Proceedings of the 39th Chinese Control Conference
July 27-29, 2020, Shenyang, China

A PID Gain Adjustment Scheme Based on Reinforcement Learning Algorithm for a Quadrotor

Zheng Qingqing, Tang Renjie, Gou Siyuan, Zhang Weizhong

School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081
E-mail: 1107834870@qq.com

Abstract: In this paper, a PID gain adjustment scheme with the basis on Reinforcement Learning Algorithm is proposed, the validity of the scheme is demonstrated with the application to the control of a quadrotor. Specifically, the PPO algorithm of reinforcement learning is utilized in the scheme to adjust a PID controller gains. The procedure and details of the scheme are presented. The experiments prove that the control strategy with this scheme can quickly make the controlled system converge and stabilize. The scheme, compared with a traditional PID controller, has a good performance in terms of control stability, anti-interference stability, and aircraft altitude stability.

Key Words: Quadrotor Control, PID controller, Reinforcement learning, PPO algorithm

1 Introduction

With remarkable development of technology, the popularity of unmanned aerial vehicle (UAV) has been on the rise, and the research on control methods for quadrotor has been one of the research hotspots in the field of aircraft control at home and abroad.

A number of groundbreaking results and applications have been proposed and verified. Common linear control methods for quadrotor aircraft include: proportional-integral-derivative (PID) [1], H_∞ control, LQR / LQG control, etc. [2]; Model-based non-linear control methods include: robust control, backstepping control, sliding mode control, adaptive control, model predictive control (MPC), etc. [3-6]; learning-based flight control methods are mainly focused on fuzzy control methods and neural network methods [7-9], iterative learning (IL), and the reinforcement learning algorithm used in this paper . Traditional UAV control methods are mainly based on precise system mathematical model, therefore, when encountering complex practical issues, the control effect is usually undesirable. In this case, the controller is supposed to autonomously cope with the unpredictable interaction between the aircraft and the environment. For such unpredictable problems, it is necessary to use a controller that can autonomously plan and ratiocinate. The exploitation of intelligent flight control systems is an active topic in the field of control [10]. Literature [11] proposed a control algorithm for training and tracking different flight trajectories of a quadrotor, which was synthesized based on a neural network dynamic model. The literature [12] adopted a PID+Q-Learning learning method to train the drone to navigate to a target position in an unknown environment, which showed that the control of the drone can be implemented by reinforcement learning. Literature [13] used an on-line adaptive reinforcement learning method to train the controller of a fixed-wing UAV, the results show that compared to a general PID controller, the controller trained by reinforcement learning performs much better when tracking a planned path. Literature [14] developed a quad-rotor drone control test environment GYM FC based on OpenAI's reinforcement learning test platform OpenAI GYM, and combined this environment with a quadrotor environment based on ROS-Gazebo to evaluate the latest reinforcement learning algorithm TRPO, DDPG and PPO. Experimental results show that the controller trained based on Proximal Policy Optimization (PPO) is superior to the other two control algorithms in almost any evaluation index. So the PPO algorithm will be exploited to overcome the current limitations of model-based control methods, and thus a more flexible and intelligent method is proposed, in which the PID controller gain adjustment scheme based on reinforcement learning is proposed. The PPO algorithm is applied to find the optimal PID parameters.

The remainder of this letter is structured as follows. Section two describes the mathematical model of the quadrotor. Section three introduces the PID gain adjustment scheme using the PPO algorithm. Section four compared with the classic PID controller to demonstrate the performance improvements of the PID controller trained with PPO and conclusions are drawn in Section five.

2 Non-linear mathematical model of quadrotor

A quadcopter is an aircraft with four motors using a propeller propulsion system. It has six degrees of freedom (DOF), three rotational and three translational. The quadrotor with a cross-shaped symmetrical structure is chosen to show the performance of the PPO scheme as depicted in Fig. 1.

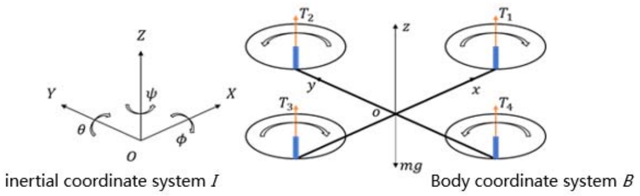

Fig. 1: Body coordinate system and inertial coordinate system

The dynamics of a quadrotor have been investigated by many researchers. According to Newton's second law, the dynamic model of a quadrotor with respect to the inertial coordinate system can be defined as

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = \left[U_1(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\right]/m \\ \dot{v}_y = \left[U_1(\cos\phi\sin\theta\cos\psi - \sin\phi\cos\psi)\right]/m \\ \dot{v}_z = \left[U_1\cos\phi\cos\theta - mg\right]/m \\ \dot{\phi} = p \\ \dot{\theta} = q \\ \dot{\psi} = r \\ \dot{p} = (lU_2 + (I_y - I_z)qr)/I_x \\ \dot{q} = (lU_3 + (I_z - I_x)pr)/I_y \\ \dot{r} = (U_4 + (I_x - I_y)pq)/I_z \end{cases} \tag{1}$$

Where $[x,y,z]^T$ represents the position of the quadrotor in the inertial coordinate; $[\phi,\theta,\psi]^T$ is the attitude angle in the body coordinate system; m is the mass of the quadrotor; $[p,q,r]^T$ represents the angular velocity of the quadrotor; $[I_x,I_y,I_z]^T$ is the moment of inertia matrix; $U=[U_1,U_2,U_3,U_4]^T$ is the control input which is determined by the rotation speeds of the motors $\omega=[\omega_1,\omega_2,\omega_3,\omega_4]^T$, and U_1,U_2,U_3,U_4 represent the thrust control amount, roll torque control amount, pitch and yaw torque control amount. The relationship between the U and the rotation speed of each rotor can be expressed as

$$U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} T_1+T_2+T_3+T_4 \\ T_4-T_2 \\ T_3+T_1 \\ T_1-T_2+T_3-T_4 \end{bmatrix} = \begin{bmatrix} K_t & K_t & K_t & K_t \\ 0 & -K_t & 0 & K_t \\ -K_t & 0 & K_t & 0 \\ K_d & -K_d & K_d & -K_d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{2}$$

where K_t is the comprehensive lift coefficient and K_d is the comprehensive resistance coefficient.

3 PPO algorithm

The PPO algorithm is proposed by OpenAI. It is a new type of Policy Gradient algorithm, since the traditional policy gradient algorithm is greatly affected by the step size, too big step size will also deteriorate the final learning effect. In response to this problem, the PPO algorithm proposes a new objective function that can be updated in small batches through multiple training steps, thereby solving the problem of step size selection in traditional policy gradient algorithms. The PPO algorithm can achieve an optimal balance in terms of algorithm complexity, accuracy, and difficulty of implementation. There are two main ways to implement, the first is implemented by CPU simulation. The second is implemented by GPU-accelerated simulation [15]. This paper choses the first method, the implementation and basic principles are shown below.

Define $r_t(\theta)$ as the probability ratio under the new and old policies, $r_t(\theta) = \dfrac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}$

$$L(\theta) = \hat{\mathrm{E}}t\left[r_t(\theta)A_t - \beta KL\left[\pi_{\theta_{old}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)\right]\right] \tag{3}$$

Compute $d = \hat{\mathrm{E}}_t\left[KL\left[\pi_{\theta_{old}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)\right]\right]$

— If $d < d_{t\arg}/1.5, \beta \leftarrow \beta/2$

— If $d > d_{t\arg} \times 1.5, \beta \leftarrow \beta \times 2$

Where π_θ is a stochastic policy and θ_{old} is the vector of policy parameters before update. A_t is an estimator of the advantage function at time step t. The value of the estimated advantage can be interpreted as an advantage in value gained by taking a certain action over the action from the current policy π_θ [16]. It is assumed that the difference between π_θ and $\pi_{\theta_{old}}$ is not large. The penalty coefficient β is used for the next policy update. In other words, if the same state inputs the network in different times, the difference in the probability distribution of the actions obtained by the neural network cannot be too large, that is, *KL* divergence has a range limit. The expectation $\hat{\mathrm{E}}_t[\ldots]$ indicates the empirical average over a finite batch of samples.

4 Gain Adjust Scheme

In this section, we propose the PID gain adjustment scheme based on the PPO algorithm for a quadrotor. The basic structure is shown in Fig. 2.

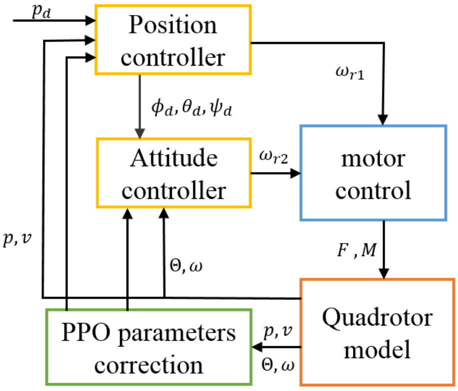

Fig. 2: PID controller gain adjustment scheme for a quadrotor

4.1 Training process of PPO algorithm

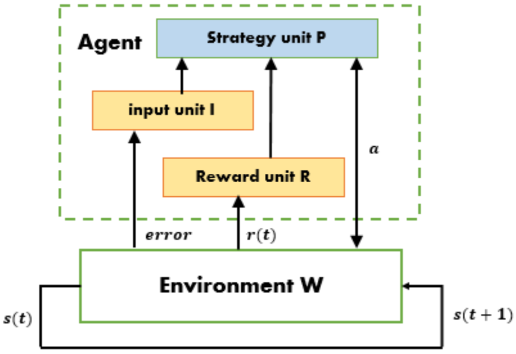

Fig. 3: PPO algorithm working structure

Its structure is composed of two parts, the quadrotor working environment W and the intelligent control system Agent. The Agent is composed of a strategy unit **P**, an input unit **I**, and a reward unit **R**. Its working mode is that the agent senses the environmental state **s** at time t, including the position error $[x_d - x, y_d - y, z_d - z]^T$,Attitude angle error $[\phi_d - \phi, \theta_d - \theta, \psi_d - \psi]^T$ and the angular velocity of each axis $[p, q, r]^T$, these data can be measured by the gyroscope and electronic speed controller. The information comes from the continuous observation state space $s_t \in S$. Based on the current state information s_t and the environment to react to the enhanced information R observed by the agent, the agent selects an action a_t to act on the environment through the strategy **P**, and the environment enters the next state s_{t+1}. This cycle is repeated until obtaining a satisfactory target state. The action function a_t is the control signal $\{\omega_1, \omega_2, \omega_3, \omega_4\}$ of the quadrotor, and $a_t \in A$. The reward function r_t reflects the value of a_t indicating the performance is good or bad. Our goal is to find a parameterized policy π_θ , where θ is called policy parameter, which maximizes the average value over states

4.2 PID controller based on PPO algorithm

The rise time x_o , overshoot x_r and amplitude x_a are respectively used as the evaluation indicators of the PID controller, and the following formula is defined as [17]

$$V(x) = exp\left[-\left(\frac{x_o - x_o^d}{\beta_1}\right)^2 - \left(\frac{x_r - x_r^d}{\beta_2}\right)^2 - \left(\frac{x_a - x_a^d}{\beta_3}\right)^2\right] \tag{4}$$

Here, x_o^d, x_r^d and x_a^d are the expected rise time, overshoot and amplitude, respectively. β_1, β_2 and β_3 are the widths of the evaluation functions. $V(x)$ is the current control evaluation of the control performance of the system ($V(x) \in [0,1]$). Thus the PPO algorithm modifies the PID controller gains as

$$\begin{cases} P(t+1) = P(t) \\ P^+(t+1) = P(t) + \Delta P(t) \\ P^-(t+1) = P(t) - \Delta P(t) \end{cases} \tag{5}$$

When the PID parameters are not modified at time t + 1, then $P(t+1) = P(t)$; when the PID parameter is modified in the forward direction, then $P^+(t+1) = P(t) + \Delta P(t)$; when the PID parameter is modified in the negative direction, there is $P^-(t+1) = P(t) - \Delta P(t)$ .The parameter adjustment of PID controller is realized by the PPO learning model. where $\Delta P(t)$ is the adjustment item, and its implementation is shown as

$$\Delta P(t) = \alpha_0 \times (1 - V(t)) \frac{(V(t) - V(t-1))^2}{P(t) - P(t-1)} \tag{6}$$

Where α_0 is the controller adjustment amplitude parameter, which is generally set to 1. When the evaluation formula $V(x)$ changes greatly before and after the time, $\Delta P(t)$ is larger, which indicates that a greater degree of correction is required. When the difference between the time before and after $V(x)$ is small, or even close to 0, $\Delta P(t)$ is small, and it indicates that the controller at this time is in a better control state, fine-tuning the control parameters or not adjusting.

In order to finally compare the performance of the UAV control algorithm based on PPO training and the traditional PID control algorithm, and quantify the final index of the control effect, here we define the evaluation index of the control effect as "control error". The meaning is the difference between the current position and attitude of the drone and the target point. If the error is 0% after training, the training has reached the preset goal. In the training process, because we cannot guarantee that each training error can reach 0%, we only count the time for the training process with an error within 10%.

During training, the value function is defined as the following expression:

$$r(t) = -\left(k_1 \times \|\Delta p(t)\| + k_2 \times \|\Delta\Theta(t)\| + k_3 \times \|\Delta\omega(t)\|\right) \tag{7}$$

Here, $\Delta p(t)$ is the position error at time t, $\Delta\Theta(t)$ represents the attitude error, $\Delta\omega(t)$ represents the attitude angular velocity error (if the target attitude angular velocity is not limited, it is taken as 0). The weight k_1 takes a larger value, and the weights k_2 and k_3 take a smaller value. Because the control of the quadrotor usually requires a higher position accuracy, and the attitude control accuracy requirements are relatively low.

5 Simulation Validation

5.1 PID controller based on PPO algorithm

We set the number of episodes as 5000 and 1000 steps per episode (the number of trainings and iteration steps is set according to experience and is related to the complexity of the model and the performance of the machine being trained).

For each training, start with (0,0,0,0,0,0) and (5,3,1,0,0,0) as the final goal. At each iteration, the gain parameter adjustment of the PID controller is conducted according to the current state, and then call the quadrotor model simulation. The reward of each step is calculated according to the simulated state, and the PPO algorithm parameters are updated. Performing 1000 steps to calculate the total return and save all PID parameters after calibration. After completing 5000 trainings in this way, the reward of the PPO algorithm is expected to converge, and the relationship between the training quadrotor state and the PID parameters also tends to be optimal, so different PID gain parameters are selected according to different states.

Finally, the PPO parameters settings: the number of PPO trainings is 5000, the iteration steps are 1000, the discount factor is 0.95, Actor –learning rate is 0.01, Critic-learning rate is 0.02, d_{targ} is 0.01.

We save the reward data for each iteration training, and perform simulation analysis on these data. The results are shown in Fig. 4.

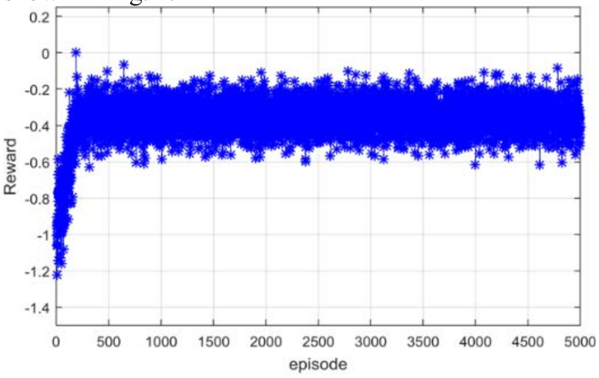

Fig. 4: Relationship between each training cycle and the average value of rewards

The average reward range of the simulation results is approximately [-1, 0]. It can be seen from the simulation results that in the first training cycle, the reward after about 500 rounds of iterative learning training tends to be stable, and then changes in a relatively stable range until the end of training.

5.2 Comparison PPO-corrected PID controller and traditional PID controller

In this section, the performance of a PID controller modified by the PPO scheme is compared with a traditional PID controller with and without noise interferences. The noise signal in the position state is set to a uniform random number between [-0.002m, 0.002m], and the noise signal in the attitude angle is set to a uniform random number between [-0.0025rad, 0.0025rad]. Counting the rise time, overshoot, and steady-state error of each experiment.

First, the simulation results of the attitude controller without noise interference are analyzed. The simulation results are shown in Table 1.

Table 1: Controller performance analysis (no noise interference)

performance	axis	PPO	PID
Rise time (s)	z	0.95	1.1
	x	4.55	5.15
	y	3.95	4.81
Overshoot (%)	x	4.1	5.2
	y	3.5	4.1
	z	0.92	1.1
Control error (%)	x	0.17	0.22
	y	0.08	0.13
	z	2.88	3.94

It can be observed from Table 1 that in the absence of noise interference, the performance of the PID controller after PPO training is better than that of the PID controller before calibration. There is a certain degree of performance improvement in the perspectives of rise time, overshoot and control error: the rise time increased by 13.64%, 11.65%, and 17.88%, the overshoot performance index increased by 21.15%, 14.63%, and 16.36%, and the control error index increased by 22.73%, 38.46%, and 27.46%, respectively.

Secondly, we analyze the simulation results after adding noise interference, results are shown in Table 2. In the actual system, due to the interference of environmental factors, the controller output will often produce a certain error, which will affect the control system, after the PID controller gain is adjusted by the PPO algorithm, the output error can be reduced by sacrificing the overshoot and rise time.

Table 2: Controller performance analysis (noise interference)

performance	axis	PPO	PID
Rise time (s)	z	1.34	1.1
	x	5.82	5.15
	y	5.34	4.81
Overshoot (%)	x	5.73	5.2
	y	4.82	4.1
	z	1.22	1.1
Control error (%)	x	1.54	4.42
	y	1.55	3.55
	z	3.25	8.65

It can be known from Table 2 that when there is a large interference in the environment, the controller after PPO training increases the stability of the controller at a smaller control cost (increasing system rise time and overshoot). The control error index increased by 64.16%, 56.34% and 62.43% respectively.

In summary, the PID controller with the PPO adjustment scheme can show an excellent control performance. When there is a severe interference in the external environment, it has a strong ability to deal with sudden conditions and always keeps the quadrotor in a more stable control state.

5.3 Attitude simulation and performance analysis

In the experiments of previous Section 5.2, a simulation result with noise and typical characteristics was selected and compared with the uncorrected PID controller simulation results to analyze the performance of the controller based on PPO correction. This simulation was done 100 times. The settings are as follows: make the initial position (0,0,0), the target position is (5,3,1), and the initial and end attitude angles are (0,0,0). For comparison, an unmodified PID controller is used. The simulation time is 40 seconds, and the time to reach the target position is set to 15 seconds. After reaching the target, the drone hover in the air for 25 seconds.

The attitude simulation results are shown in Fig.5.

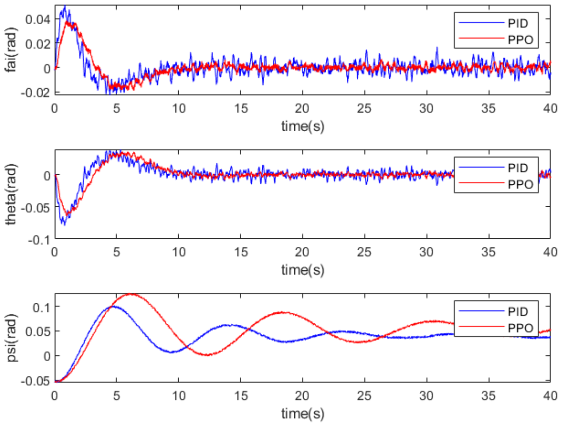

Fig. 5 The comparison of quadrotor flight attitude angle

It can be seen from Fig. 5 that the PID controller corrected by PPO has strong anti-interference ability of ϕ and θ. After the controller enters a stable state (after 10s), even if there is an analog noise signal, the ϕ and θ are much smaller than the situation with the traditional PID controller.

5.4 Position simulation and performance analysis

Next, the relative coordinate position and flight trajectory of the UAV based on the inertial reference system are simulated and analyzed. The simulation results are shown in Fig. 6.

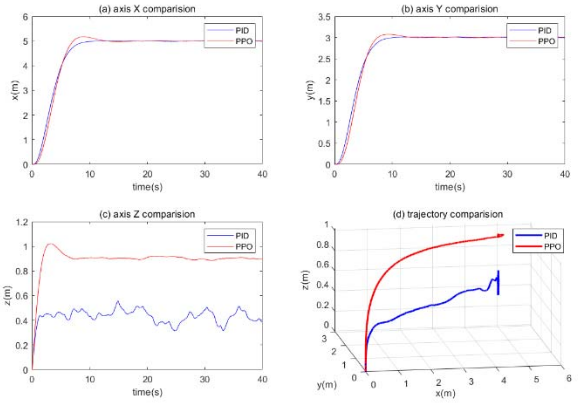

Fig. 6 flight path of quadrotor

From the position simulation results in Fig. 6, it can be known that using the traditional PID control algorithm, during the quadrotor flight, the x-axis and y-axis can reach the target position quickly and stabilize. However, the tracking of the z-axis is poor, the quadrotor cannot reach the predetermined target point, which has a more obvious effect of shaking in the hovering state. The PPO-trained controller has strong stability and quadrotor reaches the predetermined target point and the UAV is stable in the hovering state without obvious shaking.

6 Conclusion

Aiming to propose a PID gain adjust scheme based on reinforcement learning algorithm, the quadrotor is applied to validate this scheme. This paper presentes a control system that corrects the PID controller parameters for a quadrotor by the PPO algorithm. This method greatly reduces the training time, and about 500 rounds can make the strategy converge. Compared with the uncorrected PID controller, the performance of the PID controller corrected by the PPO is shown much better. The results show that the controller designed in this paper has some advantages over the PID controller in terms of response time, overshoot, and control error. Additionally, the corrected PID controller can significantly improve the stability of the controller at a smaller control cost in the presence of external interferences. Moreover, it has strong anti-interference ability in roll and pitch attitude. From the trajectory simulation results, the controller based on reinforcement learning training can well guarantee the high stability of the quadrotor.

References

[1] John S. Artificial intelligent-based feedforward optimized PID wheel slip controller[C]// AFRICON 2013. IEEE, 2013.

[2] Bouabdallah S, Noth, André, Siegwart R . PID vs LQ control techniques applied to an indoor micro quadrotor[C]// IEEE/RSJ International Conference on Intelligent Robots & Systems. IEEE, 2004.

[3] Madani T, Benallegue A. Backstepping Control for a Quadrotor Helicopter[J]. 2006.

[4] Xu R , Ozguner U. Sliding Mode Control of a Quadrotor Helicopter[C]// Proceedings of the 45th IEEE Conference on Decision and Control. IEEE, 2006.

[5] Diao, Chen,Xian, Bin,Yin, Qiang, et al. A nonlinear adaptive control approach for quadrotor UAVs[C].//2011 8th Asian Control Conference. [v.1].2011:223-228.

[6] Farid Kendoul.Survey of Advances in Guidance, Navigation, and Control of Unmanned Rotorcraft Systems[J]. Journal of Field Robotics,2012,29(2):315-378.

[7] Nicol C , Macnab C J B , Ramirez-Serrano A . Robust neural network control of a quadrotor helicopter[C]// 2008 Canadian Conference on Electrical and Computer Engineering. IEEE, 2008

[8] Dierks T, Jagannathan S. Output Feedback Control of a Quadrotor UAV Using Neural Networks[M]. IEEE Press, 2010.

[9] Bansal S, Akametalu A K, Jiang F J, et al. Learning quadrotor dynamics using neural network for flight control[C]. Decision and Control. IEEE, 2016:4653-4660.

[10] F. Santoso, M. A. Garratt, and S. G. Anavatti, “State-of-the-art intelligent flight control systems in unmanned aerial vehicles,” IEEE Transactions on Automation Science and Engineering, 2017.

[11] Bansal S, Akametalu A K, Jiang F J, et al. Learning quadrotor dynamics using neural network for flight control[C]. Decision and Control. IEEE, 2016.

[12] Pham H X, La H M, Feilseifer D J, et al. Autonomous UAV Navigation Using Reinforcement Learning[J]. arXiv: Robotics, 2018.

[13] Kimathi, Ethe K, Kihato, et al. Application Of Reinforcement Learning In Heading Control Of A Fixed Wing UAV Using X-Plane Platform[J]. International Journal of Scientific & Technology Research, 2017, 6(2): 285-289.

[14] Koch W, Mancuso R , West R , et al. Reinforcement Learning for UAV Attitude Control[J]. 2018.

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov. Proximal Policy Optimization

Algorithms[J]. 2017.

[16] Hwangbo J , Sa I , Siegwart R , et al. Control of a Quadrotor with Reinforcement Learning[J]. IEEE Robotics and Automation Letters, 2017:1-1.

[17] Zhu Weihua. Simulation Study of Diesel Engine Adjustment Based on Reinforcement Learning PID Controller [D]. Harbin Engineering University, 2011.