33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS 2022 SWEDEN

Using Deep Reinforcement Learning to Improve the Robustness of UAV Lateral-Directional Control

Rui Wang[1], Zhou Zhou [1, 2], Xiaoping Zhu[2], Liming Zheng[3]

[1] College of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China
[2] Science and Technology on UAV Laboratory, Northwestern Polytechnical University, Xi'an 710065, China
[3] Faculty of Aerospace Engineering, Delft University of Technology, 2629 HS, Delft, Netherlands

Abstract

For a small low-cost Unmanned Aerial Vehicle (UAV), the accurate aerodynamics and flight dynamics characteristics wouldn't be obtained easily, and the control coupling is serious, so the robustness of its flight controller must be considered carefully. In order to solve the problem, a Lateral-Directional (Lat-Dir) flight control method based on Deep Reinforcement Learning (DRL) are proposed in this paper. Firstly, based on the nominal state, three control laws are designed: classical Proportional Integral Derivative (PID) control, Linear Quadratic Gaussian (LQG) control based on modern control theory, and Deep Reinforcement Learning (DRL) control based on Twin Delayed Deep Deterministic Policy Gradient (TD3) method. In order to solve the problem of incomprehensible physical meaning of neural network in DRL, a simplified control strategy network is derived based on the inspiration of PID controller. In order to solve the problem that the reward function of DRL is difficult to be determined, the weights of the optimal quadratic function designed by LQG method are adopted, and the weights of control output considering discretization is added also. Then, the three controller are applied to nominal flight state and deviation state respectively, and the numerical flight simulation is carried out. The results show that, in the nominal state, the performance of DRL is close to the LQG and better than the PID. In the deviation state, which the lateral and directional static stable derivatives are changed artificially from stable to neutral stable, the rise time and adjustment time of the DRL change slightly, while the LQG degrades seriously and appears instable, and it is proved that the proposed DRL control method has better performance robustness.

Keywords: Unmanned Aerial Vehicle (UAV), flight control, Deep Reinforcement Learning (DRL), strategy network, reward function

1. General Introduction

The aerodynamics and flight dynamics characteristics of a small UAV are difficult to obtain accurately due to some technical and cost reasons. And the dynamics characteristics would change greatly in different flight states and are sensitive to external disturbance. Especially for the lateral-directional (Lat-Dir) flight, a typical MIMO system with serious coupling between flight and control. Because the classical PID flight control law design method is based on SISO system, it usually needs to be designed iteratively several times to obtain satisfied results. The modern control method represented by Linear Quadratic Gaussian (LQG) can provide better performance for MIMO systems in theory. However, an accurate dynamics model is necessary prior to LQG design. If the model deviates from the designed state, its performance would not be guaranteed.

Reinforcement learning is a goal oriented algorithm for strategy learning through interaction with the environment. Its basic idea is: the agent learns how to map the state to action through interaction with the environment, so as to maximize the rewards [1]. The delay of reward and trial-and-error makes reinforcement learning independent of the environment and has certain forward-looking

optimization characteristics. Therefore, reinforcement learning can be used to achieve the best adaptive control effect[2]. Reinforcement learning has been applied in multirotor UAV control[3][4], UAV trajectory tracking[5], control allocation[6], adaptive flight control[7] and so on.

Deep reinforcement learning techniques for motion control have recently taken a major qualitative step, since the successful application of Deep Q-Learning to the continuous action domain [8]. In ref. [9], a model-free approach called Deep Deterministic Policy Gradient (DDPG) was developed. Using the same learning algorithm, network architecture and hyper-parameters, the DDPG algorithm robustly solves more than 20 simulated physics tasks, including classic problems such as cartpole swing-up, dexterous manipulation, legged locomotion and car driving.

In ref. [10], a Proximal Policy Optimisation (PPO) algorithm is employed to train a sweep-wing UAV landing in three simulated environments with steady-state wind and turbulence. The performance of each model is assessed in simulation by obtaining the mean reward across a range of conditions. The flight test demonstrates that models trained with atmospheric disturbances perform better in the real world, achieving higher mean rewards than the baseline models that are trained without simulated wind.

In ref. [11], the flight test and verification of a neural network longitudinal controller for a fixed-wing UAV that is trained offline by DDPG algorithm are carried out. The flight test verification is performed utilizing a reference autopilot LQR controller and a safety monitoring algorithm. When detected that the predicted state of the aircraft is propagated to unsafe zone by neural network, it will switch from neural network to LQR controller automatically. The switching logic uses formal verification method and reachability analysis to expand the known safety zone, so as to extend the operation time of the neural network controller.

In ref. [12], a versatile Gazebo-based reinforcement learning framework has been designed and validated with a continuous UAV landing task. The UAV landing maneuver on a moving platform has been solved by means of the novel Deep Deterministic Policy Gradients (DDPG) algorithm. Several experiments have been performed in a wide variety of conditions for both simulated and real flights, demonstrating the generality of the approach.

Ref. [13] combines deep reinforcement learning (DRL) with meta-learning and proposes a novel approach, named meta twin delayed deep deterministic policy gradient (Meta-TD3), to realize the control of unmanned aerial vehicle (UAV), allowing a UAV to quickly track a target in an environment where the motion of a target is uncertain. Compared with the deep deterministic policy gradient (DDPG) and twin delayed deep deterministic policy gradient (TD3) algorithms, the Meta-TD3 algorithm has achieved a great improvement in terms of both convergence value and convergence rate. In UAV target tracking problem, Meta-TD3 only requires a few steps to train，which enables a UAV to adapt quickly to a new target movement mode more and maintain a better tracking effectiveness.

In ref. [14], a novel UAV autonomous tracking and landing approach based on a deep reinforcement learning strategy is presented in this paper, with the aim of dealing with the UAV motion control problem in an unpredictable and harsh environment. Instead of building a prior model and inferring the landing actions based on heuristic rules, a model-free method based on a partially observable Markov decision process (POMDP) is proposed. In the POMDP model, the UAV automatically learns the landing maneuver by an end-to-end neural network, which combines the Deep Deterministic Policy Gradients (DDPG) algorithm and heuristic rules. A Modular Open Robots Simulation Engine (MORSE)-based reinforcement learning framework is designed and validated with a continuous UAV tracking and landing task on a randomly moving platform in high sensor noise and intermittent measurements. The simulation results show that when the moving platform is moving in different trajectories, the average landing success rate of the proposed algorithm is about 10% higher than that of the PID method. As an indirect result, a state-of-the-art deep reinforcement learning-based UAV control method is validated, where the UAV can learn the optimal strategy of a continuously autonomous landing and perform properly in a simulation environment.

Ref.[15] infers that, Artificial Intelligence (AI) is expected to revolutionize all areas of space operations in the coming years. The work presents a novel framework that uses the highly researched artificial intelligence paradigm, reinforcement learning, to perform online learning. The

spacecraft attitude control problem is used as a benchmark, with experimental results for using reinforcement learning to train neural network spacecraft attitude controllers. Additionally, experimental results in a simulation environment are also shown to compare and contrast two state-of-the-art single-agent continuous control reinforcement learning algorithms, PPO and TD3, to motivate their use in the online learning scenario. It is shown that the off-policy algorithm of TD3 produces a more desirable controller than PPO in the formulation given, likely due to its explicit exploration and sample efficiency.

Although artificial intelligent control methods represented by Deep Reinforcement Learning (DRL) are booming, but there are still some problems for engineering applications, such as difficult to determine the structure of neural network, high computational power requirements of hardware, poor real-time performance and so on.

This paper presents an algorithm of Lat-Dir flight control law based on improved DRL, which is easy to be realized in engineering. In this algorithm, the reward function is designed inspired by LQG control method, and the structure of neural network is inspired and simplified by PID control method. Based on the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, deep reinforcement learning training is carried out to obtain the optimal control law matrix.

Taking a small UAV as the example, the roll angle control law is designed by using PID, LQG and the improved TD3 DRL algorithm respectively, and numerical flight simulations are carried out in the nominal state and deviation state respectively. In the deviation state, the lateral and directional static stable derivatives are changed artificially to neutral stable, the lateral and directional damping derivatives are doubled, and the control efficiency of aileron and rudder are reduced by 50%. Since the deviations of aerodynamics parameters during flight is uncertain, the same control law derived from nominal state is adopted for the deviation state.

2. Lat-Dir flight dynamics model of an example UAV

2.1 General Lat-Dir flight dynamics model

The stat- space form of the Lat-Dir flight dynamics model of a conventional UAV is:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{F}\boldsymbol{w} \tag{1}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{H}\boldsymbol{v} \tag{2}$$

Where, ***A*** and ***B*** are the system matrix and control matrix respectively. Their specific expressions can be seen in ref. [16].

The state variables matrix $\boldsymbol{x}$ is:

$$\boldsymbol{x} = [\beta, p, r, \phi]^{\mathrm{T}} \tag{3}$$

Where, β, p, r, ϕ are sideslip angle, roll rate, yaw rate and roll angle respectively.

In order to reduce the cost and gross weight, the angle of attack and sideslip angle sensor are not available for a conventional low-cost small UAV, and only IMU (inertial measurement unit) is applied to measure the angular velocity and attitude angle. Therefore, the ordinary observation variables matrix $\boldsymbol{y}$ of a small low-cost UAV is:

$$\boldsymbol{y} = [p, r, \phi]^{\mathrm{T}} \tag{4}$$

So, the output matrix ***C*** is:

$$\boldsymbol{C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$\boldsymbol{u}$ is the control variables matrix, which consist of aileron δ_a and rudder δ_r for a conventional UAV, namely:

$$\boldsymbol{u} = [\delta_a, \delta_r]^{\mathrm{T}} \tag{6}$$

$\boldsymbol{w}$ and $\boldsymbol{v}$ are the process noise and measurement noise respectively, which are independent

Gaussian white noise respectively:

$$\boldsymbol{w} \sim \boldsymbol{N}(0,1) \tag{7}$$

$$\boldsymbol{v} \sim \boldsymbol{N}(0,1) \tag{8}$$

In order to evaluate the robustness of the controller in the design process, the influence of uncertainty should be considered. According to the characteristics of uncertainty error, multiplicative perturbation is applied to describe the $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices in Eq. (1) in the following form:

$$\boldsymbol{A} = \boldsymbol{A}_0 * \Delta_{\boldsymbol{A}} \tag{9}$$

$$\boldsymbol{B} = \boldsymbol{B}_0 * \Delta_{\boldsymbol{B}} \tag{10}$$

Where: the subscript "0" represents the nominal state. Δ_A and Δ_B represents the amplitude of the perturbation. The operation symbol "*" represents the multiplication of the elements at the corresponding position of the left and right matrix.

2.2 Characteristics of the example UAV

Taking a small conventional layout UAV as an example, its outline is shown in the figure below:

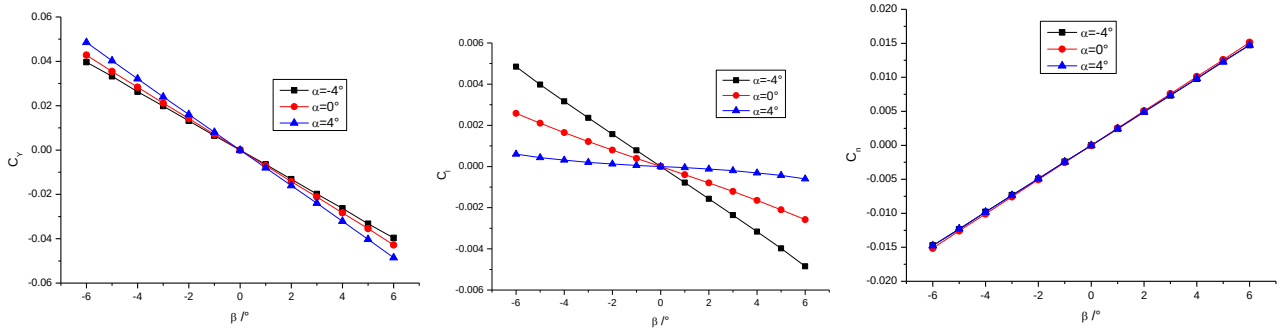
Figure 1 – The outline of the example UAV

Its main parameters are shown in the table below:

Table 1 - Main parameters of the example UAV

Parameter	Value	Parameter	Value
Span / m	1.6	Gross mass / kg	2.92
Chord / m	0.208		
Length / m	1.5	Wing load / (kg/m^2)	8.8
Wing Area / m^2	0.332	Aspect ratio	7.71
I_{xx} / (kg.m^2)	0.119	I_{zz} / (kg.m^2)	0.423
I_{yy} / (kg.m^2)	0.311	I_{xz} / (kg.m^2)	-0.015

The Lat-Dir aerodynamics characteristics are shown as Table 2 and Figure 2:



(a) coefficient of lateral force (b) coefficient of rolling moment (c) coefficient of yawing moment

Figure 2 - The static Lat-Dir aerodynamics characteristics of the example UAV

Table 2 - Main Lat-Dir aerodynamics parameters of the example UAV

Parameter	Value	Parameter	Value
$C_{Y\beta}$ / (1/rad)	-0.4407	$C_{Y\delta a}$	-0.0412
$C_{l\beta}$ / (1/rad)	-0.0169	$C_{l\delta a}$	-0.2695
$C_{n\beta}$ / (1/rad)	0.1427	$C_{n\delta a}$	-0.0043
C_{Yp}	-0.0747	$C_{Y\delta r}$	0.0881
C_{lp}	-0.5130	$C_{l\delta r}$	-0.0009
C_{np}	-0.0985	$C_{n\delta r}$	-0.0412
C_{Yr}	0.3843		
C_{lr}	0.0740		
C_{nr}	-0.1950		

According to the above parameters, the state-space matrices A, B and C of the example UAV in cruise flight state are obtained as follows:

$$\boldsymbol{A}=\begin{bmatrix} -0.4822 & 0.0307 & -0.9786 & 0.6233 \\ -29.8780 & -35.0892 & 5.5870 & 0 \\ 55.2576 & -0.6286 & -3.9710 & 0 \\ 0 & 1 & 0.0349 & 0 \end{bmatrix} \tag{11}$$

$$\boldsymbol{B}=\begin{bmatrix} -0.0226 & 0.0482 \\ -10.8249 & -0.0348 \\ -0.1726 & -1.4383 \\ 0 & 0 \end{bmatrix} \tag{12}$$

$$\boldsymbol{C}=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

The deviation of aerodynamic characteristics caused by flight state change, inaccurate results of calculation and other reasons can be expressed by the perturbation matrix Δ_A and Δ_B in the state space equation. Their values are:

$$\Delta_{\boldsymbol{A}}=\begin{bmatrix} \mathrm{N}_{C_{Y\beta}} & 1 & 1 & 1 \\ \mathrm{N}_{C_{l\beta}} & \mathrm{N}_{C_{lp}} & \mathrm{N}_{C_{lr}} & 1 \\ \mathrm{N}_{C_{n\beta}} & \mathrm{N}_{C_{np}} & \mathrm{N}_{C_{nr}} & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{14}$$

$$\Delta_{\boldsymbol{B}}=\begin{bmatrix} 1 & 1 \\ \mathrm{N}_{\delta_a} & \mathrm{N}_{\delta_a} \\ \mathrm{N}_{\delta_r} & \mathrm{N}_{\delta_r} \\ 1 & 1 \end{bmatrix} \tag{15}$$

It indicates that the UAV runs in nominal state if the elements in Δ_A and Δ_B are all equal to 1. Considering the aerodynamics parameters deviation caused by different flight states and the imprecision of CFD results, the coefficient range of the disturbance matrix of the state equation is shown as follows:

Table 3 - Main aerodynamics parameters deviation range of the example UAV

Parameter	Value	Parameter	Value
$N_{C_{Y\beta}}$	[0.2, 2]	$N_{C_{lr}}$	[0.2, 2]
$N_{C_{l\beta}}$	[-0.5, 3]	$N_{C_{nr}}$	[0.2, 2]

$N_{C_{n\beta}}$	[0, 2]	$N_{\delta a}$	[0.5, 2]
$N_{C_{lp}}$	[0.2, 2]	$N_{\delta r}$	[0.5, 2]
$N_{C_{np}}$	[0.2, 2]		

The process noise matrix ***F*** is as follows:

$$\boldsymbol{F} = [0.05, 0.05, 0.05, 0]^{\mathrm{T}} \tag{16}$$

According to the error characteristics of the sensors, the measurement noise matrix ***H*** is obtained as follows:

$$\boldsymbol{H} = [0.001, 0.001, 0.003]^{\mathrm{T}} \tag{17}$$

3. Control law design

In order to fully study the performance of intelligent control method, especially in unconventional control conditions with drastic parameters and state change, this section takes the roll angle control of UAV as an example, and designs the control law by using the classical PID method, the modern LQG method and the improved TD3 DRL method respectively for comparison.

3.1 Classic PID method

By using the cascade PID control method, the control law of the roll angle control loop of the UAV can be designed as follows:

$$p_{\mathrm{c}} = K_{\phi}(\phi_{\mathrm{c}} - \phi) \tag{18}$$

$$r_{\mathrm{c}} = \frac{\mathrm{g}}{V}\tan\phi_{\mathrm{c}} \tag{19}$$

$$\delta_{\mathrm{a}} = (K_{\mathrm{p}} + \frac{1}{s}K_{\mathrm{pI}})(p - p_{\mathrm{c}}) \tag{20}$$

$$\delta_{\mathrm{r}} = K_{\mathrm{r}}(r - r_{\mathrm{c}}) \tag{21}$$

The structure diagram of the control law is：

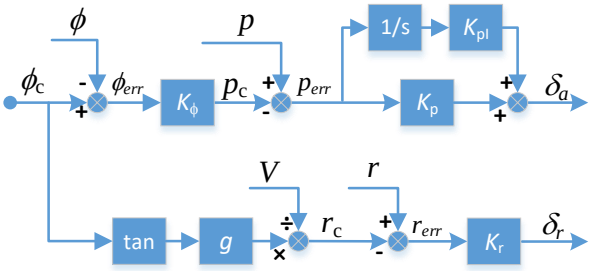


Figure 3 – Structure diagram of classical PID controller

Where, K_{ϕ}, K_{p}, K_{pI} and K_{r} are the gains of PID controller. r_{c} is the theoretical yaw rate corresponding to the zero sideslip angle determined by the coordinated turning condition. g is the acceleration of gravity, $g \approx 9.8\mathrm{m}/s^2$.

By applying both locus and bode diagram methods, the controller gains for the example UAV can be obtained as follows: K_{ϕ}=2, K_{p}=0.8, K_{pI}=0.4, K_{r}=0.6.

3.2 Modern LQG method

Linear Quadratic Gaussian (LQG) method is a common method in modern control theory. LQG is an improvement of linear quadratic (LQ) method, which can be used to design the optimal controller of systems with noise, thus LQG is more practical than LQ method. The typical structure of a LQG controller is shown in the figure below:

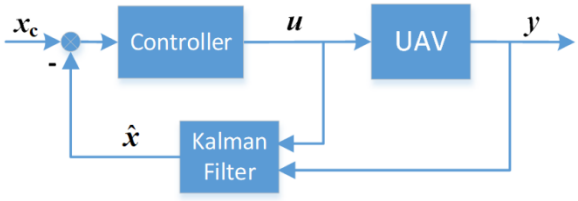


Figure 4 - Structure diagram of LQG controller

The basic idea of LQG method is: Taking a given quadratic function as the objective function, the optimal full state feedback controller is constructed according to the solution of algebraic Riccati equation, and the state of the system can be estimated by a Kalman filter.

The control law of LQG is:

$$\boldsymbol{u} = -\boldsymbol{K}_{\mathrm{LQG}}(\hat{\boldsymbol{x}} - \boldsymbol{x}_{\mathrm{c}}) \tag{22}$$

Where, $\boldsymbol{x}_c$ is the desired state variables matrix, $\hat{\boldsymbol{x}}$ is the observation of the system state $\boldsymbol{x}$, $\boldsymbol{K}_{\mathrm{LQG}}$ is the gain of the controller.

In order to determine the $\boldsymbol{K}_{\mathrm{LQG}}$, the following quadratic objective function of LQG controller shall be constructed first:

$$J = \min\{\int_0^{t_f} [(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{c}})^{\mathrm{T}} \boldsymbol{Q}(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{c}}) + \boldsymbol{u}^{\mathrm{T}} \boldsymbol{R}\boldsymbol{u}]dt\} \tag{23}$$

Where, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the weighting matrices of system state and control output respectively.

Solving the optimal quadratic objective function Eq. (23) is equivalent to solving the following algebraic Riccati equation[17]:

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P} + \boldsymbol{Q} = 0 \tag{24}$$

Where, $\boldsymbol{A}$ and $\boldsymbol{B}$ are the same system matrix and control matrix of the controlled system as shown in Eq. (1), $\boldsymbol{Q}$ is the same weighting matrix as shown in Eq. (23).

The unknown variable in Eq. (24) is $\boldsymbol{P}$. After solving it, the control gain $\boldsymbol{K}_{\mathrm{LQG}}$ can be solved according to the following formula:

$$\boldsymbol{K}_{\mathrm{LQG}} = \boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P} \tag{25}$$

All the system state variables $\boldsymbol{x}$ must be available before applying the LQG method. But as can be seen before, sideslip angle sensor is not equipped in the small low-cost UAV, that is to say, it is essential to derive the $\boldsymbol{x}$ by the $\boldsymbol{y}$ in Eq. (4). So a Kalman filter is employed, which uses the available observation variables $\boldsymbol{y}$, together with the system matrix *A* and output matrix *B* of a dynamics system, to estimate the full system state variables $\boldsymbol{x}$ optimally.

The state-space equation of a Kalman filter is as follows:

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\hat{\boldsymbol{x}}) \tag{26}$$

$$\dot{\boldsymbol{P}} = \boldsymbol{A}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{F}\boldsymbol{F}^{\mathrm{T}} - \boldsymbol{L}\boldsymbol{G}\boldsymbol{L}^{\mathrm{T}} \tag{27}$$

$$\boldsymbol{L} = \boldsymbol{P}\boldsymbol{C}^{\mathrm{T}} \tag{28}$$

Where: $\hat{\boldsymbol{x}}$ is the observed value derived from the Kalman filter on the state of the UAV. The input of the Kalman filter is the observation output $\boldsymbol{y}$ and control output $\boldsymbol{u}$ of the UAV, and the output of the Kalman filter is $\hat{\boldsymbol{x}}$. Therefore, an output feedback control can be realized by using $\hat{\boldsymbol{x}}$ to replace the $\boldsymbol{x}$ in Eq. (22).

For the roll angle control of the example UAV, it can be designed that:

$$\boldsymbol{Q} = diag(0,\ \ 0.5,\ \ 0.5,\ \ 1.2) \tag{29}$$

$$\boldsymbol{R} = diag(0.1,\ \ 0.1) \tag{30}$$

where, the "*diag*" function refers to the construction of an array into a diagonal matrix. In this case the final gain of the roll angle controller derived by LQG method is:

$$\boldsymbol{K}_{\mathrm{LQG}}=\begin{bmatrix}-0.640 & -1.638 & -0.054 & -3.745\\ 0.683 & 0.013 & -1.508 & 0.528\end{bmatrix} \tag{31}$$

3.3 Improved TD3 DRL method

Reinforcement Learning (RL) is a kind of machine learning, which is closely related to dynamic programming and optimal control theory. The basic idea of RL is to explore the optimal strategy through the interaction between agent and environment, so as to maximize the rewards. When the deep neural network is used to store the optimal strategy information, it is called Deep Reinforcement Learning (DRL). In this paper, TD3 algorithm is used for DRL. There are two key points in applying the DRL algorithm to the design of flight control. One is to find a reward function to depict the designer's intention exactly. The second is to find suitable deep actor and critic neural network structures.

3.3.1 Introduction of TD3 algorithm

The emergence of Twin Delayed Deep Deterministic policy gradient algorithm (TD3) is intend to solve the problem that DDPG (Deep Deterministic Policy Gradient) is not easy to converge because the estimated value function is too large. Just as its name implies, TD3 develops double critic network and their target network from the basic DDPG, together with actor network (or called policy network) and its target network, there are 6 deep neural networks in TD3 algorithm. In this algorithm, the two sets of independent critic networks are used to estimate the reward function at the same time, and then the smaller value is selected as the update target to solve the overestimation. The delayed update technology is also used to make the update frequency of the critic network larger than actor network, so as to obtain more stable convergence performance. The empirical playback buffer is also used to store historical data, which improves the sampling efficiency. Random sampling technology can break up the correlation between samples and stablize the learning process of the agent.

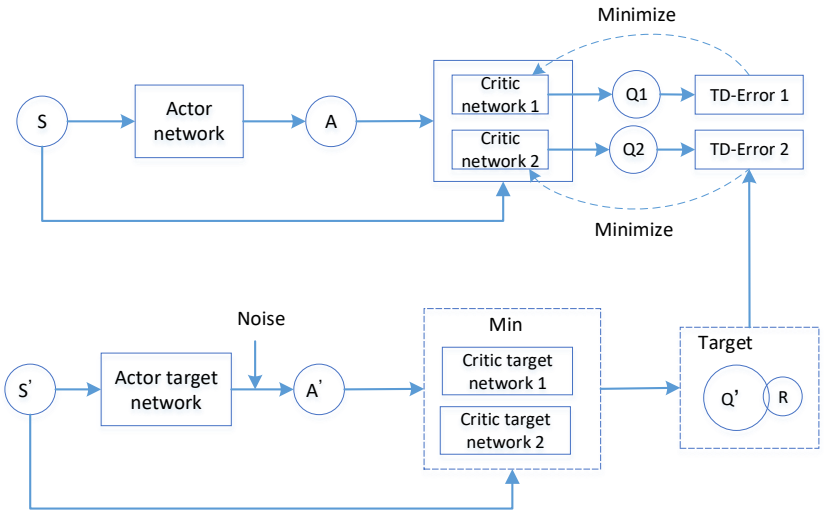


Figure 5 - Structure diagram of TD3 algorithm

3.3.2 Reward function

The reward function affects both the control objective and control performance. It abstracts the state and control variables of a dynamic system into a value, and carries out optimal control according to this value. Appropriate reward function can greatly save the design cost of intelligent control algorithm and achieve the desired control effect more efficiently. Based on the results of LQG control, this paper also selects the state variables $\boldsymbol{x}$ and control variables $\boldsymbol{u}$ of the Lat-Dir dynamic equation of UAV as optimization variables. In addition, considering the time difference characteristics of reinforcement learning, the difference of control variables $\dot{\boldsymbol{u}}$ is employed also. Thus, the reward function r of a single step i is:

$$r_i=[\boldsymbol{x}_i-\boldsymbol{x}_{\mathrm{c},i}]^{\mathrm{T}}\boldsymbol{Q}_{\mathrm{TD3}}[\boldsymbol{x}_i-\boldsymbol{x}_{\mathrm{c},i}]+\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}_i-\boldsymbol{u}_{i-1}\end{bmatrix}^{\mathrm{T}}\boldsymbol{R}_{\mathrm{TD3}}\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{u}_i-\boldsymbol{u}_{i-1}\end{bmatrix} \tag{32}$$

And the reward of an entire episode R is:

$$R = -\sum_{i=1}^{n} r_i \tag{33}$$

Where: the subscript i represent the i^{th} step in an episode. The value of $\boldsymbol{Q}_{\mathrm{TD3}}$ is equal with that of in Eq.(23) . In order to reduce the excessive control while getting greater reward, it is necessary to clap the control output together with the control rate. Therefore, in the reward function, $\boldsymbol{u}_i$ and $\boldsymbol{u}_i - \boldsymbol{u}_{i-1}$ should be evaluated at the same time. and for the roll controller of the example UAV, $\boldsymbol{R}_{\mathrm{TD3}}$ can be determined as:

$$\boldsymbol{R}_{\mathrm{TD3}} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.01 \end{bmatrix} \tag{34}$$

3.3.3 Design of Policy network

The main parameters of deep neural network include the number of network nodes, the number of layers, the type of activation function and so on. If the number of network nodes is too small, it will not be enough to describe the complex control response relationship of a UAV. Increasing the number of network nodes can obtain smoother output response, but it will consume more memory, which is not conducive to the deployment of the network on a practical flight control system. Moreover, when the number of nodes increases to a certain extent, it's gain would not increase obviously, but the training difficulty of the network would increase dramatically. Therefore, reasonable network structure is one of the key technologies of DRL design.

In order to obtain a reasonable policy network, firstly, the classical cascade PID control law in Eq. (18) (19)(20)(21)is rearranged as the following formula:

$$\delta_{\mathrm{a}} = K_{\mathrm{p}} p + K_{\mathrm{pI}} \frac{1}{s} p + K_{\mathrm{p}} K_{\phi} (\phi_{\mathrm{c}} - \phi) - \frac{1}{s} K_{\mathrm{pI}} K_{\phi} (\phi_{\mathrm{c}} - \phi) \tag{35}$$

$$\delta_{\mathrm{r}} = K_{\mathrm{r}} (r - r_{\mathrm{c}}) \tag{36}$$

Because $\frac{1}{s} p = \phi$, the above formula can be written in the following matrix form:

$$\boldsymbol{u} = \boldsymbol{K}_{\mathrm{PID}} \boldsymbol{S} \tag{37}$$

Where:

$$\boldsymbol{K}_{\mathrm{PID}} = \begin{bmatrix} K_{\mathrm{p}} & 0 & K_{\mathrm{pI}} & K_{\mathrm{p}} K_{\phi} & K_{\mathrm{pI}} K_{\phi} \\ 0 & K_{\mathrm{r}} & 0 & 0 & 0 \end{bmatrix} \tag{38}$$

$$\boldsymbol{S} = \begin{bmatrix} p \\ r_{\mathrm{c}} - r \\ \phi \\ \phi_{\mathrm{c}} - \phi \\ \frac{1}{s}(\phi_{\mathrm{c}} - \phi) \end{bmatrix} \tag{39}$$

Noticing that a common full connection layer deep neural network structure is:

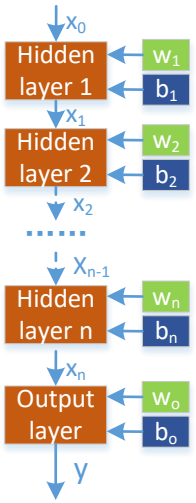
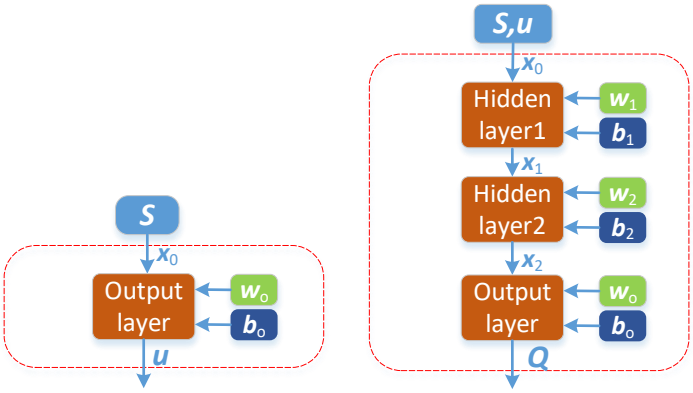


Figure 6 - Structure diagram of a common full connection layer deep neural network

If the input of the network layer is $\boldsymbol{S}$, the output is $\boldsymbol{u}$, and the number of network layers is 1, at this time, the structure of the policy network can be depicted as Figure 7 (a):



(a) policy network (b) Critic network

Figure 7 - Structure diagram of the deep neuro network

Furthermore, if the offset parameter $\boldsymbol{b}_o = 0$, and the activation function is 1, the formula of the policy network will have a form consistent with Eq. (37)(38)(39). The unique network parameter matrix $\boldsymbol{W}_o$ is similar to $\boldsymbol{K}_{PID}$ in Eq. (37)(38), but all the 10 elements are adjustable.

There are three advantages in the policy network: (1) Taking the classical PID control law as a reference, and containing more adjustable parameters, the control effect would be better than the classical PID controller. (2) The network structure is very simple, and the training efficiency of reinforcement learning is higher. When deploying the network, the demand for the computing power of the flight controller is also reduced. (3) All observation measurements (all related to p, r, ϕ) can be measured by common low-cost sensors, which is conducive to engineering application.

3.3.4 Design of Critic network

The structure of the critic network can be seen in Figure 7 (b).

The critic network has three layers, and the number of nodes in each layer is 128. Its input is the measurement of the flight state $\boldsymbol{S}$ together with the control output $\boldsymbol{u}$ of the UAV, and its output is the Q value of the network. In order to obtain the gradient of the critic more easily, the activation function of both the hidden layer and the output layer are all “relu” function. Therefore, the formula of the network is:

$$\boldsymbol{x}_0 = [\boldsymbol{S}^{\mathrm{T}}, \boldsymbol{u}^{\mathrm{T}}] \tag{40}$$

$$\boldsymbol{x}_1 = \mathrm{relu}(\boldsymbol{w}_1 \boldsymbol{x}_0 + \boldsymbol{b}_1) \tag{41}$$

$$\boldsymbol{x}_2 = \mathrm{relu}(\boldsymbol{w}_2 \boldsymbol{x}_1 + \boldsymbol{b}_2) \tag{42}$$

$$Q = \mathrm{relu}(\boldsymbol{w}_o \boldsymbol{x}_2 + \boldsymbol{b}_o) \tag{43}$$

As mentioned above, in the TD3 algorithm the two *Q*-value networks and their target networks adopt the same network structure described in Eq. (40)(41)(42)(43), but their update process is independent.

3.3.5 Training of reinforcement learning

The improved TD3 algorithm proposed in this paper is used for the reinforcement learning training of the roll angle control loop of the example UAV.

The super parameters of TD3 algorithm are set as follows:

Table 4 - Super parameters of TD3 algorithm

Parameter	Value
TRAIN_EPISODES	500
TEST_EPISODES	1
MAX_STEPS	500
BATCH_SIZE	64
EXPLORE_STEPS	10000
HIDDEN_DIM	128
UPDATE_ITR	3
Q_LR	2.0e-4
POLICY_LR	1e-4
POLICY_TARGET_UPDATE_INTERV	3
EXPLORE_NOISE_SCALE	0.05
EVAL_NOISE_SCALE	0.05
REWARD_SCALE	1
REPLAY_BUFFER_SIZE	5e5
GAMMA	0.995
SOFT_TAU	1e-3

The obtained learning curve is shown in Figure 8. Among them, “mean (1)” refers to the reward of a single episode, “mean (10)” and “std (10)” refer to the average reward and its standard deviation of 10 adjacent episodes.

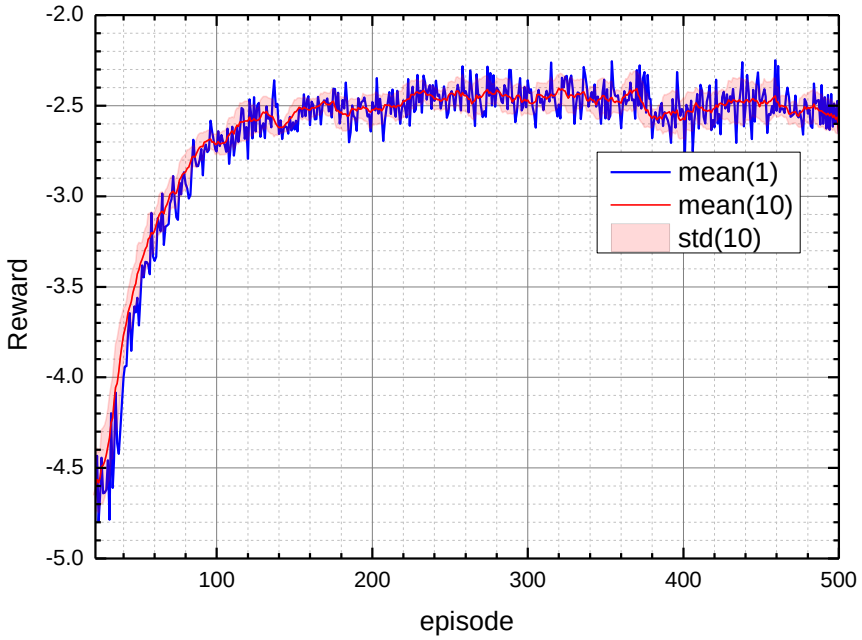


Figure 8 - Learning curve

It can be seen from Figure 8 that, the best reward appears in the 459th episode, its value is -2.249. In this case, the trained optimal policy network parameter $\boldsymbol{W}_o$ is:

$$\boldsymbol{W}_o = \begin{bmatrix} 1.415 & -0.437 & 0.053 & -3.324 & -0.112 \\ 0.143 & -0.234 & -0.091 & -0.082 & 0.018 \end{bmatrix} \tag{44}$$

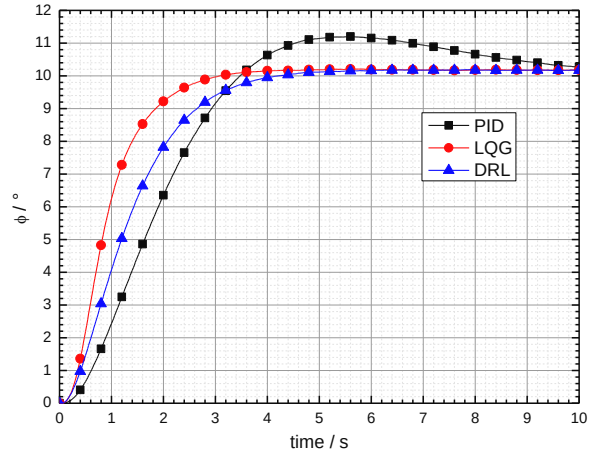
4. Numerical flight simulation

The designed PID, LQG and DRL controllers above are respectively applied to the example UAV to conduct the flight simulation for comparison, which the roll angle is controlled from initial 0 to desired 10 degrees. The corresponding roll angle response history and so on of the nominal state and deviation state are shown in the following sections.

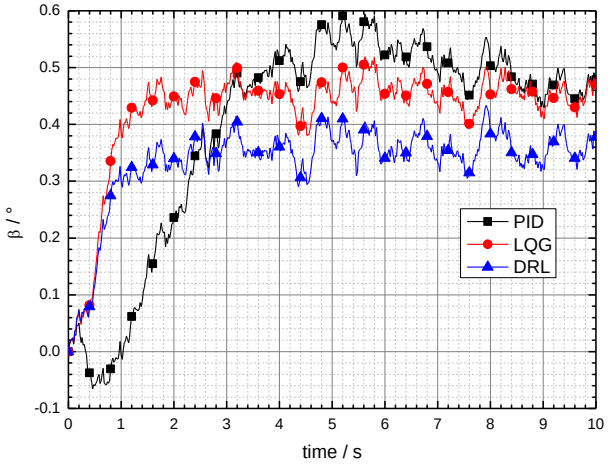
4.1 Results of flight simulation

4.1.1 nominal state

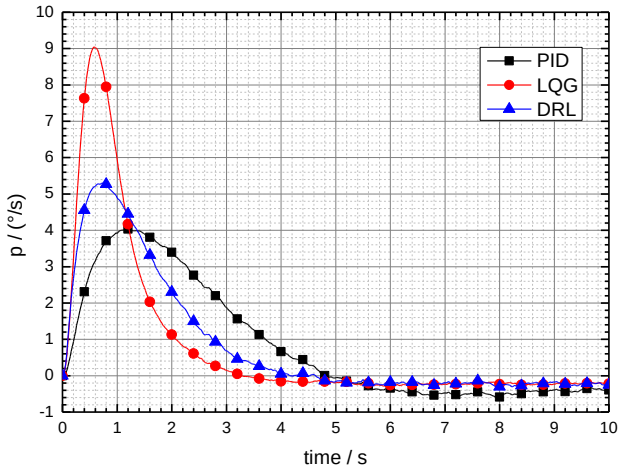
The numerical flight simulation results of the three controllers of the example UAV in nominal state are shown in Figure 9. The flight dynamics model of the UAV is shown in Section 2, and note that the process noise and measurement noise are all considered in the simulation.



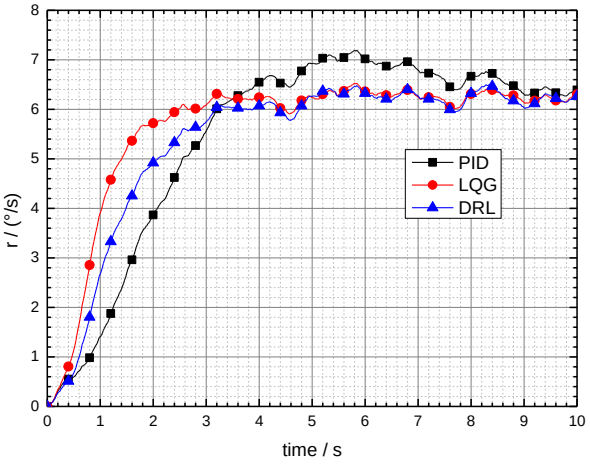
(a) roll angle



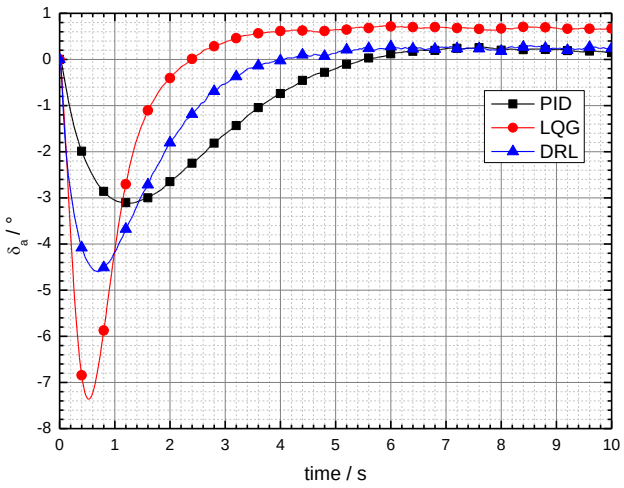
(b) side slip angle



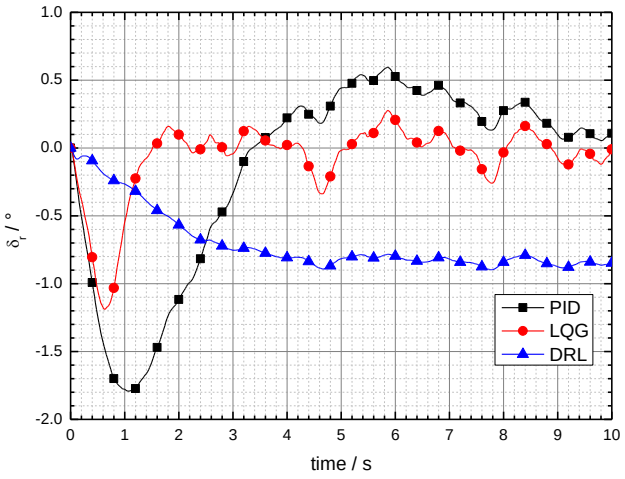
(c) roll rate angle



(d) yaw rate angle



(e) aileron deflect angle



(f) rudder deflect angle

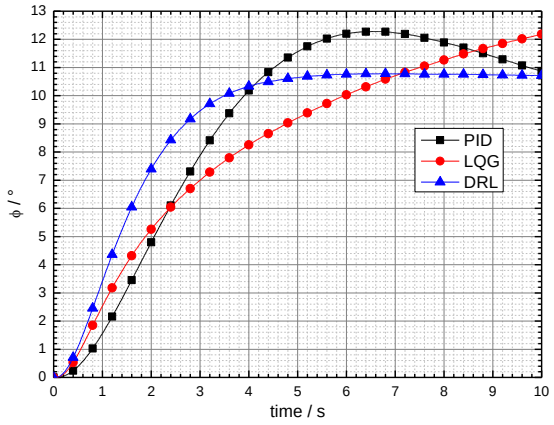
Figure 9 - Roll control responses with different control methods under nominal conditions

4.1.2 deviation state

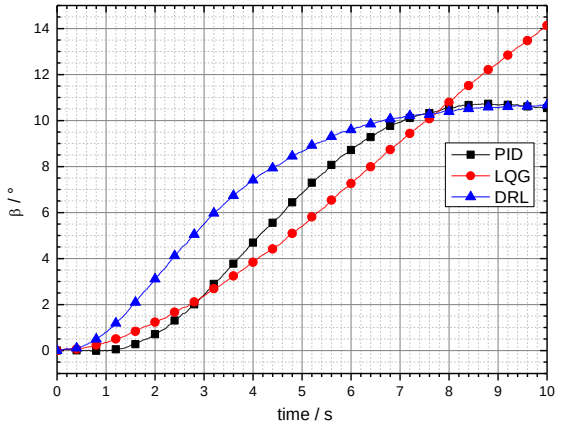
The numerical flight simulation results of the same three controllers of the example UAV in deviation state are shown in Figure 10. Where the exact deviation of the parameters are shown in Table 5.

Table 5 - Main aerodynamics parameters deviation of the example UAV

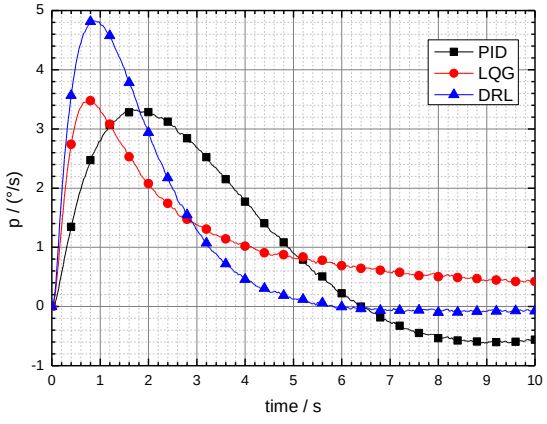
Parameter	Value	Parameter	Value
$N_{C_{Y\beta}}$	1	$N_{C_{lr}}$	2
$N_{C_{l\beta}}$	0	$N_{C_{nr}}$	2
$N_{C_{n\beta}}$	0	$N_{\delta a}$	0.5
$N_{C_{lp}}$	2	$N_{\delta r}$	0.5
$N_{C_{np}}$	2		



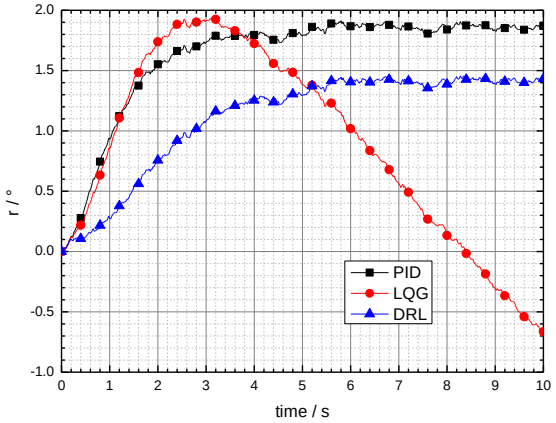
(a) roll angle



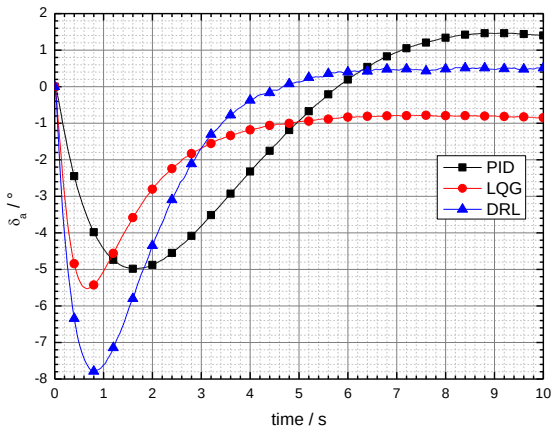
(b) side slip angle



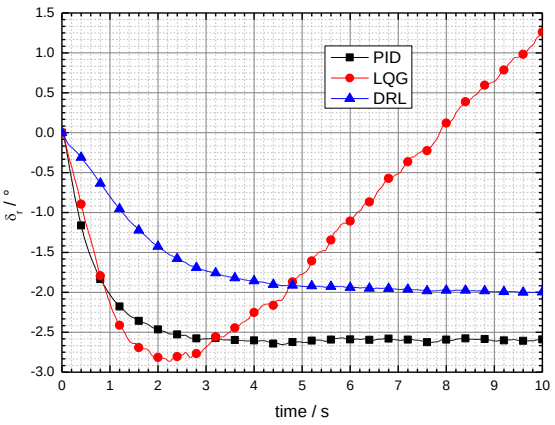
(c) roll rate angle



(d) yaw rate angle



(e) aileron deflect angle



(f) rudder deflect angle

Figure 10 - Roll control responses with different control methods under deviation conditions

4.2 Results analysis

According to the above flight simulation results, the performance of the three controllers in nominal state and deviation state is shown in the table below:

Table 6 - Statistics of the performances of roll angle control

Parameter	PID		LQG		DRL	
	nominal	deviation	nominal	deviation	nominal	deviation
Adjustment time (10% error band) (s)	6.78	9.76	1.86	>10	2.64	2.70
Maximum overshoot	11.9%	22.8%	2.1%	>20%	1.9%	7.8%
Max roll rate (deg/s)	4.03	3.32	9.05	3.48	5.30	4.83
Max aileron deflection angle (deg)	3.11	4.98	7.36	5.53	4.60	7.79

4.2.1 nominal state

In nominal state, the adjustment time of the classical PID control method is 6.78 seconds, and the maximum overshoot is 11.9%. Its performance is the worst of the three methods. This is because the structure of the PID controller is too simple, and based on SISO theory, the performance is reduced when it is applied to the MIMO problem of Lat-Dir flight control of UAV.

The adjustment time of the LQG controller is 1.86 seconds, and the maximum overshoot is 2.1%. The performance is much better than that of PID. This is because the LQG controller make use of the full state information of UAV for feedback control, in this way, the poles of the dynamics system of the UAV can be assigned to any desired position in theory.

The adjustment time of DRL controller is 2.64 seconds and the maximum overshoot is 1.9%, which is close to the performance of LQG controller. Furthermore, it can be found that the maximum roll rate and aileron deflection angle of DRL controller are only about 60% of that of LQG, namely, the cost of DRL controller is much smaller than that of LQG. Since Kalman filter is unnecessary for the DRL controller for state estimation, the DRL controller is better than LQG controller in engineering.

4.2.2 deviation state

In deviation state, the adjustment time of PID controller is increased to 9.76 seconds, and the overshoot is increased to 22.8%. Compared with the nominal state, it is increased by 44% and 10% respectively, and it can be seen that its performance degraded obviously.

The performance of LQG controller degraded seriously, the adjustment time is larger than 10 seconds, resulting in instability actually. This is because the effect of LQG controller depends on the state estimation output of Kalman filter. Under the condition of unknown disturbance and deviation, the Kalman filter can only use the data in nominal state. In this situation, the estimated state is quite different from the actual state, so the performance of LQG controller would not be guaranteed.

The adjustment time of the improved DRL method is 2.70 seconds, and the overshoot is 7.8%. Compared with the nominal state, it only increases by 2% and 6% respectively, and the maximum roll rate is only 0.5deg/s lower than the nominal state. Therefore, it can be concluded that the DRL controller has little change in the deviation state compared with the nominal state and shows good performance robustness.

5. Conclusion

Inspired by PID and LQG control method, this paper presents an improved TD3 deep reinforcement learning Lat-Dir flight control law design method. Flight simulations are carried out in the nominal state and deviation state respectively. The results show that the DRL controller based on improved TD3 algorithm has the advantages of clear physical meaning, simple structure of policy network and strong performance robustness.

References

[1] Richard S. Sutton, Andrew G. Barto. A. *Reinforcement Learning: An Introduction*. 2nd edition, MIT Press, 2018.

[2] Said G. Khan, Guido Herrman, Frank L. Lewis, Tony Pipea, Chris Melhuish. Reinforcement learning and optimal adaptive control: An overview and implementation examples. *Annual Reviews in Control*, 2012, 36 (1): 42-59.

[3] Hwangbo J, Sa I, Siegwart R, et al. Control of a Quadrotor with Reinforcement Learning. *IEEE Robotics and Automation Letters*, 2017, 2(4): 2096-2103.

[4] Stingu P E, Lewis F L. *Adaptive dynamic programming applied to a 6dof quadrotor*. Computational modeling and simulation of intellect: Current state and future perspectives, IGI Global, 2011: 102-130.

[5] Choi S, Kim S, Kim H J. Inverse reinforcement learning control for trajectory tracking of a multirotor UAV. *International Journal of Control, Automation and Systems*, 2017, 15(4): 1826-1834.

[6] Pieter Simke de Vries, Erik-Jan van Kampen. Reinforcement Learning-based Control Allocation for the Innovative Control Effectors Aircraft. *AIAA 2019-0144*.

[7] Ferrari, Silvia, Stengel, Robert F. Online Adaptive Critic Flight Control. *Journal of Guidance Control & Dynamics*, 2004, 27(5):777-786.

[8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529-533, 2015.

[9] Lillicrap TP, Hunt JJ, Pritzel A, et al. Continuous control with deep reinforcement learning. *4th International Conference on Learning Representations*. San Juan, UT, USA. 2016.

[10] Liam Fletcher, Robert Clarke, Thomas Richardson, et al. Reinforcement Learning for a Perched Landing in the Presence of Wind. *AIAA 2021-1282*.

[11]Daksh Shukla, Ratan Lal, Dustin Hauptman, et al. Flight Test Validation of a Safety-Critical Neural NetworkBased Longitudinal Controller for a Fixed-Wing UAS, *AIAA AVIATION FORUM, VIRTUAL EVENT*, June 15-19, 2020.

[12]Rodriguez-Ramos, A., Sampedro, C., Bavle, H. et al. A Deep Reinforcement Learning Strategy for UAV Autonomous Landing on a Moving Platform. *J Intell Robot Syst*, 93, 351–366 (2019).

[13]Bo li Zhigang-Gan-Daqing-Chen-and-Dyachenko-Sergey-Aleksandrovich. UAV Maneuvering Target Tracking in Uncertain Environments Based on Deep Reinforcement Learning and Meta-Learning. *Remote Sensing*. 2020, 12, 3789.

[14]Jingyi xie Xiaodong-Peng-Haijiao-Wang-Wenlong-Niu-and-Xiao-Zheng. UAV Autonomous Tracking and Landing Based on Deep Reinforcement Learning Strategy. *SENSORS*, 2020, 20(19).

[15] Jacob g. elkins, Rohan Sood, Clemens-Rumpf. Bridging Reinforcement Learning and Online Learning for Spacecraft Attitude Control. *Journal of Aerospace Information Systems*, 2022, 19(1): 62-69.

[16]Ranjan Vepa. *Flight dynamics, simulation, and control for rigid and flexible aircraft*. CRC Press, Boca Raton, 2015.

[17]Richard C. Dorf, Robert H. Bishop. *Modern control systems*. 8th edition, Addison Wesley Longman, Inc. 1998.

6. Contact Author Email Address

mailto: wangrui@nwpu.edu.cn

7. Copyright Statement