neural Network (RBFNN) to estimate the unknown disturbances and apply it to the design of the controller by the backstepping technique. Wang et al. [18] proposed using the DDPG (Deep Deterministic Policy Gradient) to control the landing of drones and study the effect of hyperparameters and different networks on training. The DQN (Deep Q-Network) algorithm was first proposed by DeepMind of Google in 2013 [19] and proved to be very stable over a decade of use, compared to the continuity algorithm, with DQN employing discretization. This improves the data efficiency and robustness of the control system and reduces the amplification of uncertainties in the system due to in-flight noise and other disturbances [20]. Due to the “black box” effect of AI learning [21], it is difficult to provide human-understandable explanations for why a neural network solution model reaches a decision, which involves providing human-understandable explanations for why a model reaches a decision and how it works. The importance of explaining black-box modeling, especially in high-stakes decision problems, such as finance, aviation, healthcare, etc. [22]. It is difficult to address the issue of building trust in models for large-scale use in flight control, especially in landing progress.

Accordingly, this paper puts forth a methodology that integrates traditional PID control with a deep reinforcement learning network, DQN, which is mainly used in this paper, allowing for the automated landing control of UAVs to a designated landing zone. The principal contributions of this paper can be enumerated as follows:

- (1) PID control represents a fundamental element of the UAV landing controller, whereby the objective is to achieve convergence of the UAV flight state parameters and landing conditions by means of adjusting the parameters in the control model in response to a range of landing conditions and perturbations. The objective is to enhance the controllers' stability and noise resistance following parameter tuning.

- (2) The flight state of the UAV is discretized into the controller of the incremental PID combined with the deep learning framework DQN for several training iterations. The designed reward function outputs the reward Q value of the training, and the optimal PID controller model is obtained after the convergence of the reward Q value is reached. This approach allows for the interpretability of the deep learning network model to be achieved.
- (3) Experiments are conducted in order to ascertain the efficacy and robustness of the proposed control model, which is designed to control the UAV to land automatically on the ground platform in the event of certain external perturbations. The parameters in the control model are evaluated in conjunction with conventional control methods in order to determine their effectiveness.

The following Section outlines the paper as follows: Section 2 introduces the model of fixed-wing UAV and derives the state space equations; Section 3 provides an explanation of the basic controlling algorithmic principles involved in this paper, including the PID and the deep reinforcement learning DQN structure; Section 4 specifies the controller structure and the neural network framework designed in this paper. In Section 5, the control structure of this paper is simulated and tested to verify its validity and assess its performance under simulated atmospheric external conditions proving the promotion of the control algorithm. Section 6 presents a summary of the conclusions of the current work and outlines future developments.

2. Fixed-Wing UAV Modeling

The present study focuses on the utilizations of fixed-wing UAV as research object. In dynamic modeling, the model is simplified to a reasonable extent, with the influence of factors such as the elastic deformation of the aircraft body and the rotation being ignored. In light of the fact that the UAV is situated in an environment in close proximity to the earth's surface and at a relatively low altitude, it is reasonable to assume that the gravitational force remains constant and that the UAV's airframe can be treated as a rigid body. The force analysis of a six-degree-of-freedom fixed-wing aircraft is illustrated in Figure 1. In this paper, we primarily model the fixed-wing UAV in both the ground coordinate system and the airframe coordinate system, subsequently analyzing the position change (ξ) and attitude change (η) of the UAV. The state vector x_p of the UAV is defined as follows:

$$x_p = [x, y, z, \phi, \theta, \psi]^T \quad (1)$$

where ϕ represents the roll angle, θ represents the pitch angle, and ψ represents the yaw angle, which are denoted as the position change $\xi = [x, y, z]^T$ and attitude change $\eta = [\phi, \theta, \psi]^T$, respectively.

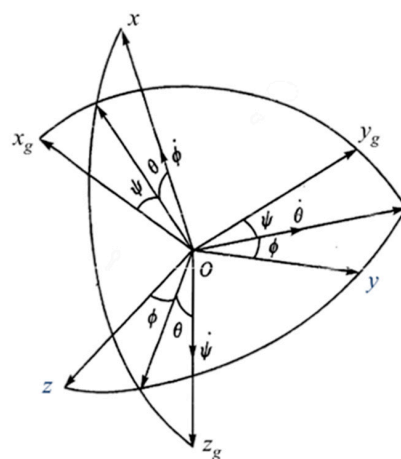


Figure 1. Six degrees of freedom fixed-wing UAV.

The objective is to analyze the motion of the UAV during the landing process. The kinematics equations situated at the center of mass of the body are established in the body coordinate system in order to obtain the UAV kinematics equations and the rotational kinematics equations:

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = L_{bg}^T \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} \cos \phi \\ -\dot{\theta} \sin \phi \end{bmatrix} + L_{bg} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (3)$$

where V_x, V_y, V_z are the three components of the UAV inertial velocity in the ground coordinate system; u, v, w are the three components of the UAV inertial velocity in the airframe coordinate system; p, q, r are the three components of the UAV angular velocity in the airframe coordinate system; g is the ground coordinate system, and b is the UAV fuselage coordinate system, and L_{bg} is the coordinate transformation matrix from the ground to the fuselage coordinate system, and the computation formula is:

$$L_{bg} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \theta \cos \psi \sin \phi - \sin \psi \cos \phi & \sin \theta \sin \psi \sin \phi + \cos \psi \cos \phi & \sin \phi \cos \theta \\ \sin \theta \cos \psi \cos \phi + \sin \psi \sin \phi & \sin \theta \sin \psi \cos \phi - \cos \psi \sin \phi & \cos \phi \cos \theta \end{bmatrix} \quad (4)$$

In order to analyze the forces acting on the UAV during the landing flight, it is necessary to consider the effects of gravity, lift, thrust, and air resistance, in the air. In accordance with the fundamental principles of flight dynamics, the dynamic equations are derived at the center of mass of the airframe within the airframe coordinate system. This enables the parallel equations of motion and rotational equations of the UAV to be established:

$$\begin{bmatrix} X - mg \sin \theta \\ Y + mg \sin \phi \cos \theta \\ Z + mg \cos \phi \cos \theta \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx} \dot{p} + (I_{zz} - I_{yy})qr - I_{zx}(pq + \dot{r}) \\ I_{yy} \dot{q} + (I_{xx} - I_{zz})rp + I_{zx}(p^2 - r^2) \\ I_{zz} \dot{r} + (I_{yy} - I_{xx})pq + I_{zx}(qr - \dot{p}) \end{bmatrix} \quad (6)$$

where m is the total mass of the UAV; g is the gravitational acceleration; I_{xx} is the roll axis moment of inertia; I_{yy} is the pitch axis moment of inertia; I_{zz} is the yaw axis moment of inertia roll; and I_{zx} is the yaw inertia product.

Taking into account the incremental force and aerodynamic coefficients generated by each rudder control surface of the UAV, including the internal and external actuator and rudder deflection, the aerodynamic force and coefficients can be found in ref. [23], as shown as follows:

$$\left\{ \begin{array}{l} C_D = C_{D0}(\alpha, \beta) + C_{D\delta}(\delta_a, \delta_e, \delta_r) + C_{Dq} \cdot \frac{qc_{ref}}{2V} \\ C_S = C_{S0}(\alpha, \beta) + C_{S\delta}(\delta_a, \delta_e, \delta_r) + C_{Sp} \cdot \frac{pb_{ref}}{2V} + C_{Sr} \cdot \frac{rb_{ref}}{2V} \\ C_L = C_{L0}(\alpha, \beta) + C_{L\delta}(\delta_a, \delta_e, \delta_r) + C_{Lq} \cdot \frac{qc}{2V} \\ C_l = C_{l0}(\alpha, \beta) + C_{l\delta}(\delta_a, \delta_e, \delta_r) + C_{lp} \cdot \frac{pb}{2V} + C_{lr} \cdot \frac{rb}{2V} \\ C_m = C_{m0}(\alpha) + C_{m\delta}(\delta_a, \delta_e, \delta_r) + C_{mq} \cdot \frac{qc}{2V} \\ C_n = C_{n0}(\alpha) + C_{n\delta}(\delta_a, \delta_e, \delta_r) + C_{np} \cdot \frac{pb}{2V} + C_{nr} \cdot \frac{rb}{2V} \end{array} \right. \quad (7)$$

where C_D, C_S, C_L are the drag, side force, and lift coefficients, b_{ref} represents the wing span, and c_{ref} represents the mean aerodynamic chord.

The UAV is flying in a constant straight line along the ideal chute as the base motion state, and the parameters of the leveling state when the UAV is landing are the leveling speed $v_{trim} = 69.96$ m/s, the leveling angle of attraction $\alpha_{trim} = 8.3^\circ$, the pitching speed $q_{trim} = 0^\circ/s$, and the pitching angle of the base state $\theta_{trim} = 5.3^\circ$; the parametric model of the UAV utilized in this study is based on the model in ref. [24].

In regard to the control of fixed-wing UAV attitude stabilization, a Stability Augmentation System (SAS) was integrated into the conventional controller to provide feedback channel compensation for the leveling angle of attraction α and pitching speed q . Further details regarding the specific controller design ideas can be found in Section 4.1

During the landing flight process, the UAV baseline motion is a constant straight-line flight along the ideal downward trajectory, and the change in the flight state of the UAV after perturbation is not obvious compared with the baseline motion, so the above nonlinear model is linearized under the constant straight-line flight state. This paper takes the longitudinal landing task as an example, establishes the UAV longitudinal linear small perturbation dynamics model, and the state space equations are expressed as follows:

$$\dot{x}p = Ap xp + Bp \delta \tag{8}$$

where the UAV state vector $\dot{x}p = [u/V, \alpha, \theta, q, h/V]^T$, the vector element u/V represents the ratio of the body-axis-direction velocity perturbation u to the airspeed V , α represents the angle-of-approach perturbation; θ represents the pitch angle perturbation; q represents the pitch-angle velocity perturbation; and h/V represents the ratio of the altitude perturbation h to the airspeed V .

The UAV control maneuver input $\delta = [\delta_e, \delta_{PL}]$, which includes the UAV aerodynamic rudder including the full-motion flat-tail elevator rudder δ_e , and the external control maneuver inputs throttle stick deflection angle δ_{PL} . The matrices A_p and B_p are defined as the UAV's dynamical state matrix and the control inputs matrix, respectively, with the following parameters:

$$A_p = \begin{bmatrix} -0.0705 & 0.0475 & -0.1403 & 0 & -0.000058 \\ -0.3110 & 0.3430 & 0 & 0.99133 & 0.00102 \\ 0 & 0 & 0 & 1 & 0 \\ -0.0218 & -1.1660 & 0 & -0.2544 & 0 \\ 0 & -1 & 1 & 0 & 0 \end{bmatrix} \tag{9}$$

$$B_p = \begin{bmatrix} 0.0121 & 0.2316 \\ -0.0721 & -0.0388 \\ 0 & 0 \\ -1.1850 & 0.0023 \\ 0 & 0 \end{bmatrix} \tag{10}$$

Furthermore, the model incorporates the dynamics of the actuators and engine. The actuators are represented by a second-order model, whereas the engine is modeled as a first-order inertia model with a time delay unit. The parameters and their respective limits are presented in Table 1 below.

Table 1. Parameters of Actuator and Engine models.

Actuators					
Symbol	Model	Natural frequency (rad/sec)	Damping ratio	Amplitude limit (deg)	Rate limit (deg/sec)
δ_{el}/δ_{er}	Second order	50	0.8	[−30, 30]	[−90, 90]
Engine					
Symbol	Model	Frequency (rad/sec)	Thrust limit (N)	Rate limit (N/s)	
δ_{pl}/δ_{pr}	First order	2.4	[4448, 44,480]	[−6450, 8363]	

3. Principle of Landing Controller

The landing control of the UAV is based on the conventional PID guidance law, which is combined with a reinforcement learning algorithm to enable the automatic tuning of the gain parameters, thereby optimizing the conventional controller.

3.1. Design of Conventional UAV-Landing Controller with PID-Based Guidance Law

In the development of PID control methods, many approaches have been proposed for PID tuning [25]. For example, in the paper [26], a mathematical-based approach using Taylor expansion approximation was proposed, but such computation using classical mathematical techniques is very complicated, especially in other second- or higher-order systems; Among the algorithms for PID tuning, strategies based on metaheuristic algorithms are widely used, which can provide near-optimal solutions in a reasonable time, e.g., PSO [27], due to their intelligible performance and ease of use and can better achieve global optimization, especially in solving higher-dimensional optimization, but the effect of its overshooting is not negligible; DQN networks can be constrained by designing the reward function to constrain its controller performance and adjustment margin. Therefore, in this paper, the P controller is designed to study the design of a control algorithm based on a deep reinforcement learning algorithm to adjust the PID parameters to optimize the control effects, taking the proportional gain K_p as an example.

The landing control of the fixed-wing UAV currently employs the conventional PID controller, which encompasses the system's necessities for expediency, precision, and stability. It is important to consider external disturbances, such as real atmospheric conditions, which cannot be ignored during the landing of a UAV. The basic control law of PID, as used in this paper, can be expressed as the following Equation (11):

$$u = u_d + K_p \times e + K_p \cdot \frac{1}{T_i} \times \int e + K_p \cdot T_d \times \frac{de}{dt} = u_d + u_c \quad (11)$$

where e is the input state vector in the UAV landing process, and u_d is the disturbances that are considered real atmospheric perturbations.

In the above equation with the gain tuning of the controller, too large a proportional gain K_p will not only increase the overshoot of the system but also make the system stability margin become smaller, or even unstable. At the same time, in actual situations, considering the wind disturbance or the influence of external disturbances, it is often very difficult to adjust gain to optimize the controller.

The overall control structure of the UAV landing process with control guidance law is as follows (Figure 2). In the primary UAV landing control system, PID guidance law is employed to achieve the separation control of the position and attitude of the object, which is achieved by processing the input state vectors for the UAV in a simulated landing environment. After the controller processing, the output of the aircraft actuator and engine control the controlling operation output $[\delta_e, \delta_{pl}]$, and input the controlling operation output into the fixed-wing UAV dynamics model. Meanwhile, in the control system, the objective is to enhance the motion characteristics of the UAV. To this end, the Stability Augmentation System (SAS) is employed to furnish feedback compensation for the attitude of the UAV. The particular superimposed SAS compensation control method will be elucidated subsequently with reference to the control system.

Compensation for UAV attitude elements in the state matrix longitudinal incremental PID guidance law is designed using the PID method as follows (the speed at which the UAV approaches the landing site is set by default):

$$q = \left(K_{PH} + K_{IH} \frac{1}{s} \right) \Delta H + K_{PH} \dot{H} \quad (12)$$

with H as the height.

$$\Delta\alpha = \left(K_{PS} + K_{IS} \frac{1}{s} \right) \Delta S + K_{PS} \dot{S} \tag{13}$$

with S as the side distance tracking errors.

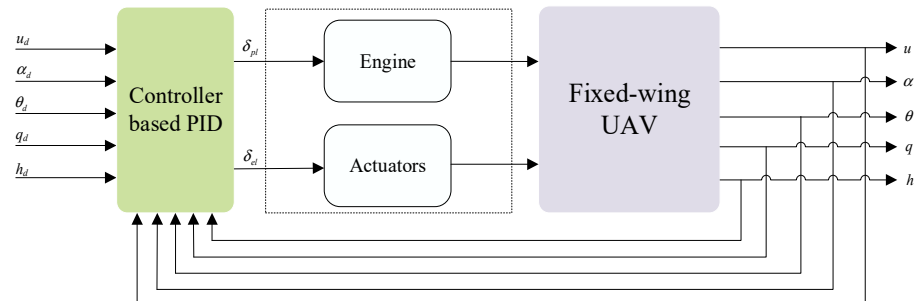


Figure 2. Control structure of UAV landing with PID-based guidance law.

3.2. Markov Decision Process

Considering that the landing control process in which the UAV and the environment with a landing platform interact to generate action commands can be regarded as a sequential decision-making process, it can be modeled using the Markov Decision Process (MDP) and solved using the Reinforcement Learning (RL) method.

The Markov decision process describes the process of interaction between the Agent and the Environment, which is the mathematical basis for the derivation of Reinforcement Learning theory. In the policy process, the state change of the system at the moment $t + 1$ is only related to the state and action at the moment t , and has nothing to do with the state and action at the previous moment. A Markov decision model is established, represented as a quintuple:

$$M = (S, A, P, \gamma, R) \tag{14}$$

S denotes the set of states of the intelligent body in the environment, i.e., the state space (s_t denotes the state at moment t , $s_t \in S$);

A denotes the set of all actions that can be performed by the intelligent body under the corresponding state (at denotes the state at moment t , $a_t \in A$);

P denotes the state transfer probability matrix of the intelligent body, describing the possibility that it can transfer from state s to state s' under action a , denoted as $P(s'|s, a)P_{s \rightarrow s'}^a$;

γ denotes the discount factor $\gamma \in [0, 1]$, which is used to calculate the reward cumulatively.

R denotes the set of rewards obtained from the state change s_t of the intelligent body, the value of the reward for executing the action a_t and transfer to the next state in the state s_t is recorded as $r(a_t, s_t)$, and also combining with the discount factor γ , the accumulated rewards can be expressed as:

$$M = (S, A, P, \gamma, R) \tag{15}$$

The environment in the algorithm of this paper mainly refers to the three-dimensional kinematics system jointly composed of the UAV and the landing platform, and in the research of this paper, the ground landing area will be simulated as the landing platform, and it is assumed that the UAV can realize the automatic landing after the distance and speed above the area reach certain conditions.

3.3. Principle of Reinforcement Learning Algorithm

RL is a process in which an intelligent body acts based on feedback from the environment, continuously interacts with the environment, with trial and error, and improves its actions through the rewards it receives. The framework of the UAV landing control system in this paper is shown in Figure 3.

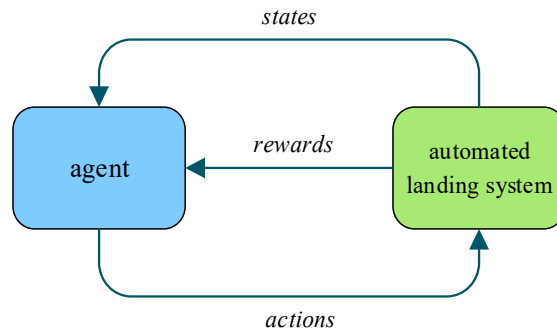


Figure 3. Enhanced Learning Systems Framework.

Using the sequential decision-making form of the Markov decision-making process, the intelligent body selects the action a_t according to the state of the UAV and the ground trolley at a certain moment, and the UAV transfers from the state s_t to the new state s_{t+1} under the joint action of the environment and the action, and at the same time, the environment gives a reward value r_t based on the reward function and feeds back to the intelligent body. When the intelligent body carries out the automatic landing control task of the UAV, the automatic landing system is the environment, and the UAV intelligent body continuously interacts with the environment, and the process is continuously cyclic, continuously improving the strategy and optimizing the selection of actions in the data obtained from the training, and ultimately obtaining the optimal strategy for realizing the automatic landing control through a large amount of training and learning.

Combined with the Dynamic Programming (DP) bellman equation, under the selection of the action that maximizes the reward, i.e., the optimal policy, the Q table is updated at each step of execution by the following formula until the intelligence reaches convergence.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a_t) - Q(s_t, a_t)] \quad (16)$$

In deep reinforcement learning due to our inability to accurately represent the Q-value, combining the network parameter w utilizes a function to approximate the action value function $Q(s, a)$:

$$Q(s, a) \approx Q(s, a, w) \quad (17)$$

Taking the target Q value as a label, the network is continuously updated by using the above formula, so that the final network output Q value approaches the target Q value to achieve convergence. Therefore, the loss function of Q network training is expressed as follows [28]:

$$L(w) = E \left[(r + \gamma \max_{a'} Q(s', a', w) - Q(s, a, w))^2 \right] \quad (18)$$

The DQN algorithm follows the idea of Q learning, but the Q value that can be obtained by selecting a certain action for a given state is instead calculated by a deep neural network. When the intelligent body carries out the automatic landing control task of the UAV, in order to better simulate reality, the combination of states and actions is almost inexhaustible, and it is almost impossible to select the optimal action by looking up the table. The flight state is used as an input to the neural network, and then the Q value of the action is obtained after the neural network algorithm.

3.4. Design of an Automatic Landing Controller Tuning Based on DQN

In the previous Section, the fundamentals of the DQN algorithm were briefly introduced. This Section focuses on the application of DQN to the tuning of the PID controller for UAV landing and the architecture of the used algorithm, as shown in Figure 4. The system receives information about the state of the UAV, including the position and attitude of the UAV, using a trained deep neural network to determine the intelligent adjustment method of gain. The training process is based on optimizing the objective function, which is the basis for learning and setting the action space of the algorithm output.

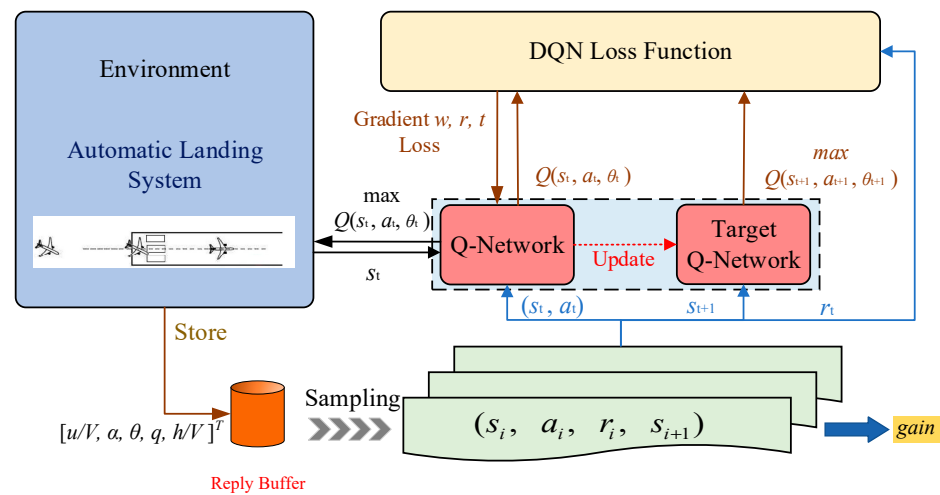


Figure 4. Architecture of DQN algorithm.

4. Controller and Algorithm Design

In the PID control of the landing process, the landing state of the UAV is greatly influenced by the controller parameter. In the UAV landing controller design, the landing routine PID control is firstly designed, while the initial gain K_i values without a large number of trainings are set, and the maneuvering inputs and flight state vectors obtained from the simulation are transferred to the deep neural network for reinforcement learning training, and the more stable and ideal gain K_i value is obtained through a large number of trainings. The framework of the PID parametric landing controller based on reinforcement learning optimization is shown in Figure 5.

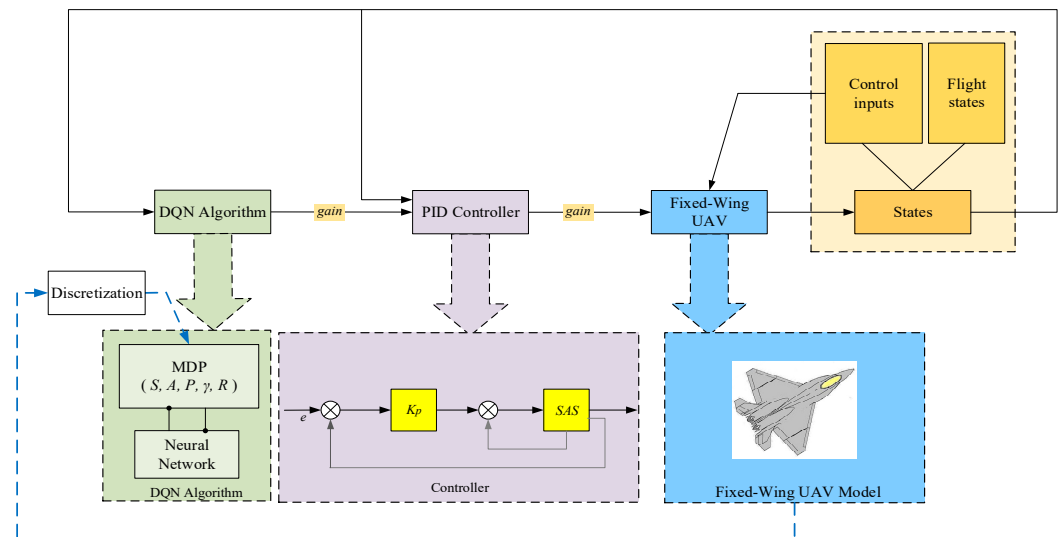


Figure 5. Optimized PID parametric landing controller framework based on RL.

4.1. Structure of Conventional Landing Control System

In the theoretical analysis of the controlled object fixed-wing UAV, its dynamics model used for the simulation is shown in Section 1. The controller design in this paper uses the PID controller as the external control of the UAV flight, and, at the same time, improves the motion characteristics of the UAV through the Stability Augmentation System (SAS) to improve the stability of the UAV. The landing control system is shown in Figure 6.

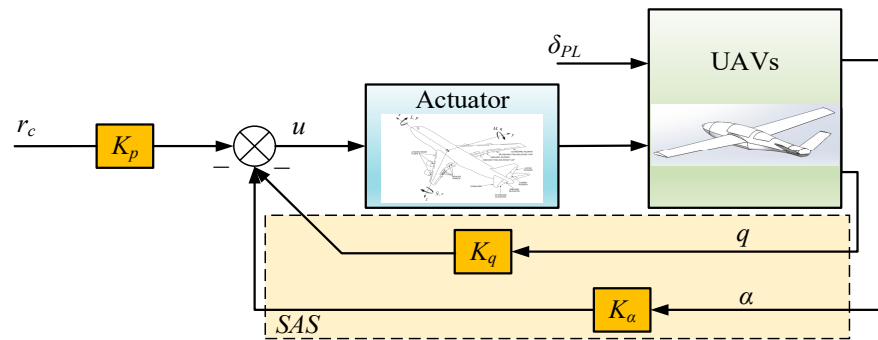


Figure 6. Conventional landing control system.

Since the actuator response of the UAV controller is much faster compared to the UAV body dynamics response [24], we used the stabilizing controller design by means of a short-period low-order approximation model. When talking about the state of the UAV, we focused on attitude stabilization, and kept only the elements related to the rate of heading angle and pitch angle in the state matrix of the UAV aircraft, with the simplified state vector being denoted as $x'_p = [\alpha, q]^T$. The design goal of the stabilization control system is to increase the short-period damping ratio to the neighborhood of the optimal damping ratio through the feedback of the state quantities, and, at the same time, to realize the feedback compensation for the corresponding feedback channels of the UAV attitude (angle of approach and pitch angular velocity).

4.2. Reinforcement Learning Algorithm Example

The DQN algorithm is a neural network method integrated on the basis of Q learning, and on the basis of Q learning, the target Q-value network is added, and the experience playback mechanism is used to construct an experience pool to remove data correlation, which solves the problem of training instability of the algorithm to update the network parameters. This chapter focuses on the DQN neural network framework designed in the article.

4.2.1. Intelligent Body Action Space

The algorithm in this paper is based on the deep learning DQN algorithm to realize the automatic tuning of UAV landing PID control parameters, which needs to determine the dimension of the designed PID parameters. The purpose of DQN design is to optimize the PID parameters and optimize the controller control effect through PID tuning. Since the controllers involved in this paper are P controllers, only the proportional gain parameter K_p is involved; therefore, the action space output by the DQN algorithm is the gain K_p .

Before inputting the PID parameter vectors, the data need to be preprocessed. Due to the continuous flight data characterization for UAV landing control and the independent discrete sample set of the DQN algorithm, this paper adopts incremental PID control; i.e., the sample space in landing control is discretized. During the training process, the PID parameters are chosen by selecting the action corresponding to the maximum Q value in the output layer of the network through the involved reward function.

4.2.2. Reward Function Design

The reward function is designed so that the intelligent body reaches the optimal response in the state through positive rewards to encourage certain action strategies and negative rewards (punishment) to discourage some action responses. During the training of the intelligent body, the problem of reward sparsity may lead to slow learning or even difficulty in reaching convergence, and the final result is poor. In this paper, we design milestones through the reward function reshaping method to guide the program toward the desired state to ensure that convergence is reached to improve training robustness. In the field of flight control especially in the progress of landing, accuracy and smoothness must be prioritized and considered by control system designers [29].

Given our focus on landing tasks, our objective is to ensure that the UAV can achieve the desired landing conditions with as high a quality as possible. It must be ascertained whether the UAV can reach the landing conditions, by considering whether the perturbations $|h - h_0|$ and $|u - u_0|$ of the altitude and the velocity in the direction of the body axis at the time of landing, compared with the ideal flight path, can converge to zero. During the landing process, UAV attitude is the most influential cause of force analysis at the moment of landing besides control inputs [30]. So, we consider the perturbations $|q - q_0|$ of the pitch angular velocity, thus correcting the attitude of the airplane. Since external perturbations such as UAVs during the landing process will affect the attitude and other parameters of the UAV, and considering that the measurement accuracy may change in the usage scenario and there is some coupling between the perturbations of the state vectors, we can relax the requirement of the state volume tracking accuracy to some extent. In the study of this paper, the state volume error of the above study can be unpunished within a certain range. In this paper, a hybrid reward signal is designed based on flight control considerations, and the design of the specific reward function includes the following three parts:

- Landing tracking accuracy

The height perturbation describes the deviation of the UAV's current altitude from the ideal flight path, which can intuitively describe the landing deviation of the UAV and is an important basis for judging flight quality. We add a penalty term based on the altitude tracking error to the DQN reward function and assign weight Q_1 to reflect the relative importance of this penalty term. Considering the existence of certain perturbations in the data, the requirement of altitude tracking accuracy can be relaxed to some extent. In other words, the height tracking error can be unpunished within a certain range, and 0.05 m is set as an acceptable range in this study.

$$r_1 = h^T Q_1 h = \begin{cases} (h - h_0)^T Q_1 (h - h_0) & |h - h_0| \geq 0.05 \text{ m} \\ 0 & |h - h_0| < 0.05 \text{ m} \end{cases} \quad (19)$$

When considering the optimization of the landing flight path, the velocity affects the state quantities in the landing analysis, such as h and α , and also has an indirect effect on the flight path optimization. At the same time, the speed also has a greater impact on flight safety; a too-large landing velocity may lead to loss of control of the UAV, so in this paper we set 69.96 m/s as the ideal landing speed u_0 . Therefore, the velocity information can be used as a penalty term with a weight of Q_2 . In addition, considering the influence of related interference, the penalty is only applied when it is greater than or equal to 3 m/s, as shown in the following equation.

$$r_2 = u^T Q_2 u = \begin{cases} (u - u_0)^T Q_2 (u - u_0) & |u - u_0| \geq 3 \text{ m/s} \\ 0 & |u - u_0| < 3 \text{ m/s} \end{cases} \quad (20)$$

- Flight attitude optimization

In light of our focus on optimizing the whole landing process, one of our objectives is to ensure good task attitude qualities. Pitch angular velocity is the differential pitch angle, being one of the main attitude angles that describes the UAV. The pitch angle mainly describes the attitude analysis of an airplane landing on the longitudinal axis, and the angular velocity as its differential can more quickly and finely characterize the magnitude of the change in the pitch angle q . The angle of the pitch angle is the same as the angle of the airplane landing on the longitudinal axis. Additionally, a threshold is established to permit a certain range of 0.05 deg/s during normal circumstances. The final penalty function for this term with a weight of Q_3 is expressed as follows:

$$r_3 = u^T Q_3 u = \begin{cases} (q - q_0)^T Q_3 (q - q_0) & |q - q_0| \geq 0.1 \text{ deg/s} \\ 0 & |q - q_0| < 0.1 \text{ deg/s} \end{cases} \quad (21)$$

The final reward function can be obtained by combining the three penalties mentioned above and taking a negative value:

$$r = -(r_1 + r_2 + r_3) = -h^T Q_1 h - u^T Q_2 u - q^T Q_3 q \quad (22)$$

4.2.3. Neural Network Framework

In the PID-based guidance law controller, the gain parameters in the controller need to be taken into account. Therefore, in this paper, the neural network structure of the corresponding DQN was built for prediction. In the experiment, the trained neural network model is applied to the prediction of control parameters in the landing process in real time. We choose the input side as a state vector that reflects the state and attitude of the fixed-wing UAV, which it receives from the UAV in real time for training. Considering the timeliness performance of the prediction network and the over-fitting of the model, we choose a neural network with two hidden layers.

Considering the establishment of the model. This paper selected the inputs to the neural network consisting of five-dimensional states: velocity perturbation in the direction of the body axis of the UAV; angle of attack; pitch angle; pitch angle velocity perturbation; and current altitude of the UAV landing process. Then, the corresponding output layer and the action space were connected so that the network could output the corresponding PID parameters. The structure of the neural network model is shown in Figure 7.

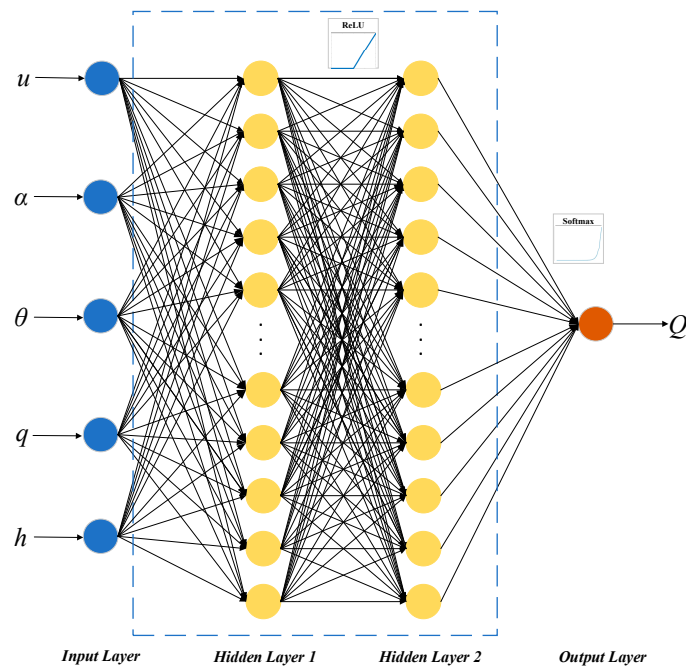


Figure 7. Neural network framework based on reinforcement learning.

The input layer of the model is defined as the state vector $\dot{x}p = [u/V, \alpha, \theta, q, h/V]^T$, which is denoted as the body-axis-direction velocity perturbation u , the angle-of-approach perturbation α , the pitch-angle perturbation θ , the pitch-angle velocity perturbation q , and the altitude perturbation h , respectively. Each sub-strategy network has two fully connected hidden layers, each of which is a full-connection layer.

Each fully connected hidden layer consists of 100 neurons and uses the *ReLU* (rectified linear unit) activation function, which is defined as follows:

$$ReLU(z) = \begin{cases} z, & \text{if } z > 0 \\ 0, & \text{if } z \leq 0 \end{cases} \quad (23)$$

Compared with other activation functions, the *ReLU* function not only saves a lot of calculations but also solves the problem of gradient disappearance. Moreover, due to the sparsity of the *ReLU* network, it is possible to better mine the relevant features of the model and fit the training data to achieve convergence.

In order to classify the output into N action classes, the node with the maximum probability (the maximum function output value) is selected, and the output layer of the network is activated by the *Softmax* function, which maps a value in $(-\infty, +\infty)$ to $(-1, +1)$, and the sum of its elements is 1, the formula is as follows:

$$\text{Softmax}(z) = \frac{e^{x_i}}{\sum_i e^{x_i}} \quad (24)$$

5. Simulation Experiment and Analysis

In order to evaluate the performance and efficacy of the proposed framework and controller, a series of simulation experiments were conducted using the software for experimentation and the UAV-Ground Landing Platform environment. The experiments compared the efficacy of traditional PID control-based methods with those based on DQN algorithms. Moreover, the study assessed the framework's capacity to evaluate anti-noise and disturbance performance, robustness, and the scope of applicability of the model in a generalized context.

5.1. DQN-Network Algorithm Training Experiment

In the simulation model presented in this paper, the gain parameter in the PID controller is calculated by the deep reinforcement learning algorithm. Hyperparameters play a crucial role in the performance of the deep reinforcement learning framework, which as employed in this paper are listed in Table 2. The simulation time is set to 60 s with a step size of 0.1 s for a total of 600 steps. To ensure that the training process covers a sufficient number of samples, a large number of sets (2000) is required to avoid the excessive effects of randomness and external perturbations. Greediness affects reinforcement learning's exploitation of an unknown action space to minimize the convergence of the loss function and achieve a globally optimal solution. An insufficient level of greediness may impede the controller's ability to learn, resulting in local optimization and the potential for missing the global optimal solution. Conversely, it may lead to difficulty in converging the trained network [16]. It is therefore necessary to select an appropriate level of greediness for the neural network based on experience and experimentation.

Table 2. The hyperparameter of the DQN algorithm.

Hyperparameter	Value
Steps	600
Episodes	2000
Hidden layer neurons (each hidden layer)	100
Greediness	0.90
Discount factor	0.99

The discount factor is a tuning parameter in the optimization algorithm, which is calculated by discounting the future rewards to the current moment, and the farther away from the current rewards, the bigger the discount, i.e., the smaller the value discounted to the current moment. Due to the requirement of flight control stability, the controller needs to have stable and reliable convergence, so the setting of the discount factor is designed to be relatively conservative. Taking into account reference [18], which also employs a high forgetting factor, we finally determine the forgetting factor to be 0.99.

5.2. Controlling Operation Input Experimentation

Prior to the simulation of the deep reinforcement learning DQN network in this paper, the effectiveness of the PID framework designed for UAV landing control in this paper should first be verified. To ensure reasonable and effective control over UAV landing, the manipulation inputs of the controller should achieve convergence. The manipulation input of the controller $\delta = [\delta_e, \delta_{PL}]$, δ_e stands for the full-motion flat-tail deflection angle, and δ_{PL} stands for the throttle stick deflection angle. The simulation results of the UAV maneuvering inputs are shown in Figure 8.

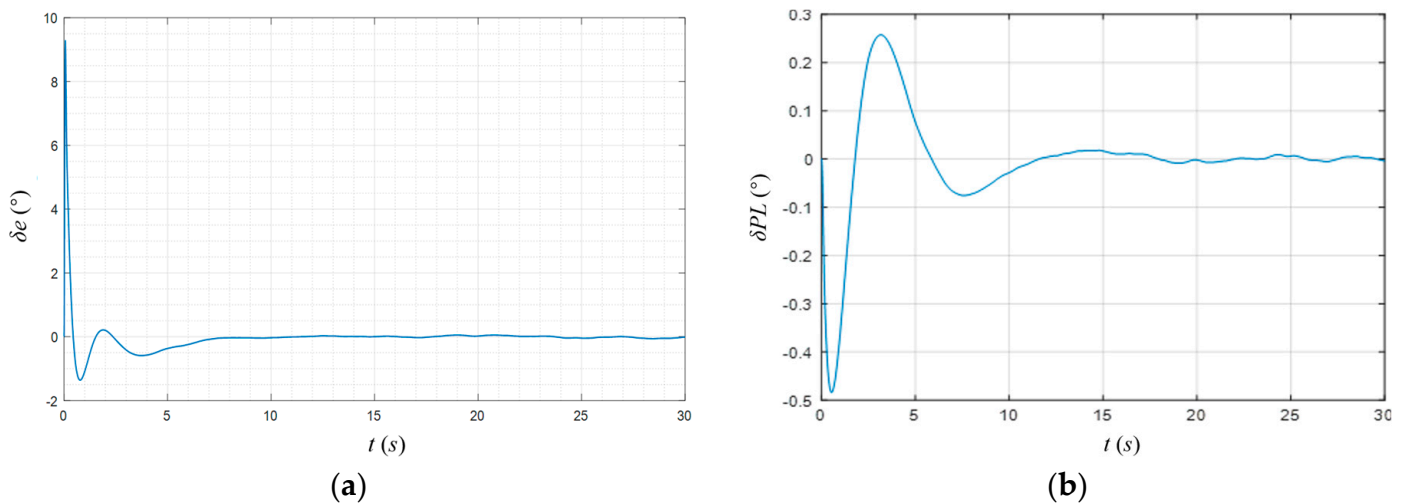


Figure 8. Working condition simulation results of uniform Controlling operation input. (a) Flat-tail deflection angle result, (b) Throttle stick deflection angle result.

As depicted in Figure 8, the full-motion flat-tail deflection angle of UAV δ_e and the throttle stick deflection angle δ_{PL} variations are within a reasonable range and finally converge to zero in finite time. The control inputs of the UAV with the described Newton–Euler equations of the UAV’s dynamic model can ensure that a reasonable landing flight path of the UAV is realized for control, and it also ensures that the state and attitude of the UAV are as expected [31].

5.3. Simulation Verification of Controller Landing Condition

Prior to implementation, the intelligent body algorithm is subjected to multiple training iterations to ascertain its efficacy and determine if the training reward value has reached a steady state. Subsequently, the landing system conditions for the UAV are simulated, and the flight state perturbation amount simulation is employed to ascertain whether the landing conditions have been met. In light of the possibility that the UAV may be subject to external perturbations in real-world scenarios, it is prudent to impose appropriate constraints on the degree of overshooting. It is essential to constrain the attitude of the UAV to an optimal level to guarantee the successful completion of the landing task. This can be achieved by reducing the gain of the attitude state vectors in a manner that prevents the algorithm from reaching convergence. In the context of training the module via the DQN method, the network’s output action gain can be restricted within the range of 0~1 (initial gain = 1). In the Reward function, $Q_1 = 0.05$, $Q_2 = 0.05$, and $Q_3 = 0.05$. The landing environment set by the trained controller is simulated, and the flight state perturbation amount is simulated to determine whether the landing conditions are met, and they are compared with an untrained conventional PID controller. After several training attempts, this Section verifies the effectiveness of the algorithm.

When training the algorithm, the fixed ground landing area is considered as the simulated landing platform, and the starting moment of the landing spot is located at the origin of the ground system while moving slowly and considering the effect of small perturbations. At the beginning of the simulation, the initial height deviation of the UAV at the landing starting point relative to the ideal descent trajectory is set to 1 m, and the UAV is below the ideal descent trajectory due to factors such as navigational errors or perceptual operational errors. The simulation conditions are as follows: At the initial moment, the landing platform is located on the ground, and the position is determined as the origin of the ground system. The distance difference between the UAV and the landing area in the horizontal direction is 831 m, and the height difference in the vertical direction is 50 m. The parameter settings of the specific training simulation are shown in Table 3, which mainly gives the initial position and velocity state initial value relationship between the UAV and the landing area.

Table 3. Training simulation parameter.

Parameter	Position	Speed	Track Angle	Pitch Angle
Value	(831, 0, 50)	70	3.5	2.71

During the landing flight, the flight state vector of the UAV is $[u, \alpha, \theta, q, h]^T$, including the deviation perturbation quantities of the velocity u , the angle of attack α , the pitch angle θ , the pitch angle velocity q , and the altitude h in comparison with the ideal glide slope. The results of the flight state working conditions simulation, based on the application of PID and DQN-PID controllers, are presented in Figure 9. As illustrated in Figure 9a–e, the flight state perturbation of the UAV under both PID and DQN-PID controllers exhibits a tendency to converge within a narrow range of perturbation after 20 s. This suggests that the UAV has reached the landing conditions at this point in time.

The results in Figure 9a show that both the DQN-trained PID gain-tuned controller and the PID conventional controller achieve convergence with the desired velocity within 20 s. However, as shown in Figure 9e, the DQN-trained controller has much less variation in altitude and velocity than the conventional, which suggests that the DQN-trained controller has far superior overshoot for landing trajectory tracking. Nevertheless, both simulated scenarios exhibit a considerable degree of overshooting during the initial phase. This is due to the fact that the controller is unable to accurately capture the desired downward slopes during the approach phase of the fixed-wing UAV. Figure 9c,d also demonstrate that the DQN-trained controllers have less overshooting of the UAV attitude and can reach convergence faster, with a more stable attitude, and are less affected by disturbances during the landing of the UAV. The final perturbation of the angle of attack can be found in Figure 9d since the simulation environment for the UAV landing presented in this paper is conducted at a low altitude, which results in unstable airflow. This causes the aerodynamic forces of the aircraft to fluctuate, leading to a change in lift and an alteration in the angle of attack. Despite the presence of this perturbation, a certain angle of attack perturbation ensures a safe landing within the parameters demonstrated in ref. [32]. Especially in Figure 9b, the amount of overshooting of the angle of attack perturbation during DQN training is almost half of that of the untrained controllers, which can well guarantee the UAV attitude during the landing process, and avoid the loss of control due to the excessive adjustment of the angle of attack that leads to stalling. Therefore, the trained controller can accomplish the landing task while minimizing the overshooting of its state quantities and avoiding the saturation effect, proving that the controller model is well trained.

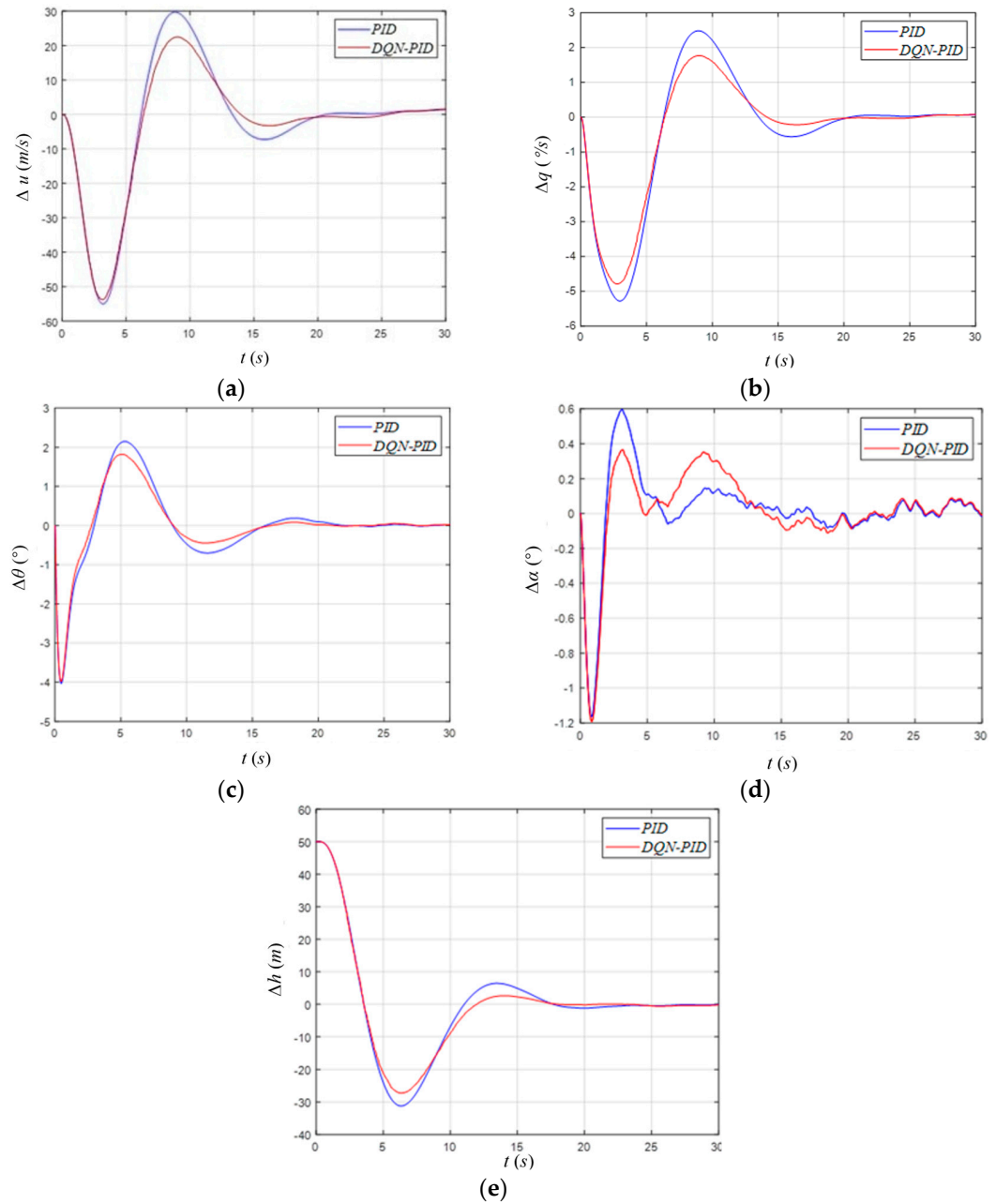


Figure 9. Control output results of fixed-wing UAV. (a) Velocity results, (b) Pitch angle velocity results, (c) Pitch angle results, (d) Angle of attack results, (e) Altitude results.

In parallel, the DQN-trained two-dimensional landing trajectory is plotted in accordance with the horizontal and vertical distances of the UAV relative to the origin, as illustrated in Figure 10. It can be posited that the optimal landing trajectory is a straight line. In other words, it represents the optimal glide slope during the landing of a fixed-wing aircraft, which exists during the landing process and on which the aircraft can descend in a more stable and safer manner in preparation for landing. It is an ideal and stable smooth curve, where the speed and pitch angle are expected to remain constant, and the landing control is smaller and more stable. As illustrated in Figure 10, the trained UAV consistently adheres to this trajectory throughout the landing process, thereby guaranteeing the successful completion of the landing task.

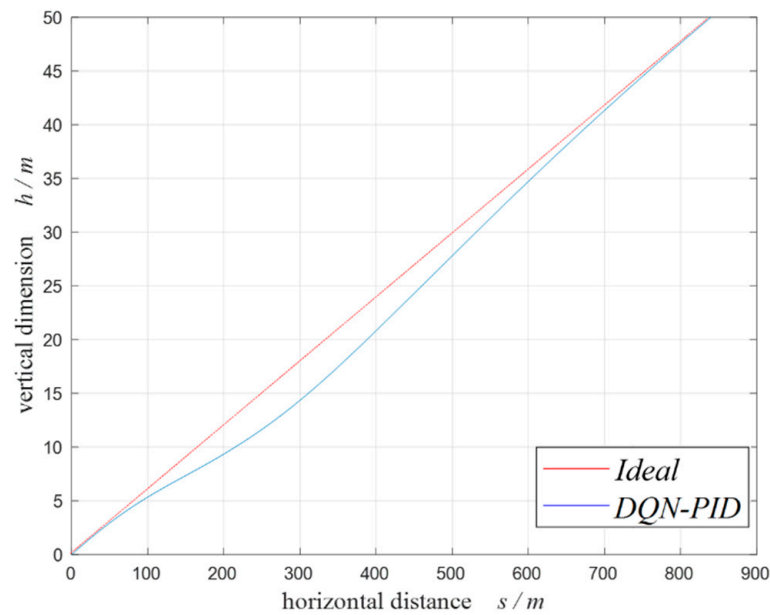


Figure 10. Working condition simulation results of uniform Controlling operation input.

Upon comparing Figures 8–10, it becomes evident that the UAV under PID and DQN-PID controllers can reach above the platform within the simulation time and satisfy the given landing conditions. However, the conventional PID control flight state ups and downs attitude fluctuation is large, and the amount of overshooting is significantly larger than that of the PID controller trained with the DQN algorithm. In terms of both the time required for the controller to achieve convergence and the amount of overshoot, the DQN-PID controller demonstrates a notable improvement relative to the conventional landing controller, exhibiting enhanced anti-interference performance.

5.4. Simulation with Real Atmospheric External Disturbance

This Section is primarily concerned with the analysis of the landing conditions of the UAV and the evaluation of the anti-disturbance performance of the control system in the context of real atmospheric external disturbances during the landing process of the fixed-wing UAV. In a real-world setting, the impact of external disturbances, such as the free atmosphere and noise, cannot be overlooked [33]. This Section begins by modeling the atmospheric environment with a disturbance that interferes with the UAV, and then simulates the controller in the presence of the external perturbation. The experimental results are presented subsequently.

5.4.1. Atmospheric Environment and Disturbance Model

The UAV landing process simulation is mainly disturbed by the atmosphere turbulence component, especially in the longitudinal direction. However, there are multiple sources of atmospheric and weak turbulence in the real environment, and its flow is characterized by nonlinearity, non-stationarity, randomness, etc. In order to better generate turbulence data, the problem that needs to be solved is the generation of random signals as input. In this paper, we use computer-generated “pseudo-random numbers” to simulate the real input signals required in the numerical process of atmospheric turbulence, whose characteristics only approximately correspond to white noise. In ref. [23], the paper modeled turbulence with the coupling of state signals and several types of white noise, which are generated by computer as well. The method of generating motion by simulation is illustrated in Figure 11, and the signal sequence of white noise generated in this paper and its power spectral density analysis results are shown in Figure 12.

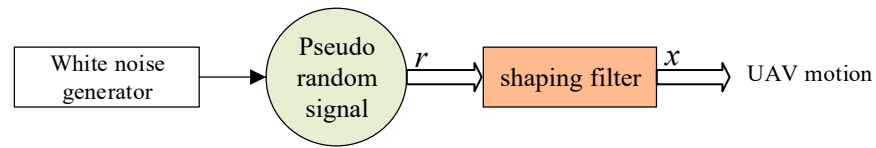


Figure 11. Motion generating method.

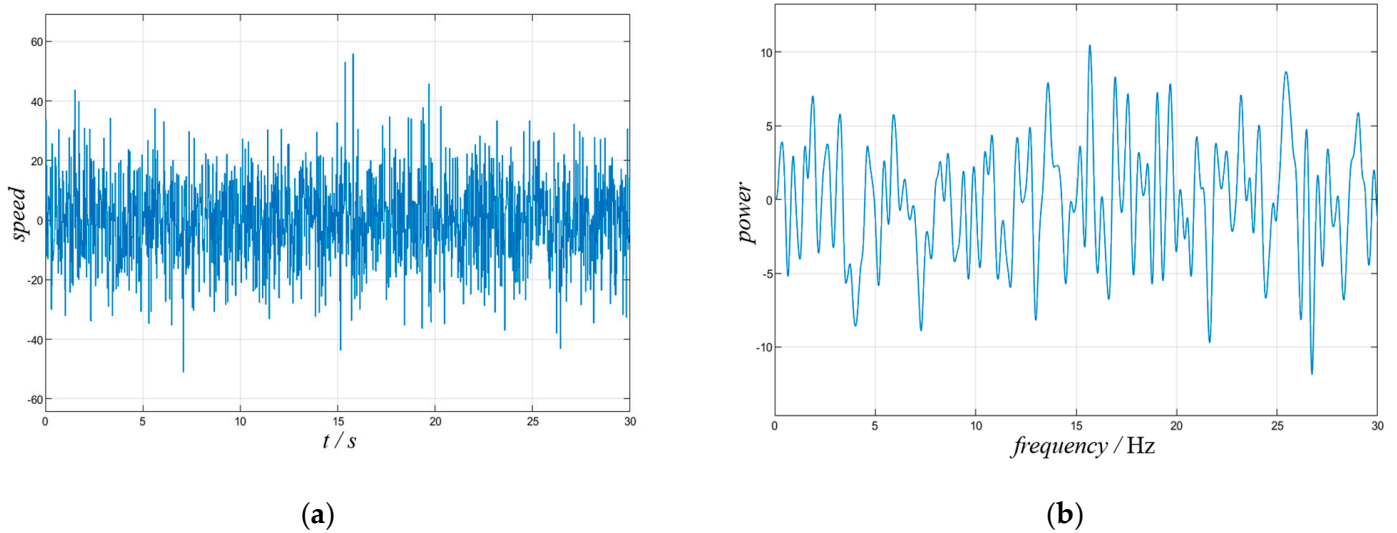


Figure 12. White Noise generation: (a) White noise sequence; (b) White noise power spectral density analysis.

Considering the white noise reflecting the randomness of the motion and more practical application, the UAV motion is constructed based on the power spectral density functions. In Equations (25) and (26), the translational and rotational disturbances shaping filter is respectively represented. The shaping filter used can be expressed using the following two transfer functions: the disturbance velocity of the free atmospheric turbulence component generated by the UAV can be approximated as the coupling of the multiplicative white noise and the system signal, that is, the turbulence component. Their transfer functions can be expressed as respectively:

$$G_T(s) = \frac{b_3s^2 + b_2s + b_1}{s^4 + a_4s^3 + a_3s^2 + a_2s + a_1} \tag{25}$$

$$G_A(s) = \frac{d_3s^2 + d_2s + d_1}{s^4 + c_4s^3 + c_3s^2 + c_2s + c_1} \tag{26}$$

The detailed forms and parameters of the shaping filter functions affecting the longitudinal motion of the UAV are given in Table 4.

Table 4. Transfer Function between the UAV Motion and White Noise.

ϕ	$G_\phi = \frac{0.334059s^2}{s^4 + 0.604s^3 + 0.79658s^2 + 0.206272s + 0.123907}$
θ	$G_{\theta_c} = \frac{0.238368s^2}{s^4 + 0.2088s^3 + 0.397556s^2 + 0.038628s + 0.034225}$
ψ	$G_\psi = \frac{0.0058s^2 + 0.1520s + 1}{s^4 + 1.2s^3 + 1.98s^2 + 0.9720s + 0.6561}$

The modeling method for the atmospheric disturbances faced by fixed-wing UAVs during landing consists of two parts. The free atmospheric turbulence component $[u_1, v_1, w_1]^T$

is anisotropic in different directions combined with the model established in ref. [24]. The calculation results of this formula will be substituted into the model for the calculation of aircraft airspeed, angle of attack, and sideslip angle, and its power spectral density functions are described by Equation (27):

$$\begin{cases} \Phi_u(\Omega) = \frac{5.66}{1 + (30.48\Omega)^2} \\ \Phi_v(\Omega) = \frac{26.59 \cdot [1 + (121.92\Omega)^2]}{[1 + (304.8\Omega)^2] \cdot [1 + (40.64\Omega)^2]} \\ \Phi_w(\Omega) = \frac{2}{1 + (30.48\Omega)^2} \end{cases} \quad (27)$$

where u , v , and w represent the three components of the atmospheric turbulence on UAV under the body shafting in the landing field, respectively.

Random wake component $[u_2, v_2, w_2]^T$, which can be generated by the white noise through the forming filter, and the constant sea surface wind in the horizontal plane is taken into account. Then, the final wind field can be expressed as follows:

$$\begin{cases} u_g = u_1 + u_2 \\ v_g = v_1 + v_2 \\ w_g = w_1 + w_2 \end{cases} \quad (28)$$

5.4.2. Controller Landing Simulation Test in the Condition of External Perturbation

In order to more accurately simulate the landing control problem of UAVs in real environments, a certain amount of external interference noise disturbance is added to the UAV during the landing process in accordance with the aforementioned setup, as well as external interference wind disturbance. Concurrently, the simulation conditions for the UAV are set to the same parameters as outlined in Section 5.3. The initial position of the UAV and the landing area, along with the initial value of the velocity state relationship and other specific training simulation parameter settings, is presented in Table 3. Figure 13 illustrates the simulation results of the controller flight state condition following conventional PID and DQN-PID training in the presence of simulated external atmospheric perturbation. The landing effect and flight attitude of the UAV are observed, and the flight state vector is recorded, at the same time, the amount of deviation perturbation from the ideal glide slope is judged, and it is observed whether it reaches the convergence condition.

As illustrated in Figure 13a–e, the flight state perturbation of the UAV can be seen to converge to a smaller range under both PID and DQN-PID controllers. This demonstrates that at this point in time, the UAV has achieved the landing conditions. Furthermore, the amount of overshooting of the DQN-trained controller is significantly smaller than that of the untrained one, and the state curves are more smooth. Additionally, the speed of convergence has been significantly shortened. The state curve is more linear and the convergence speed is significantly reduced. The simulation environment was designed to incorporate specific external interference scenarios to assess the stability of the landing control system in the presence of such disturbances. This approach was taken to ascertain the resilience of the controller in the face of post-landing interference, thereby validating its robustness.

Figure 13a depicts the amount of perturbation of the UAV velocity during the landing process, due to the presence of external perturbation, u does not finally reach the convergence to zero but conforms to the previously set error range. A comparison of the two controllers reveals that the amount of overshooting is markedly reduced in the case of the DQN-based controller. Furthermore, the speed of convergence is observed to be approximately 5 s slower, which serves to substantiate the efficacy of the controller in regulating the UAV landing conditions in the presence of external disturbances. As shown in Figure 13e, the altitude of the DQN-trained controller satisfies the UAV landing convergence condition almost 5 s earlier, which indicates that the altitude and velocity variations after neural

network training are much smaller than those of the conventional controller, and it has superior overshooting performance in landing trajectory tracking. In Figure 13b, the angle-of-attack perturbation overshoot under the DQN-trained controller model almost converges within the set error range after 12 s, which proves that the trained network can guarantee the UAV attitude during landing very well compared with the conventional PID controller. Figure 13c,d also show that the DQN-trained controller has less overshoot on the UAV attitude, reaches convergence faster, has a more stable attitude, and is less affected by disturbances during UAV landing. Therefore, after the introduction of external disturbances in the model, the DQN-trained PID-DQN controller is able to minimize the overshoot of the state quantities and accelerate the convergence speed of the UAV in trajectory and attitude while completing the landing task, proving that the controller model training is effective while having certain anti-disturbance performance and robustness.

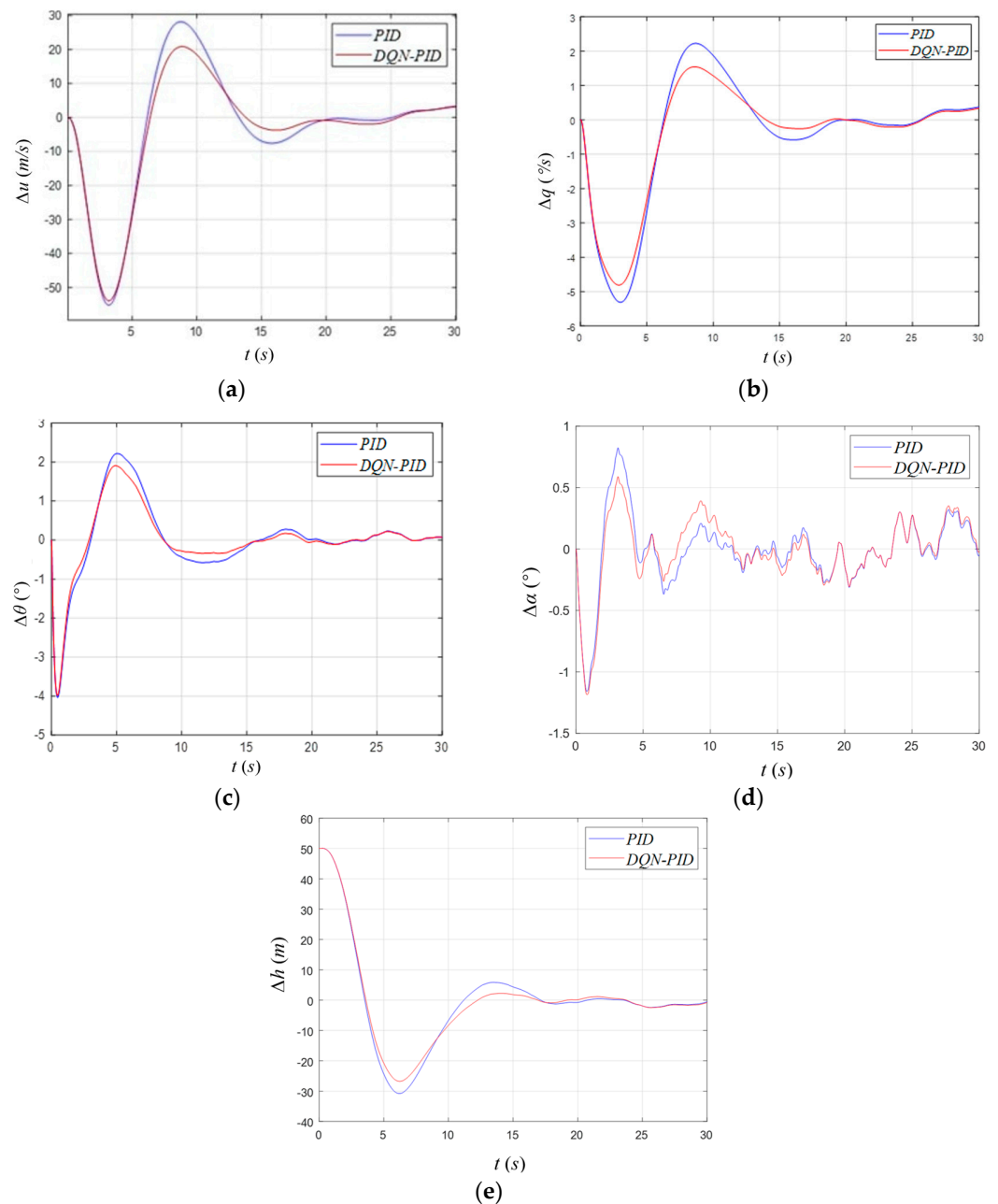


Figure 13. Control output results of fixed-wing UAV with external perturbation. (a) Velocity results, (b) Pitch angle velocity results, (c) Pitch angle results, (d) Angle of attack results, (e) Altitude results.

5.4.3. Experimental Results in an Open-Source Simulation Environment Gazebo

Considering that at present it is difficult to carry out fixed-wing UAV landing control experiments in real environments due to the limitations of experimental conditions and environments, the amount of effective result observations is obtained.

Accordingly, the efficacy of the proposed algorithm is validated through the utilization of a recognized open-source simulation environment, namely, the Gazebo simulation. The Gazebo 3D-environment scenario is constructed in accordance with the system in order to facilitate the performance of UAV autonomous landing simulation experiments. The results of the simulation demonstrate that the enhanced DQN-PID algorithm is capable of successfully completing the landing experiment. The consequences of modeling the fixed-wing UAV, which is the focus of the automatic landing controller, within the Gazebo simulation environment are illustrated in Figure 14.

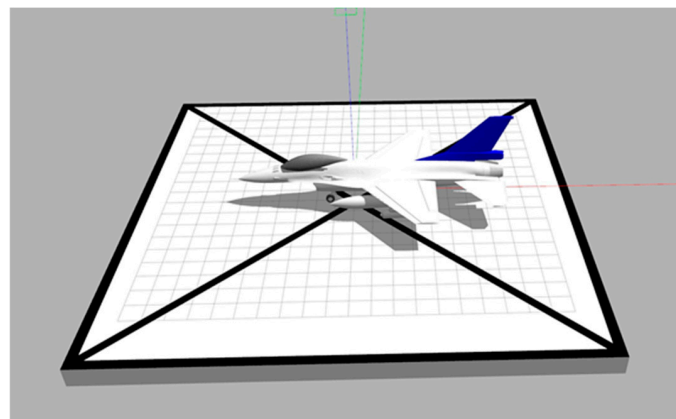


Figure 14. Fixed-wing UAV landing environment modeling.

The DQN-PID controller is employed to facilitate the autonomous landing of a fixed-wing UAV on a ground-fixed landing area within the context of a Gazebo simulation environment. The landing platform is illustrated in Figure 14. The experimental demonstration outcome is illustrated in Figure 15. Meanwhile, Figure 16 illustrates that the quantity of the resulting observed perturbations is largely consistent with the findings of the simulation environment. Due to Gazebo's modeling of real-world physical properties such as gravity, friction, collisions, etc., it is possible to simulate the non-negligible external perturbations that a fixed-wing UAV would encounter on the near-Earth surface, including atmospheric disturbances. The control output overshoots, and perturbations of the UAV are larger than in the previous discussion. Furthermore, convergence is achieved within the specified time frame, thereby substantiating the efficacy of the algorithm.

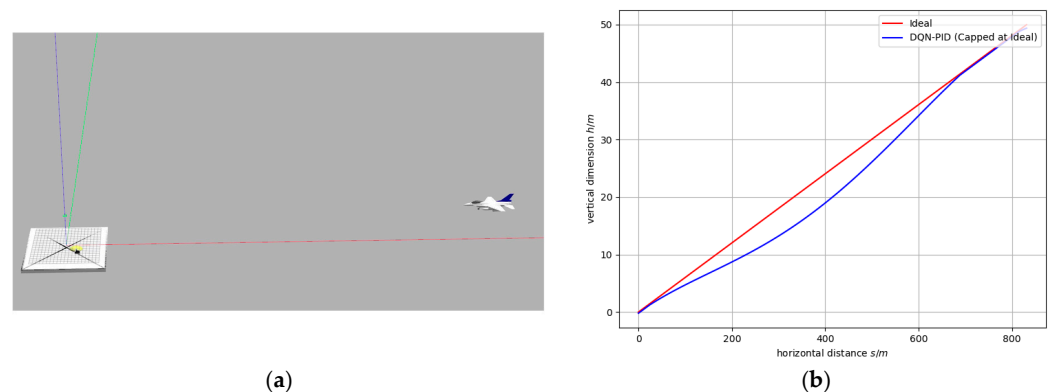


Figure 15. DQN-PID controller for fixed-wing UAV landing experiment. (a) Experimental demonstration, (b) Experimental results of uniform Controlling operation input.

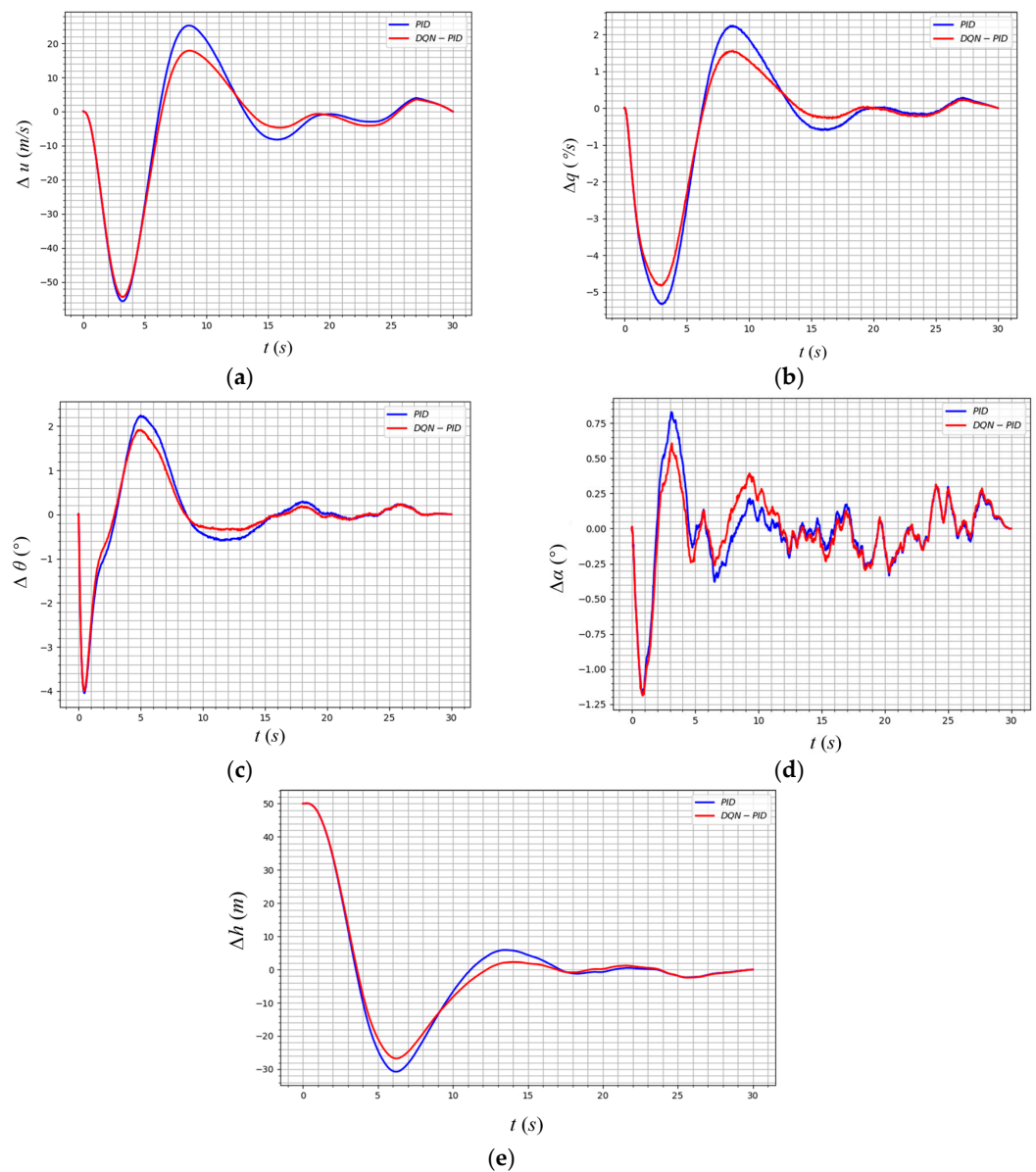


Figure 16. Control outputs results of a fixed-wing UAV in the Gazebo simulator. (a) Velocity results, (b) Pitch angle velocity results, (c) Pitch angle results, (d) Angle of attack results, (e) Altitude results.

6. Conclusions

This paper presents a DQN-PID controller based on the reinforcement learning DQN algorithm, which is designed to address the issue of conventional PID parameter tuning. The principal conclusions are as follows:

- (1) The actuator model of a UAV is established by combining the kinematics and dynamics equations of a UAV. Based on the conventional PID control combined with the stability augmentation control system (SAS), the control model is established to make the flight control more stable and realize the automatic landing control of the UAV. A control model is established based on a conventional PID control combined with a stability augmentation control system (SAS), with the objective of enhancing the stability of the flight control and enabling the automatic landing control of the UAV. A method for automatically tuning PID parameters using deep learning, based on the DQN reinforcement learning method, was devised. The landing control problem was transferred into Markov Decision Process (MDP) to analyze and establish the reward function to analyze its landing conditions. A neural network was built to calculate

the expected maximization and output the action space PID parameter values in the output layer, which realizes the adaptive tuning of PID gain parameters and the optimization of the conventional controller.

- (2) A simulation experiment model is established by combining the actuator model of a UAV with a DQN-PID landing controller. Through the experiment and analysis, comparing the working condition test of the conventional PID control and DQN-PID control, the results show that the designed controller has obvious improvement in convergence speed and overshooting amount, which improves the stability and reliability of the UAV flight control, proves that the algorithm has obtained a good learning strategy, and verifies the validity of the controller for practical application.

Moreover, in the aforementioned tests and analyses, the ground landing platform trolley under consideration is observed to be in a state of near-stationarity with respect to the ground reference system. In order to fully simulate the real landing conditions, it is possible to analyze the UAV landing conditions when the cart is in motion and to establish the UAV–cart system. For example, if the trolley is moving at a certain rate and trajectory, the UAV can capture the landing target and realize the landing control at the same time. This can be used as a reference for the analysis of the landing under more conditions of the DQN-PID control.

Author Contributions: J.L.: Conceptualization, Methodology, Software, Writing—original draft. S.X.: Methodology, Supervision, Funding acquisition. Y.W.: Funding acquisition, Formal analysis, Investigation. Z.Z.: Software, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities, grant number BLX202224 and the National Natural Science Foundation of China, grant number 12402413.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, [Shuting Xu], upon reasonable request.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

references

1. Mukherjee, A.; Misra, S.; Raghuwanshi, N.S. A survey of unmanned aerial sensing solutions in precision agriculture. *J. Netw. Comput. Appl.* **2019**, *148*, 102461. [[Crossref](#)]
2. Bhardwaj, A.; Sam, L.; Martín-Torres, F.J.; Kumar, R. UAVs as remote sensing platform in glaciology: Present applications and future prospects. *Remote Sens. Environ.* **2016**, *175*, 196–204. [[Crossref](#)]
3. Qin, T.; Zhang, G.; Yang, L.; He, Y. Research on the Endurance Optimisation of Multirotor UAVs for High-Altitude Environments. *Drones* **2023**, *7*, 469. [[Crossref](#)]
4. Asadi, K.; Suresh, A.K.; Ender, A.; Gotad, S.; Maniyar, S.; Anand, S.; Noghabaei, M.; Han, K.; Lobaton, E.; Wu, T. An integrated UGV-UAV system for construction site data collection. *Autom. Constr.* **2020**, *112*, 103068. [[Crossref](#)]
5. Gamagedara, K.; Lee, T.; Snyder, M. Delayed Kalman filter for vision-based autonomous flight in ocean environments. *Control Eng. Pract.* **2024**, *143*, 105791. [[Crossref](#)]
6. Åström, K.J.; Hägglund, T. The future of PID control. *Control Eng. Pract.* **2001**, *9*, 1163–1175. [[Crossref](#)]
7. Wu, T.; Zhou, C.; Yan, Z.; Peng, H.; Wu, L. Application of PID optimization control strategy based on particle swarm optimization (PSO) for battery charging system. *Int. J. Low-Carbon Technol.* **2020**, *15*, 528–535. [[Crossref](#)]
8. Ziquan, Y.U.; Youmin, Z.; Bin, J. PID-type fault-tolerant prescribed performance control of fixed-wing, U.A.V. *J. Syst. Eng. Electron.* **2021**, *32*, 1053–1061. [[Crossref](#)]
9. Acharya, D.S.; Mishra, S.K. A multi-agent based symbiotic organisms search algorithm for tuning fractional order PID controller. *Measurement* **2020**, *155*, 107559. [[Crossref](#)]
10. Juang, J.G.; Chien, L.H.; Lin, F. Automatic landing control system design using adaptive neural network and its hardware realization. *IEEE Syst. J.* **2011**, *5*, 266–277. [[Crossref](#)]
11. Zhao, W.; Liu, H.; Wang, X. Robust visual servoing control for quadrotors landing on a moving target. *J. Frankl. Inst.* **2021**, *358*, 2301–2319. [[Crossref](#)]
12. Zhen, Z.; Peng, M.; Xue, Y.; Jiang, S. Robust preview control and autoregressive prediction for aircraft automatic carrier landing. *IEEE Access* **2019**, *7*, 181273–181283. [[Crossref](#)]

13. Xue, Y.; Zhen, Z.; Zhang, Z.; Cao, T.; Wan, T. Automatic carrier landing for UAV based on integrated disturbance observer and fault-tolerant control. *Aircr. Eng. Aerosp. Technol.* **2023**, *95*, 1247–1256. [[Crossref](#)]
14. Kim, B.S.; Calise, A.J. Nonlinear flight control using neural networks. *J. Guid. Control Dyn.* **1997**, *20*, 26–33. [[Crossref](#)]
15. Lee, T.; Kim, Y. Nonlinear adaptive flight control using backstepping and neural networks controller. *J. Guid. Control Dyn. (JGCD)* **2001**, *24*, 675–682. [[Crossref](#)]
16. Tang, C.; Lai, Y.C. Deep reinforcement learning automatic landing control of fixed-wing aircraft using deep deterministic policy gradient. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; IEEE: Piscataway, NJ, USA, 2020.
17. Qing, Z.; Zhu, M.; Wu, Z. Adaptive neural network control for a quadrotor landing on a moving vehicle. In Proceedings of the 2018 Chinese Control And Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; IEEE: Piscataway, NJ, USA, 2018.
18. Wang, J.; Wang, T.; He, Z.; Cai, W.; Sun, C. Towards better generalization in quadrotor landing using deep reinforcement learning. *Appl. Intell.* **2023**, *53*, 6195–6213. [[Crossref](#)]
19. 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 reinforcement learning. *Nature* **2015**, *518*, 529–533. [[Crossref](#)]
20. Zhang, M.; Wu, S.; Jiao, J.; Zhang, N.; Zhang, Q. Energy- and Cost-Efficient Transmission Strategy for UAV Trajectory Tracking Control: A Deep Reinforcement Learning Approach. *IEEE Internet Things J.* **2022**, *10*, 8958–8970. [[Crossref](#)]
21. Storey, V.C.; Lukyanenko, R.; Maass, W.; Parsons, J. Explainable ai. *Commun. ACM* **2022**, *65*, 27–29. [[Crossref](#)]
22. Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; Pedreschi, D. A survey of methods for explaining black box models. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–42. [[Crossref](#)]
23. Okolo, W.; Dogan, A.; Blake, W.B. Development of an aerodynamic model for a delta-wing equivalent model II (EQ-II) aircraft. In Proceedings of the AIAA Modeling and Simulation Technologies Conference, Kissimmee, FL, USA, 5–9 January 2015; p. 0902.
24. Chen, C.; Tan, W.Q.; Qu, X.J.; Li, H.X. A fuzzy human pilot model of longitudinal control for a carrier landing task. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *54*, 453–466. [[Crossref](#)]
25. Somefun, O.A.; Akingbade, K.; Dahunsi, F. The dilemma of PID tuning. *Annu. Rev. Control* **2021**, *52*, 65–74. [[Crossref](#)]
26. Xu, L. A proportional differential control method for a time-delay system using the Taylor expansion approximation. *Appl. Math. Comput.* **2014**, *236*, 391–399. [[Crossref](#)]
27. Bouallègue, S.; Haggège, J.; Ayadi, M.; Benrejeb, M. PID-type fuzzy logic controller tuning based on particle swarm optimization. *Eng. Appl. Artif. Intell.* **2012**, *25*, 484–493. [[Crossref](#)]
28. Van Hasselt, H.; Guez, A.; Silver, D. Deep reinforcement learning with double q-learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
29. Sun, R.; Zhou, Z.; Zhu, X. Flight quality characteristics and observer-based anti-windup finite-time terminal sliding mode attitude control of aileron-free full-wing configuration UAV. *Aerosp. Sci. Technol.* **2021**, *112*, 106638. [[Crossref](#)]
30. Wu, P.; Voskuijl, M.; Veldhuis, L.L.M. An approach to estimate aircraft touchdown attitudes and control inputs. *Aerosp. Sci. Technol.* **2017**, *71*, 201–213. [[Crossref](#)]
31. Guan, Y.-J.; Li, Y.-P.; Zeng, P. Aerodynamic analysis of a logistics UAV wing with compound ducted rotor. *Aircr. Eng. Aerosp. Technol.* **2023**, *95*, 366–378. [[Crossref](#)]
32. Huang, D.; Huang, T.; Qin, N.; Li, Y.; Yang, Y. Finite-time control for a UAV system based on finite-time disturbance observer. *Aerosp. Sci. Technol.* **2022**, *129*, 107825. [[Crossref](#)]
33. Yuan, Y.; Duan, H.; Zeng, Z. Automatic Carrier Landing Control with External Disturbance and Input Constraint. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *59*, 1426–1438. [[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.