IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 1 June 2022, accepted 15 June 2022, date of publication 22 June 2022, date of current version 29 June 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3185424

Improved Reinforcement Learning Using Stability Augmentation With Application to Quadrotor Attitude Control

HANGXING WU, HUI YE, WENTAO XUE, AND XIAOFEI YANG, (Member, IEEE)
School of Electronic and Information, Jiangsu University of Science and Technology, Zhenjiang 212100, China

Corresponding author: Hui Ye (yehuicc@just.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61903163, and in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 19KJB510023.

ABSTRACT Reinforcement learning (RL) has been successfully applied to motion control, without requiring accurate models and selection of control parameters. In this paper, we propose a novel RL algorithm based on proximal policy optimization algorithm with dimension-wise clipping (PPO-DWC) for attitude control of quadrotor. Firstly, dimension-wise clipping technique is introduced to solve the zero-gradient problem of the PPO algorithm, which can quickly converge while maintaining good sampling efficiency, thus improving the control performance. Moreover, following the idea of stability augmentation system (SAS), a feedback controller is designed and integrated into the environment before training the PPO controller to avoid ineffective exploration and improve the system's convergence. The eventual controller consists of two parts: the first is the result of the actor neural network in the PPO algorithm, and the second is the output of the stability augmentation feedback controller. Both of them directly use an end-to-end style of control commands to map the system state. This control architecture is applied in the attitude control of the quadrotor. The simulation results show that the quadrotor can quickly and accurately track the command and has a small steady-state error after the training by the improved PPO algorithm. Meanwhile, compared with the traditional PID controller and basic PPO algorithm, the proposed PPO-DWC algorithm with stability augmentation framework has better performance in tracking accuracy and robustness.

INDEX TERMS Reinforcement learning, attitude control, proximal policy optimization, quadrotor, dimension-wise clipping, stability augmentation system.

I. INTRODUCTION

In recent years, reinforcement learning (RL) has made rapid progress in the field of artificial intelligence research. Combined with deep learning and data progression, deep reinforcement learning (DRL) algorithms modeled by neural networks are now successfully applied in a variety of scenarios such as investing [1], gaming [2], and traffic control [3]. At the same time, in the robot control system, the application of DRL for continuous tasks has become one of the most popular research topics, including the motion control of unmanned aerial vehicles [4], unmanned surface vehicles [5], and autonomous underwater vehicles [6].

Quadrotor UAVs are widely used in power inspection [7], urban planning [8], agricultural monitoring [9], disaster rescue [10] and other fields, and is currently developing in a safer and more efficient direction. The quadrotor is a typical underactuated nonlinear strong coupled system with 6-DOF (six degrees of freedom) for the rotational and translational motion and only four actuators for control input. For the attitude control of the quadrotor, it is not enough to be able to hover, but also perform large maneuvers in a tough environment [11]. Factors such as air resistance, the gyroscopic moment generated by the rapid rotation of the motor during flight, and the uneven distribution of the mass will affect the stability of the quadrotor flight. Many advanced control algorithms have been proposed and applied to quadrotor flight control systems, such as sliding mode control [12], adaptive control [13], robust control [14], active disturbance rejection control [15] and model predictive control [16]. In the ideal environment, the quadrotor shows excellent performance in both agility and precision. However, when there

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Tang.

are uncertainties in the environment, typical control methods that rely on precise quadrotor models find it challenging to achieve the control requirements. On the other side, most control techniques do not consider the actuator's saturation limitations in the design process, which will degrade the controller's performance and lead to instability of the system in extreme cases. Alternatives to the conventional control techniques are available through intelligent controllers [17]. In recent years, thanks to the development of machine learning, the intelligent flight control system based on DRL developed by neural networks has become a trendy research field.

It has been proven that RL algorithms can achieve success in situations close to the complexity of the real world [18]. Deep research has been carried out on the policy learning for autonomous control of quadrotors. In [19], RL achieves stable quadrotor control by training a neural network policy in a model-free manner. Combined with low-resolution images, a control policy trained with RL in [20] was used to achieve autonomous landing of an aircraft for the first time. The RL controller designed in [21] has successfully completed the tasks of hovering and trajectory tracking for a real quadrotor. Remarkably, more advanced RL algorithms such as soft actor-critic (SAC) [22], twin delayed deep deterministic policy gradient (TD3) [23] and proximal policy optimization (PPO) [24] are gradually being used in the control system of the quadrotor. An actor-critic neural-network-based controller was presented in [25] to improve the quadrotor trajectory tracking performance. In [26], MAV has successfully completed autonomous navigation under a gust environment using the SAC algorithm as a DRL framework. Sequential deep q-network (SDQN) was first used as an end-to-end learning paradigm to train control policies for autonomous landing of UAVs in [27]. Advanced tasks of UAVs in actual flight are completed in [28] by the control policy of the deep deterministic policy gradient (DDPG) algorithm. In [29], the state-of-the-art DRL algorithm PPO was used to explore the control policy of the quadrotor position loop while maintaining good sampling efficiency. Moreover, the method of integrating classical controllers with RL policies has been shown to have higher learning efficiency [30]. An MPC-guided RL policy search algorithm is studied in [31] for learning quadrotor autonomous flight. In order to improve the tracking accuracy and robustness, a method that introduces an integral compensator in the actor-critic neural network was investigated in [32]. In [33], the PPO algorithm was suggested to correct the parameters of the quadrotor PID controller, which significantly reduced the training time and ensured the high stability of the quadrotor.

Among the advanced RL algorithms off the shelf, PPO has been evaluated as one of the most suitable algorithms for synthesizing high-precision attitude flight controllers [34]. However, due to its structural factors, problems such as vanishing gradients of clipping samples can also lead to inefficient learning of agents in high action-dimensional tasks. Some variant algorithms based on PPO have been proposed to solve the zero gradient problem. In [35], a two-phase policy gradient algorithm (PPG) that advances training and distills features is proposed to optimize the value function using a higher-level sample reuse method, which solves gradient vanishing and improves sample efficiency. A PPO algorithm based on relative Pearson (RPE) divergence is proposed in [36], through which an explicit minimization target can be yielded, and the latest policy is restricted to the baseline policy. Although the improvement of the algorithm can improve the training efficiency in the benchmark, the application of the PPO needs to be further studied for the specific quadrotor control system.

In this paper, the original PPO algorithm is improved for training the quadrotor attitude controller to achieve higher precision control in a shorter time. The algorithm is altered in two ways: algorithm structure and combination of classical control theory. The results of the original PPO algorithm are used as a benchmark to demonstrate the advantages of the new algorithm. The main contributions of this paper are listed as follows:

1) By estimating the advantage value of dimension importance sampling (IS) weight clipping, a new proxy target with a mechanism of clipping the IS weight of each action dimension separately is proposed to improve sample efficiency and achieve stable training for quadrotor attitude control.

2) A stability augmentation controller is introduced into the RL gain to speed up the process of training the quadrotor control policy and significantly improve the motion control accuracy of the quadrotor.

The remainder of this paper is organized as follows. Section II introduces the modeling of quadrotor and the basic principle of PPO algorithm. In Section III, the disadvantages of the PPO algorithm are analyzed. Then, the PPO algorithm with dimension-wise clipping and the PPO algorithm combined with stability augmentation controllers are introduced, respectively. Section IV gives simulation details and results, and our conclusions are summarized in Section V.

II. BACKGROUND

A. DYNAMIC MODEL OF QUADROTOR

The basic structure of the quadrotor is shown in Fig. 1. The UAV control system is an under-actuated system with four inputs and six outputs. The inertial coordinate system and the body coordinate system fixed on the quadrotor are established to describe the attitude of the quadrotor. $F_i(i = 1, 2, 3, 4)$ are the thrusts generated by the four rotors and $\Omega_i(i = 1, 2, 3, 4)$ are the rotational speeds of the rotors. ϕ, θ and ψ denote three Euler angles of the quadrotor.

For rotational motion, applying Euler's equation of rotation to the quadrotor frame, the absolute derivative in dynamic coordinates can be expressed as follows:

$$\boldsymbol{M} = \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} = \boldsymbol{M}_\tau + \boldsymbol{M}_c + \boldsymbol{M}_f \tag{1}$$

where $\boldsymbol{I} = diag(I_x, I_y, I_z)$ is the diagonal inertia matrix of the quadrotor and $\boldsymbol{\omega} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ is the angular velocity of the three axes of the quadrotor. $\boldsymbol{M}$ is mainly composed of the following parts: the control torque $\boldsymbol{M}_\tau$, the gyroscopic effect

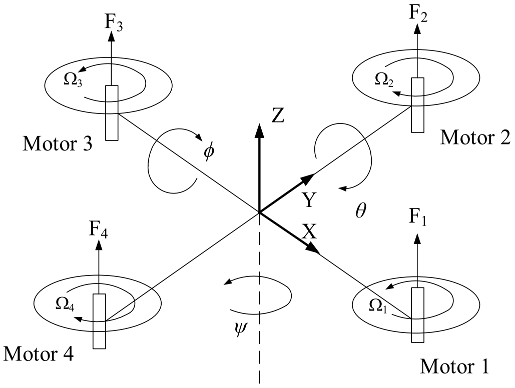


FIGURE 1. The structure model of the quadrotor.

torque $\boldsymbol{M}_c$ and the rotational dynamic resistance torque $\boldsymbol{M}_f$. The control torque $\boldsymbol{M}_\tau$ can be obtained as

$$\boldsymbol{M}_\tau = \begin{bmatrix} L(F_4 - F_2) \\ L(F_3 - F_1) \\ c(F_1 - F_2 + F_3 - F_4) \end{bmatrix} = \begin{bmatrix} Lb(\Omega_4^2 - \Omega_2^2) \\ Lb(\Omega_3^2 - \Omega_1^2) \\ cb(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix} \tag{2}$$

where L is the distance from each rotor to the center of mass, c is the proportional coefficient of reaction torque and thrust, and b is the thrust factor. The gyroscopic effect from spinning rotors can be written as $\boldsymbol{M}_c = [-I_p\dot{\theta}\Omega_d, I_p\dot{\phi}\Omega_d, 0]^T$, where the disturbance effect from each rotor is $\Omega_d = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$ and I_p is the moment of inertia of each rotor. The rotational resistance torque can be expressed as $\boldsymbol{M}_f = [-d_\phi\dot{\phi}, -d_\theta\dot{\theta}, -d_\psi\dot{\psi}]^T$, where d_ϕ, d_θ and d_ψ are the drag coefficients.

Finally, the nonlinear dynamic rotation equation of the quadrotor is as follows:

$$\begin{cases} \ddot{\phi} = [Lb(\Omega_4^2 - \Omega_2^2) - I_p\dot{\theta}\Omega_d - d_\phi\dot{\phi} + \dot{\theta}\dot{\psi}(I_y - I_z)]/I_x \\ \ddot{\theta} = [Lb(\Omega_3^2 - \Omega_1^2) + I_p\dot{\phi}\Omega_d - d_\theta\dot{\theta} + \dot{\phi}\dot{\psi}(I_z - I_x)]/I_y \\ \ddot{\psi} = [cb(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) - d_\psi\dot{\psi} + \dot{\varphi}\dot{\theta}(I_x - I_y)]/I_z \end{cases} \tag{3}$$

Equation (3) should be discretized for RL implementation to define a Markov Decision Process (MDP). In our work, the state space is defined as $\boldsymbol{s}_t = \{\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}\}$ which includes the Euler angles and the angular velocity. The action space is selected as $\boldsymbol{a}_t = \{a_1, a_2, a_3, a_4\}$, where $a_i, i = 1, 2, 3, 4$ is the throttle level of each rotor and can be obtained by $a_i = \Omega_i^2/\Omega_m^2$. Ω_m is maximum speed of each rotor. The state transition function can be obtained as follows:

$$\boldsymbol{s}_{t+1} = \boldsymbol{f}(\boldsymbol{s}_t, \boldsymbol{a}_t) \tag{4}$$

where $\boldsymbol{f} = [f_1, f_2, f_3 f_4, f_5, f_{6,}]^T$ is the smooth nonlinear function vector.

B. REINFORCEMENT LEARNING

From the basic principles of RL, it can be seen that RL has some essential differences compared with supervised learning and unsupervised learning. The data samples are static training sets with labels in traditional supervised learning. However, RL is a continuous decision-making process. In the training process of RL, the agent does not have any instruction information, and it updates its policy by getting reward values through interaction with the environment. The agent finally obtains the best policy through continuous trial and error.

The basic block diagram of RL is shown in Fig. 2. It trains an optimal policy through continuous trial-and-error interactions between the agent and environment. Agent generally consists of two parts, RL algorithm and policy, where the policy is usually a function approximator with adjustable parameters, such as neural network. When training begins, the agent chooses an action a_t to act on the environment. The entire environment model reaches a new state $\boldsymbol{s}_t$, generating a reward value r_t simultaneously. The RL algorithm continuously updates policy parameters based on action $\boldsymbol{a}_t$, state $\boldsymbol{s}_t$, and reward r_t. The agent and environment interact in a continuous loop to generate data samples. The agent finds the optimal control policy when the accumulated reward is maximized during training.

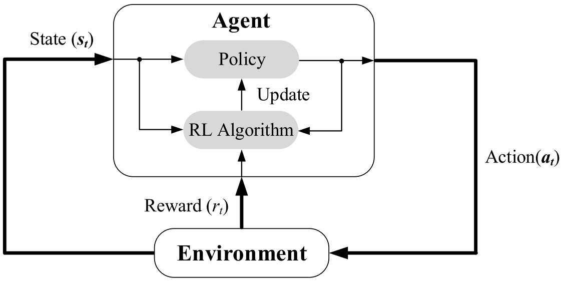


FIGURE 2. Basic block diagram of RL algorithm.

Policy gradient (PG) with importance sampling (IS) is a classic policy-based RL algorithm. The PG algorithm mainly sets $J(\boldsymbol{\lambda})$ as the performance function and maximizes $J(\boldsymbol{\lambda})$ by updating the policy. In each iteration, PG updates the new policy $\pi_{\tilde{\lambda}}$ by analyzing the current policy π_{λ} from time step t:

$$J_\lambda(\tilde{\lambda}) = \sum\nolimits_t \left[\rho_t(\tilde{\lambda}) A_{\pi_\lambda}(\boldsymbol{s}_t, \boldsymbol{a}_t)\right] \tag{5}$$

where $\rho_t(\tilde{\boldsymbol{\lambda}}) = \frac{\pi_{\tilde{\lambda}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_\lambda(\boldsymbol{a}_t|\boldsymbol{s}_t)}$ is the IS weight and $A_{\pi_\lambda}(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is the advantage value.

The traditional PG algorithm is greatly affected by the large IS weight, which directly leads to the unable final learning effect or the long policy convergence time. PPO algorithm proposes a new objective function to solve the problem of selecting step size by storing multiple training steps for mini-batch updating. Moreover, it has two other characteristics. One is that it has been proven to have excellent performance in solving continuous action problems with networks. The other is that it adds importance sampling technology to update, so that PPO algorithm can achieve the optimal balance in terms of algorithm complexity, accuracy and implementation difficulty.

PPO uses the objective clipping function to bound the policy update of the current policy to achieve stable learning. Policy π_{λ_i} generates the current sample batch $B_i = \{(\boldsymbol{s}_{i,0}, \boldsymbol{a}_{i,0}, r_{i,0}), \ldots, (\boldsymbol{s}_{i,N-1}, \boldsymbol{a}_{i,N-1}, r_{i,N-1})\}$ with length N when the iteration starts from the i-th time. Then according to multiple mini-batches sampled in B_i, π_λ completes the update. Due to the difference between the policy π_{λ_i} that generates B_i and the target policy π_λ of policy updating, PPO calibrates the statistical difference according to the IS weight ρ_t. In addition, PPO reduces the IS weight in order to limit the amount of policy updates to ensure the stability of learning. Therefore, the objective function of PPO is given by the following:

$$\hat{J}_{PPO}(\boldsymbol{\lambda}) = \frac{1}{M}\sum_{t=0}^{M-1} \min\left(\rho_t \hat{A}_t, clip(\rho_t, 1-\varepsilon, 1+\varepsilon)\hat{A}_t\right) \quad (6)$$

where $\hat{A}_t$ is the estimate of $A_{\pi_{\lambda_i}}(\boldsymbol{s}_t, \boldsymbol{a}_t)$ and B_i randomly sampled M samples in each mini-batch.

However, it is precisely because PPO clipping the overall likelihood ratio causes the gradient of the cutting samples to vanish completely. Therefore, in the task of high action dimension, PPO also has the problem of low sample efficiency, which affects learning efficiency and tracking accuracy in complex quadrotor systems. The improved PPO algorithm presented in this work is an attempt to solve this problem.

C. NETWORK STRUCTURE

The neural network used in PPO is based on the Actor-Critic network structure. Due to the good generalization ability of neural networks, the multilayer perceptron (MLP) structure proposed in [19] is used. There may be a better network configuration, but in fact the neural network is quite versatile. Its configuration has been able to approximate a controller for similar tasks. MLP is a fully connected feedforward artificial neural network trained using supervised learning with backpropagation. Its initial weight is a Gaussian random number with mean 0 and standard deviation 1. The structure of the actor-critic neural network is shown in Figure 3. For the Actor network, the input layer is the quadrotor state $\boldsymbol{s}_t$, and the output layer is the signal that controls the rotational speed of the four rotors of the quadrotor. Each network has two hidden layers, each with 64 nodes, which are neurons with *tanh* activation functions. The critic neural network has the same structure. The only difference is that its output is an estimated value function that evaluates the advantage of selecting a given action $\boldsymbol{a}_t$ in a given state $\boldsymbol{s}_t$.

III. PROPOSED APPROACH

A. PPO WITH DIMENSION-WISE CLIPPING (PPO-DWC)

In (6), r_t is a function of the optimization policy variable $\boldsymbol{\lambda}$, $\hat{A}_t$ is fixed for the policy π_{λ_i} that is generated from the given action of the current sample batch B_i. Therefore, in general, the cost maximization for $\boldsymbol{\lambda}$ is to increase ρ_t when $\hat{A}_t > 0$, and decrease ρ_t when $\hat{A}_t < 0$. PPO restricts the number of policy updates by clipping the objective function. The advantage is that this clipping mechanism can prevent ρ_t from becoming too small or too large, especially for many complex environments, a stable update range is more conducive to faster and more efficient training. The disadvantage is that when the dimension sample is too high, it is easy to cause a zero gradient problem, resulting in local optimization. When we simplify the clipped objective function:

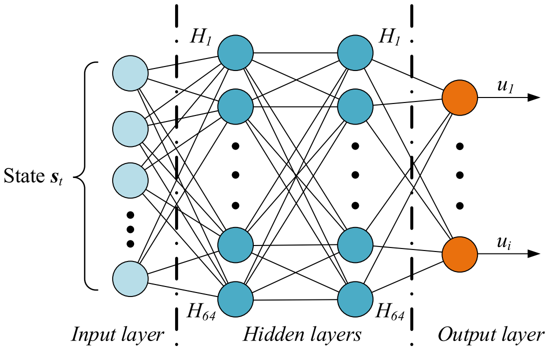


FIGURE 3. The structure model of the quadrotor.

$$\hat{J}_t = \min\left(\rho_t \hat{A}_t, clip(\rho_t, 1-\varepsilon, 1+\varepsilon)\hat{A}_t\right) \quad (7)$$

It can be seen that when $\hat{A}_t < 0$ and $\rho_t < 1-\varepsilon$, $\hat{J}_t = (1-\varepsilon)\hat{A}_t$, and when $\hat{A}_t > 0$ and $\rho_t > 1+\varepsilon$, $\hat{J}_t = (1+\varepsilon)\hat{A}_t$. In these two cases, $\hat{J}_t$ is a constant and the gradient disappears. The problem of this kind of zero gradient is very severe [37], especially in high-action dimension tasks. Because PPO directly clips the loss function, the sample efficiency is strongly affected by the zero-gradient samples created by PPO.

Gaussian distribution is often considered as a random policy for RL when performing action tasks:

$$\boldsymbol{a}_t \sim \pi_{\boldsymbol{\lambda}}(\cdot|\boldsymbol{s}_t) = N\left(\boldsymbol{\mu}, \sigma^2 \boldsymbol{I}\right) \quad (8)$$

where $\mu = (\mu_0, \mu_{1,\ldots,}\ \mu_{D-1})$ is the mean vector, D is the action dimension, σ is a standard deviation parameter, $\boldsymbol{I}$ is the identity matrix and thus policy parameter $\boldsymbol{\lambda} = (\boldsymbol{\mu}, \sigma)$.

When policy $\pi_{\boldsymbol{\lambda}}$ is decomposed into policy dimensions, $\pi_{\boldsymbol{\lambda},d}(\cdot|s_t) \sim N\left(\boldsymbol{\mu}_d, \sigma^2 \boldsymbol{I}\right)$. Assuming that $\boldsymbol{a}_{t,d}$ is the d-th element of $\boldsymbol{a}_t$, it can be drawn as follows:

$$\pi_{\boldsymbol{\lambda}}(\boldsymbol{a}_t|\boldsymbol{s}_t) = \prod\nolimits_{d=0}^{D-1} \pi_{\boldsymbol{\lambda},d}\left(\boldsymbol{a}_{t,d}|s_t\right) \quad (9)$$

It can be seen that $\pi_{\boldsymbol{\lambda}}$ grows exponentially with the increase of D, which leads to an excessive weighting of IS. In response to this problem, combining the advantages of the clipping mechanism, the IS weights of each dimension will be clipped separately, and the new IS weight function is proposed, as shown in (10):

$$\rho_t = \frac{\pi_{\boldsymbol{\lambda}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\boldsymbol{\lambda}_i}(\boldsymbol{a}_t|\boldsymbol{s}_t)} = \prod_{d=0}^{D-1} \frac{\pi_{\boldsymbol{\lambda},d}\left(\boldsymbol{a}_{t,d}|\boldsymbol{s}_t\right)}{\pi_{\boldsymbol{\lambda}_i,d}\left(\boldsymbol{a}_{t,d}|\boldsymbol{s}_t\right)} \quad (10)$$

In addition, an additional loss is proposed to prevent the IS weight from being too far from (6). The IS weights are constrained with a simple KL divergence:

$$J_{IS} = D_{KL} = \frac{1}{M}\sum_{m=0}^{M-1}\left(\prod_{d=0}^{D-1}\pi_{\boldsymbol{\lambda},d}\left(\boldsymbol{a}_{t,d}\,|\boldsymbol{s}_t\right)\ln\frac{\pi_{\boldsymbol{\lambda},d}\left(\boldsymbol{a}_{t,d}\,|\boldsymbol{s}_t\right)}{\pi_{\boldsymbol{\lambda}_i,d}\left(a_{t,d}\,|\boldsymbol{s}_t\right)}\right) \tag{11}$$

α_{IS} is set as an adaptive weighting factor to constrain divergence:

$$\begin{cases}\text{if } J_{IS} < J_{targ}/2, & \alpha_{IS} = \alpha_{IS}/2\\ \text{if } J_{IS} > J_{targ}\times 2, & \alpha_{IS} = \alpha_{IS}\times 2\end{cases} \tag{12}$$

Finally, the objective function is given as follows:

$$\hat{J}(\boldsymbol{\lambda}) = \frac{1}{M}\sum_{t=0}^{M-1}\left[\prod_{d=0}^{D-1}\min\left(\rho_{t,d}\hat{A}_t, clip(\rho_{t,d}, 1-\varepsilon, 1+\varepsilon)\right)\hat{A}_t\right] - \alpha_{IS}J_{IS} \tag{13}$$

Dimension-Wise clipping successfully solves the zero gradient problem of PPO. Proximal policy optimization with dimension-wise clipping (PPO-DWC) improves the learning efficiency of the algorithm and effectively reduces the disappearance of gradient. Algorithm 1 shows the complete iterative process.

Algorithm 1 PPO-DWC

```
Input: max iterations $L$, epochs $K$
Initialize:
Initialize weights of policy networks $\lambda_i$ ($i$ =1,2,3,4) and critic network
Initialize IS weighting factors $\alpha_{IS} = 1$, learning rate $\zeta$
Initialize replay buffer E
Load the quadrotor dynamic model
for $iteration = 1$ to $L$ do
  Randomly initialize states of quadrotor
  Load the desired states
  $\lambda_i \leftarrow \boldsymbol{\lambda}$
  Sample trajectory $B_i$ of size N from $\pi_{\boldsymbol{\lambda}_i}$
  Store $B_i$ in replay buffer E
  Compute advantage estimations $\hat{A}_t$ by using all off-policy trajectories $B_i, \ldots, B_{i-L+1}$ in E
  for $epoch = 1$ to $K$ do
    for each gradient step do
      Sample mini-batch size $M$ from the sample batches in E
      Compute the objective function $\widehat{J}(\boldsymbol{\lambda})$
      Update $\lambda \leftarrow \boldsymbol{\lambda} + \zeta\nabla_{\boldsymbol{\lambda}}\widehat{J}(\boldsymbol{\lambda})$
    end for
  end for
  Update $\alpha_{IS}$ as (8)
end for
```

B. PPO WITH PD STABILITY AUGMENTATION CONTROLLER (PPO-PD)

RL is to find the correct policy for an unstable system through constant trial and error. In the learning process, there are many invalid data in the random actions generated, which is not conducive to the rapid convergence of the agent to the optimal policy. In many cases in actual engineering, an SAS will be used to achieve the control requirements more quickly. For example, for a balance bike, without the feedback of the stability augmentation system, it is just a unicycle that is difficult to control. After the stability augmentation system is added, learning to ride a unicycle becomes much easier, which is also available for machine learning. Inspired by this, we can introduce the idea of SAS into the learning process of RL.

Before the RL training control policies, we can design a stability augmentation feedback controller to stabilize the equilibrium point first. For a nonlinear system (4), the original current action of RL is $\boldsymbol{a}_t$. Assuming that the target point is an unstable equilibrium point, our goal is to learn $\boldsymbol{a}_t$ to make $\boldsymbol{s}_t$ tend to 0, which is also the basic idea of RL training control policies. After integrating the stability augmentation feedback, we define

$$\boldsymbol{a}_t = \boldsymbol{k}(\boldsymbol{s}_t) + \boldsymbol{a'}_t \tag{14}$$

where $\boldsymbol{k}(\boldsymbol{s}_t)$ is the stability augmentation state feedback and $\boldsymbol{a'}_t$ is the action generated by RL. Substituting (14) into (4), we obtain the new environment, including a stability augmentation controller as

$$\boldsymbol{s}_{t+1} = \boldsymbol{f}(\boldsymbol{s}_t, \boldsymbol{k}(\boldsymbol{s}_t) + \boldsymbol{a'}_t) = \boldsymbol{f'}(\boldsymbol{s}_t, \boldsymbol{a'}_t) \tag{15}$$

Once the RL controller is trained for (15) to choose an action $\boldsymbol{a'}_t$, we can obtain the action $\boldsymbol{a}_t$ of the original environment (4) by using (14).

For the quadrotor UAV system, proportional differential (PD) controllers are usually designed in the roll, pitch, and yaw channels to ensure local stability of the attitude [38]–[40]. In this work, we use the following PD control as the stability augmentation feedback:

$$\begin{cases}\tau_{\phi,t} = -k_{P\phi}\phi - k_{D\phi}\dot{\phi}\\ \tau_{\theta,t} = -k_{P\theta}\theta - k_{D\theta}\dot{\theta}\\ \tau_{\psi,t} = -k_{P\psi}\psi - k_{D\psi}\dot{\psi}\end{cases} \tag{16}$$

where k_{Pi} and $k_{Di}(i = \phi, \theta, \psi)$ respectively represent the proportional and differential coefficients of the roll, pitch and yaw channel. Moreover, the total lift level of the quadrotor is assumed to be T. Then it can be obtained:

$$\boldsymbol{k}(\boldsymbol{s}_t) = \frac{1}{4}\begin{bmatrix} T + \tau_{\psi,t} - 2\tau_{\theta,t}\\ T - \tau_{\psi,t} - 2\tau_{\phi,t}\\ T + \tau_{\psi,t} + 2\tau_{\theta,t}\\ T - \tau_{\psi,t} + 2\tau_{\phi,t}\end{bmatrix} \tag{17}$$

It must be pointed out that the role of stability augmentation feedback is to construct a local convergence region for the original system. The goal of RL is to find a solution that can reach the convergence region. Using the stability augmentation controller could avoid a large number of trial and error in the training process, and improve the learning efficiency. Moreover, it can be used in combination with other RL algorithms.

On the other hand, compared with individual PD feedback, the RL algorithm will enhance control performance to deal with extreme conditions, such as state overshoot or actuator saturation. Algorithm 2 describes the learning process of an RL controller with stability augmentation.

Algorithm 2 PPO-PD

```
Input: max iterations $L$, epochs $K$, actors $N$, time steps $T$
Initialize:
Initialize weights of policy networks $\lambda_i (i = 1,2,3,4)$ and critic network
Load the quadrotor dynamic model
for $iteration = 1$ to $L$ do
  Randomly initialize states of quadrotor
  Load the desired states
    for $actor = 1$ to $N$ do
      for $time\ step = 1$ to $T$ do
        Calculate the stability augmentation feedback $k(s_t)$ with state $s_t$
        Run policy $\boldsymbol{\lambda}$ to generate RL gains $\boldsymbol{a}'_t$
        Record reward $r_t$ and the next state $\boldsymbol{s}_{t+1}$
      Store transition $(s_t, a_t, r_t, s_{t+1})$ into replay buffer
        Compute advantage estimations $\widehat{A}_t$
      end for
    for $epoch = 1$ to $K$ do
        Optimize the loss target with mini-batch size $M \le NT$
        then compute the objective function $\widehat{J}(\boldsymbol{\lambda})$
        Update $\lambda$ w.r.t $\widehat{J}(\boldsymbol{\lambda})$
    end for
end for
```

C. SYSTEM FRAMEWORK

In the learning process, the RL algorithm is required to stabilize the attitude while the quadrotor can be released from any attitude in the state space. Neural networks are used to receive the state of the quadrotor, provide the throttle level of each rotor, and seek the optimal policy in iterations.

1) PPO-DWC FRAMEWORK STRUCTURE

The network structure of the system is shown in Fig. 4. Two neural networks are used in the training of PPO-DWC, one is the critic neural network, and the other is the actor neural network with parameter λ. Four policy sub-networks with parameters λ_i $(i = 1, 2, 3, 4)$ compose the actor neural network. Their weights will be optimized during the training phase.

During training, the current state of the quadrotor will enter replay buffer $\mathbf{E}$ as a state vector $[\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$. Since PPO adopts batch training, after the actor network collects a batch of state vectors, its network parameters are copied to the old actor network. In the next batch of training, the four sub-networks continue to be trained and updated. At the same time, the parameters of the old actor network remain unchanged until copied by a new round of network parameters. After the policy update, the output of the old neural network will be dimensionally clipped to obtain $\pi_{\boldsymbol{\lambda}}$ and the IS weight function ρ_t as the input of the PPO-DWC operation.

For the critic network, the advantage values as its output will evaluate the quality of the measures taken to achieve these states. After updating by minimizing its parameter, the critic neural network also feeds the advantage value to the operation side to complete the entire update process of the actor network.

After updating the policy, the outputs of the four sub-networks are μ_i and σ_i $(i = 1, 2, 3, 4)$, which correspond to the four sets of mean and standard deviation of Gaussian distribution. A group of actions is randomly sampled from a Gaussian distribution and normalized to a_i $(i = 1, 2, 3, 4)$. a_i becomes the input of the quadrotor, and the quadrotor generates a new state.

2) PPO WITH STABILITY AUGMENTATION FRAMEWORK STRUCTURE

The system structures of PPO-PD and PPO-DWC are fundamentally different. PPO-DWC mainly analyzes the algorithm structure and clips the policy dimension to optimize the convergence of the optimal function, while PPO-PD uses the classical PPO algorithm and introduces a stability augmentation controller in the state observation stage to cooperate with RL to complete policy optimization, which is a synchronous process. The quadrotor will output the state to the stabilization module and the RL module respectively, and the input obtained is also the result of the combination of the stability augmentation feedback and the RL gain. The system framework is shown in Fig. 5.

For the attitude angle tracking task, our goal is to minimize the cumulative tracking error. In order to evaluate the performance of the quadrotor in terms of robustness, the reward signal is as simple as possible. Therefore, the reward function is given by:

$$r = -\alpha * \sqrt{\phi^2 + \theta^2 + \psi^2} - \beta * (\dot{\phi}^2 + \dot{\theta}^2 + \dot{\psi}^2) \qquad (18)$$

IV. SIMULATION AND RESULTS

In this section, the proposed improved PPO control policy is applied to the quadrotor UAV. Table 1 lists the model parameters. In order to fly safely, physical constraints should be imposed on the states of the quadrotor. The range of attitude angular velocity is set to ±258 °/s, which also meets the limitation of the gyroscope sensor. The range of attitude angle is set to ±45°. When the quadrotor's attitude exceeds 45° during training, it will be considered a bad training session, and the training round will be terminated early. We use Python to

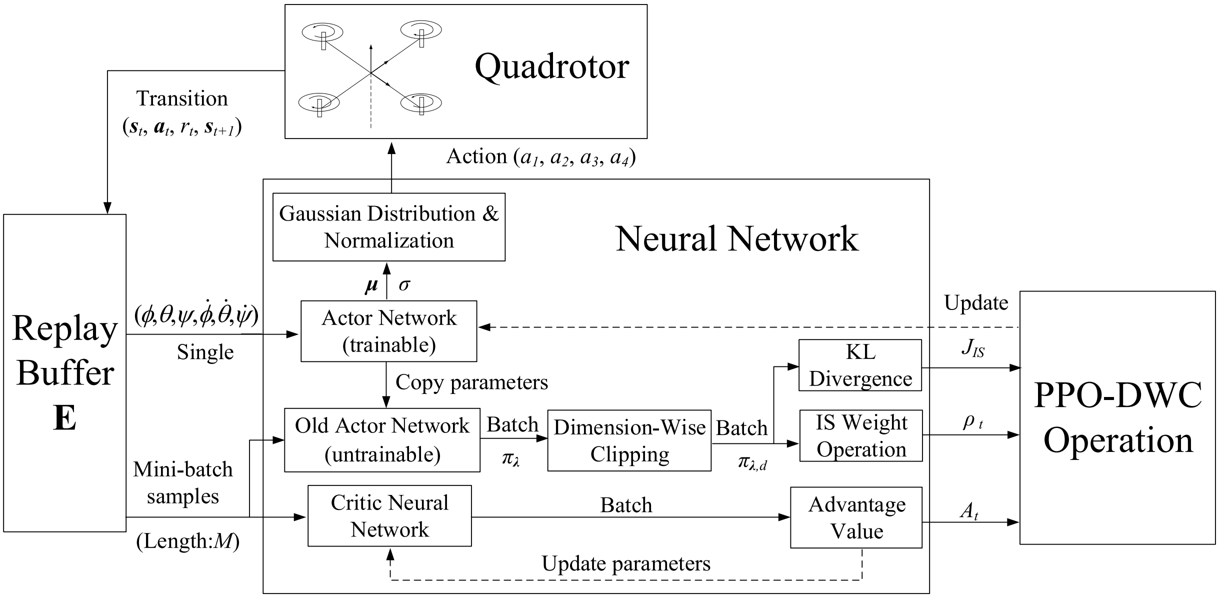


FIGURE 4. PPO-DWC system network framework structure.

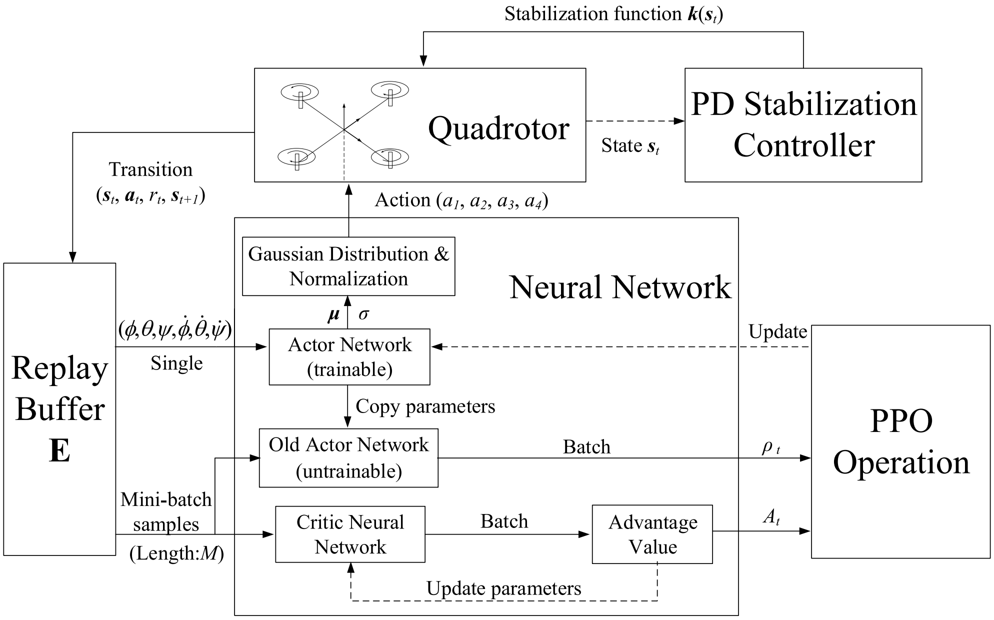

Stabilization function $k(s_t)$
Quadrotor
State s_t
PD Stabilization Controller
Transition
(s_t, a_t, r_t, s_{t+1})
Action (a_1, a_2, a_3, a_4)
Neural Network
Gaussian Distribution & Normalization
μ σ
Actor Network (trainable)
Update
Replay Buffer E
$(\phi,\theta,\psi,\dot{\phi},\dot{\theta},\dot{\psi})$
Single
Copy parameters
Old Actor Network (untrainable)
Batch
ρ_t
PPO Operation
Mini-batch samples
(Length:M)
Critic Neural Network
Batch
Advantage Value
A_t
Update parameters

FIGURE 5. PPO-PD system network framework structure.

develop the training simulation environment of the quadrotor. TensorFlow tools are utilized to build neural networks for learning and training [41]. Its library calls are computed on a laptop GPU (NVIDIA GeForce GTX 1650Ti). The simulations do not involve parallel computing techniques.

A. PPO-DWC ABILITY TEST

The control policies learned by the original PPO algorithm and the PPO-DWC algorithm are compared from offline training efficiency and control performance.

1) OFFLINE TRAINING

After defining the actor-critic network structure, Table 2 gives the training parameters of Algorithm 1 in the offline learning phase.

The training task is that the quadrotor can adjust to the desired attitude [0, 0, 0] in a randomly initialized state. Two indicators are used: average value loss and average accumulated rewards which are in a negative correlation to measure the learning effect. When training the quadrotor, the error should become smaller and smaller. In each step, the smaller

TABLE 1. **Training parameters.**

Parameter	Value
Reward discount factor γ	0.99
Actors N	4
Learning rate ζ	0.00025
Value function coefficient	0.01
Entropy coefficient	0.5
Batch size M	128
Maximum number of iterations L	1000
Simulation time per step	0.02 s
Reward function coefficients α	1
Reward function coefficients β	0.01

TABLE 2. **Parameters used in the simulator.**

Parameter	Description	Value
L	Arm length	0.31 m
b	Thrust gain	0.000572
K_ψ	Reaction torque gain	0.000172
I_p	Propeller moment of inertia	0.0073
I_x	X-axis moment of inertia	$0.008\ \mathrm{kg} \cdot \mathrm{m}^2$
I_y	Y-axis moment of inertia	$0.008\ \mathrm{kg} \cdot \mathrm{m}^2$
I_z	Z-axis moment of inertia	$0.03\ \mathrm{kg} \cdot \mathrm{m}^2$
d_x	X-axis air resistance coefficient	0.001
d_y	X-axis air resistance coefficient	0.001
d_z	X-axis air resistance coefficient	0.001

the error of the quadrotor attitude is, the larger the reward value is. A larger and more stable accumulated reward fully reflects a more accurate and faster control policy. In this study, we calculate the average value of each 50 groups of data, and evaluate the value loss and accumulated reward. PPO-DWC and the original PPO algorithm are used to train on the same network structure and training parameters, respectively. The comparison results are shown in Fig. 6. The training phase has a total of 6000 iterations. The total time cost of the computation of the two algorithms is almost the same because they are trained under the same network structure and training parameters, which takes 1932.4 s. It can be seen that PPO-DWC makes the value loss converge faster during training, which is due to its strong sampling efficiency. At the same time, ten independent simulations are carried out for the two algorithms. The shadow part in the figure represents the standard deviation of the ten simulation results. Obviously, the errors of the two algorithms are convergent. After a certain round of training, the original PPO algorithm still has errors, while the error of the PPO-DWC algorithm is smaller, and the convergence is faster.

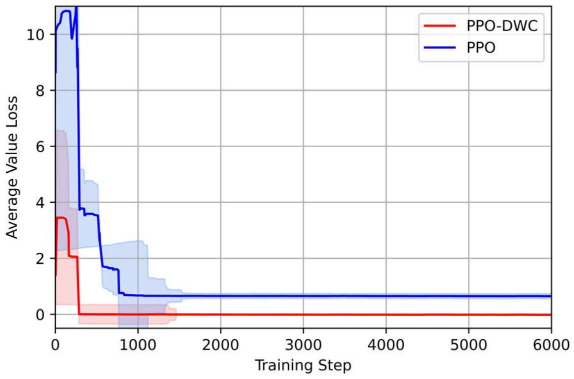


FIGURE 6. **Average value loss in the evaluation of policies learned by PPO-DWC and PPO.**

The average accumulated reward is shown in Fig. 7, which means that PPO-DWC has a higher convergence speed and higher reward. The PPO-DWC training progress stabilizes after about 500 training steps, taking only 152.6 s. While the original PPO algorithm converges after about 1500 steps, which takes 482.4 s. It can also be seen from the standard deviation that the PPO-DWC algorithm is consistent in the simulation.

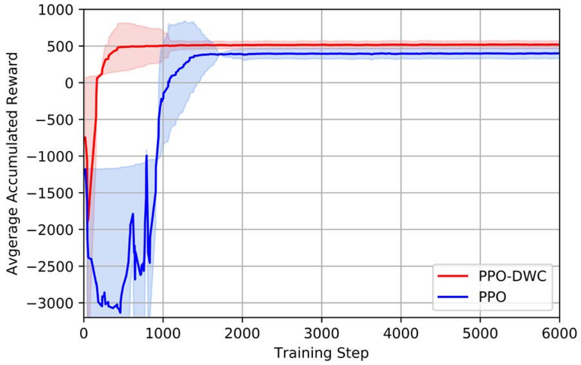


FIGURE 7. **Average accumulated reward in the evaluation of policies learned by PPO-DWC and PPO.**

After 6000 training iterations, we test the control policies learned through the PPO-DWC and original PPO algorithms in the environment of quadrotor rotational motion. The initial attitude of the quadrotor is $[-30, -20, -10]$ °, which is set within the safe range. The attitude angles of the quadrotors are recorded for 10 seconds. Fig. 8 shows the results of the two algorithms. It is noticeable that both algorithms can obtain convergent policies in the quadrotor attitude control task. PPO-DWC has higher accuracy.

In order to highlight the advantages of PPO-DWC, we compare the mean average error (MAE) of the attitude angles provided by the two algorithms. As shown in Fig.9, the control performance of PPO-DWC is better than the original PPO algorithm.

2) ROBUSTNESS TEST

In the offline learning phase, a stable robust control policy has been trained. In order to test its generalization ability, two different robustness tests are implemented. A traditional PID controller will also be added to make a comparison. Similar to PPO, PID also controls the target by trial and error. According to the error between the actual output of the quadrotor state and the desired command, the output is repeatedly adjusted to

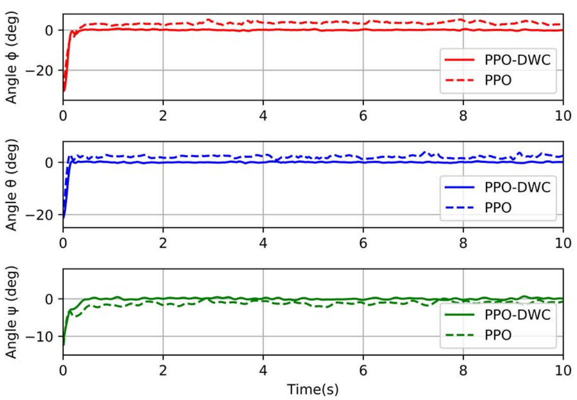


FIGURE 8. Attitude curves of control policies learned by PPO-DWC and PPO.

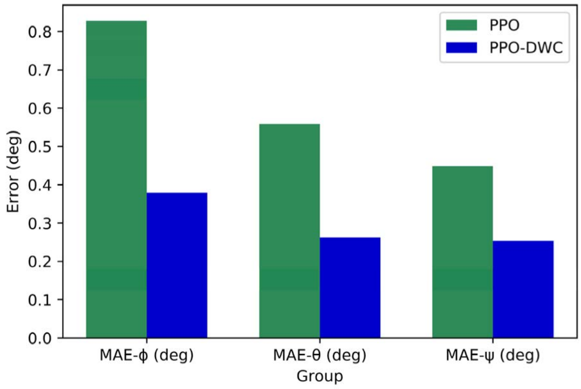


FIGURE 9. Mean average errors at the steady state of quadrotor using the control policies learned by PPO-DWC and PPO.

achieve the given desired value. Moreover, many other control algorithms are based on precise model dynamics, which is incomparable to the model-free PPO learning algorithm.

Case I: Model generalization test of different sizes. In this case, we change the distance from the rotor to the center of mass (the radius of the quadrotor) to test the generalization ability of the control policy. Assuming the initial flight attitude of the quadrotor is $[-15, -10, -10]$ ° and the desired attitude is $[0, 0, 0]$ °. The attitude changes of the quadrotor are observed during 10 seconds of flight through three control policies as PID, original PPO and PPO-DWC. In addition, the sum of the absolute error of three attitude angles is introduced to demonstrate the dynamic performance of the three control policies. A smaller sum of error means a faster control policy and higher accuracy.

We assume that the radius of the quadrotor model in the offline learning phase is 0.31m as the standard radius. In the robustness test, the mass of the quadrotor model remains unchanged. We change the radius range from 0.1m (65% smaller) to 1.2m (300% larger). A total of 12 simulations are carried out. The change of the attitude angle under the three control policies is shown in Fig. 10. When the radius is in 0.31m∼0.6m, the three control policies can make the quadrotor reach the steady state very well. However, when the radius gradually increases, the PID controller becomes unstable. Compared with the previous steady state, the quadrotor based on the PID controller begins to oscillate violently, and the error between the real attitude and the desired attitude becomes larger. The result also suggests that the quadrotor model performs more consistently under the RL control policies. The control performance is better than the PID control, which fully reflects the good generalization ability of RL. The result presented by the PPO-DWC policy and the original PPO policy indicates that the steady-state error of PPO-DWC is significantly smaller than that of the original PPO under the same number of training steps. The model controlled by PPO-DWC policy can be quickly and accurately stabilized, demonstrating its high efficiency in learning control tasks in complex environments. Its improvement over the baseline PPO is quite apparent.

It can also be verified in Fig. 11 that in the test model set, the control policy learned in the PPO-DWC algorithm has the slightest influence on the response of attitude tracking. With the increase of radius, the control performance of RL and PID controller degrade, but the decrease of PID control performance is more prominent. The steady-state error of PPO-DWC is the smallest. Moreover, it can be found that the control policy has better performance in a small quadrotor. The rapidness of the aircraft is benefited from the slight aerodynamic drag and moment of inertia of the smaller quadrotor.

Case II: Model generalization test under random initial state. Simulations under different initial states of the quadrotor are conducted to observe the control efficiency of the PPO-DWC algorithm. The control task is to adjust the quadrotor from a random initial attitude within a safe range to a desired steady state $[0, 0, 0]°$. A total of 20 simulations are carried out. We observe and record the attitude changes of the quadrotor in the 10 seconds of flight. The results are shown in Fig. 12. It can be seen that when the quadrotor starts to operate from different initial attitudes, the PPO-DWC control strategy can effectively make the quadrotor reach a stable state with a small steady-state error. This demonstrates the good performance of the PPO-DWC offline policy.

B. PPO-PD CONTROLLER ABILITY TEST

In this test, the same quadrotor model and neural network parameters are used as in **Subsection *A*** to observe the performance of the PPO algorithm with the PD stability augmentation controller from the two aspects of learning efficiency and control performance.

1) OFFLINE TRAINING

Before the RL control policy training, three groups of PD parameters as the stability augmentation controller of the system are set, which are shown in Table 3. In order to observe the effect of the PD stability augmentation controller in the RL training stage, the three groups of parameters given are also representative.

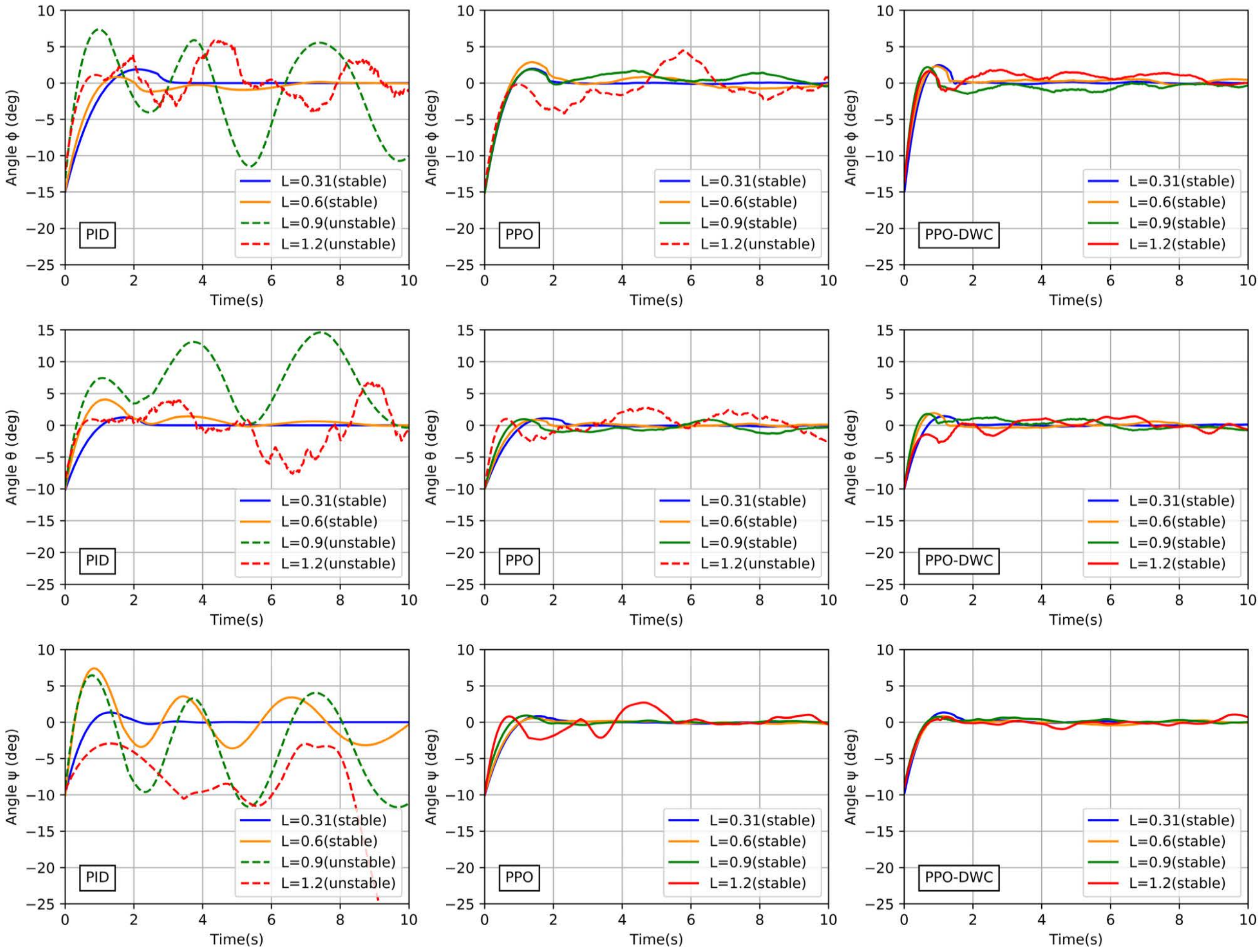


FIGURE 10. Testing results of case I. Comparison of the control performance with PID controller, PPO algorithm and PPO-DWC algorithm on quadrotor models in different sizes.

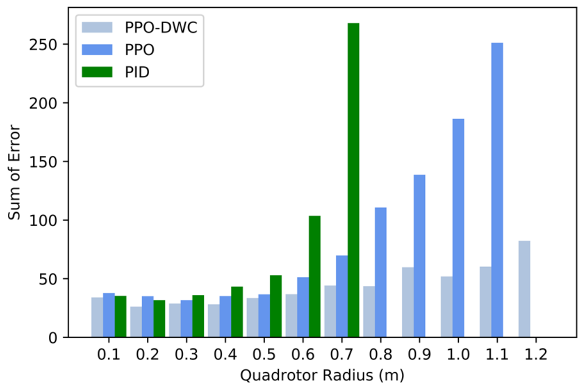


FIGURE 11. Testing results of case I. Sum of error in different sizes with the PPO-DWC algorithm, PPO algorithm and PID controller.

In the first group of PD parameters, K_{Di} $(i = 1, 2, 3) = 0$, the stability augmentation controller is only proportional control. Without RL gain, the system is in an unstable

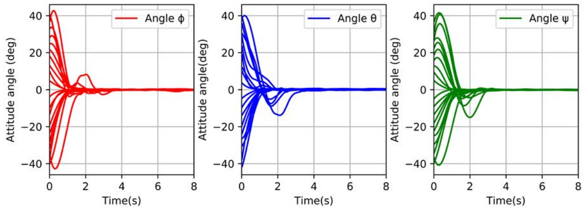


FIGURE 12. Testing results of case II. PPO-DWC control performance test in 20 different initial states.

state. On this basis, the proportional feedback can speed up the adjustment process, quickly respond to the command, and reduce the steady-state error. The introduction of the K_D parameter in the second group of parameters improves the stability of the system, accelerates the dynamic response speed of the system, and reduces the adjustment time. At the same time it reduces the overshoot and overcomes the oscillation, thereby improving the dynamic performance of the system. The third group of parameters is based on the second group

TABLE 3. PD controller parameters.

Parameter	K_{P1}	K_{P2}	K_{P3}	K_{D1}	K_{D2}	K_{D3}
I	-2	-2	-3	0	0	0
II	-2	-2	-3	-0.2	-0.2	-0.3
III	-2	-2	-3	-0.25	-0.25	-0.35

of parameters, which is to modify K_D parameters according to the control effect of the quadrotor system. The system can be stabilized without RL gain, and it is also a relatively good group of the three data groups.

In this subsection, the average loss and average accumulated reward are also used as the standard to measure the learning effect. The larger the reward value, the smaller the error between the attitude and the desired state at each step. We also calculate the average value of each 50 groups of data as a set of samples to compare the learning algorithm's value loss and accumulated reward after introducing the stability augmentation controller. The final comparison of the average value loss is shown in Fig. 13. Overall, the value loss of PPO with PD stability augmentation controller is minimal at the beginning of training and finally tends to be stable. It indicates that stability augmentation feedback is beneficial for the training efficiency of RL. By comparing the final convergence states of the four algorithms, it can be seen that the PPO algorithm with PD parameter III has the least value loss. It also shows that when adjusting the PD parameter, the more stable the PD parameter of the system is, the more beneficial the RL is to learn the desired control policy.

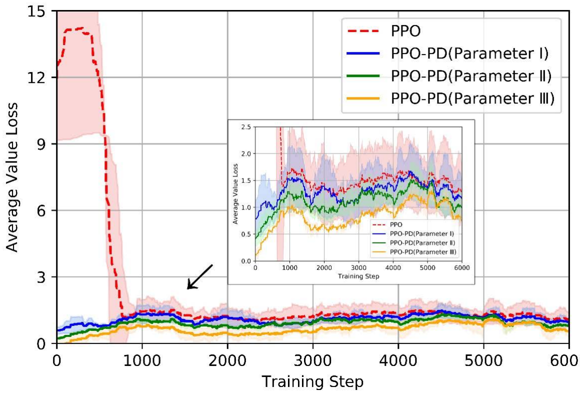


FIGURE 13. Average value loss in the evaluation of policies learned by PPO and PPO-PD.

The average accumulated reward is shown in Fig. 14, which is negatively correlated with the value loss. It is worth noting that the PPO algorithm with stability augmentation controller all converges the policy in about 350 steps, which takes about 105s, less than the time of the original PPO algorithm. The average accumulated reward of PPO-PD (Parameter I) under the final convergence is similar to the original PPO algorithm. This is because the quadrotor itself is still unstable under proportional control. Ultimately the quadrotor reaches equilibrium mainly depending on the effect of RL gain. However, due to the role of proportional adjustment in the initial training, the system avoids a large number of random trials and errors, so the error of the initial accumulated reward is far less than that of the original PPO algorithm. PPO-PD with parameter II can obtain a higher reward value after adding the K_D parameter, which is better than the PPO algorithm and PPO-PD with Parameter I algorithm. As shown in the PPO-PD result with Parameter III, the more stable the PD parameter makes the system, the more reward it gets under RL gain.

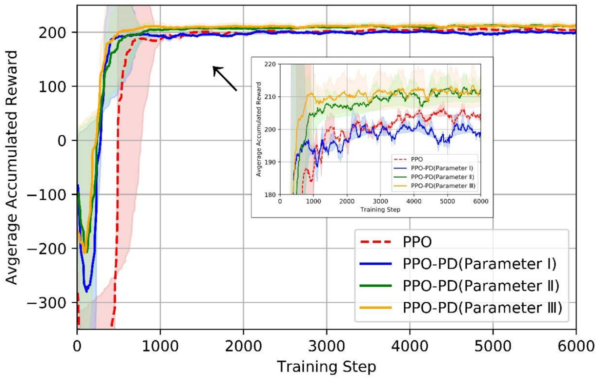


FIGURE 14. Average accumulated reward in the evaluation of policies learned by PPO and PPO-PD.

2) STABILITY TEST

Using the three groups of parameters listed in Table 3, the four RL algorithms have generated the control policies of the quadrotor in the offline training phase. In order to carry out the stability test, we set the initial flight attitude of the quadrotor as $[-30, -30, -30]$ °, which is relatively difficult to reach the desired attitude. The desired attitude is still $[0, 0, 0]$ °. The model parameters of the quadrotor are shown in Table 1. The attitude changes of the quadrotor are observed under the same model flying for 8 seconds. The comparison results of attitude changes under the four control policies are shown in Fig. 15.

When using the first group of PD parameters, since K_{Di} $(i = 1, 2, 3) = 0$, the stability augmentation controller is only proportional control. It can be observed that without RL gain, the proportional control cannot make the quadrotor stable. After introducing RL gain, the quadrotor can be stabilized under Parameter I. Because the proportional adjustment accelerates the response speed of the system, the quadrotor can reach the desired attitude more quickly. Therefore, its control performance is obviously better than the original PPO control policy. The differentiation element is introduced when the second group of PD parameters is used. The PD stabilization controller can only make the roll and pitch angles reach the desired state. The yaw angle cannot

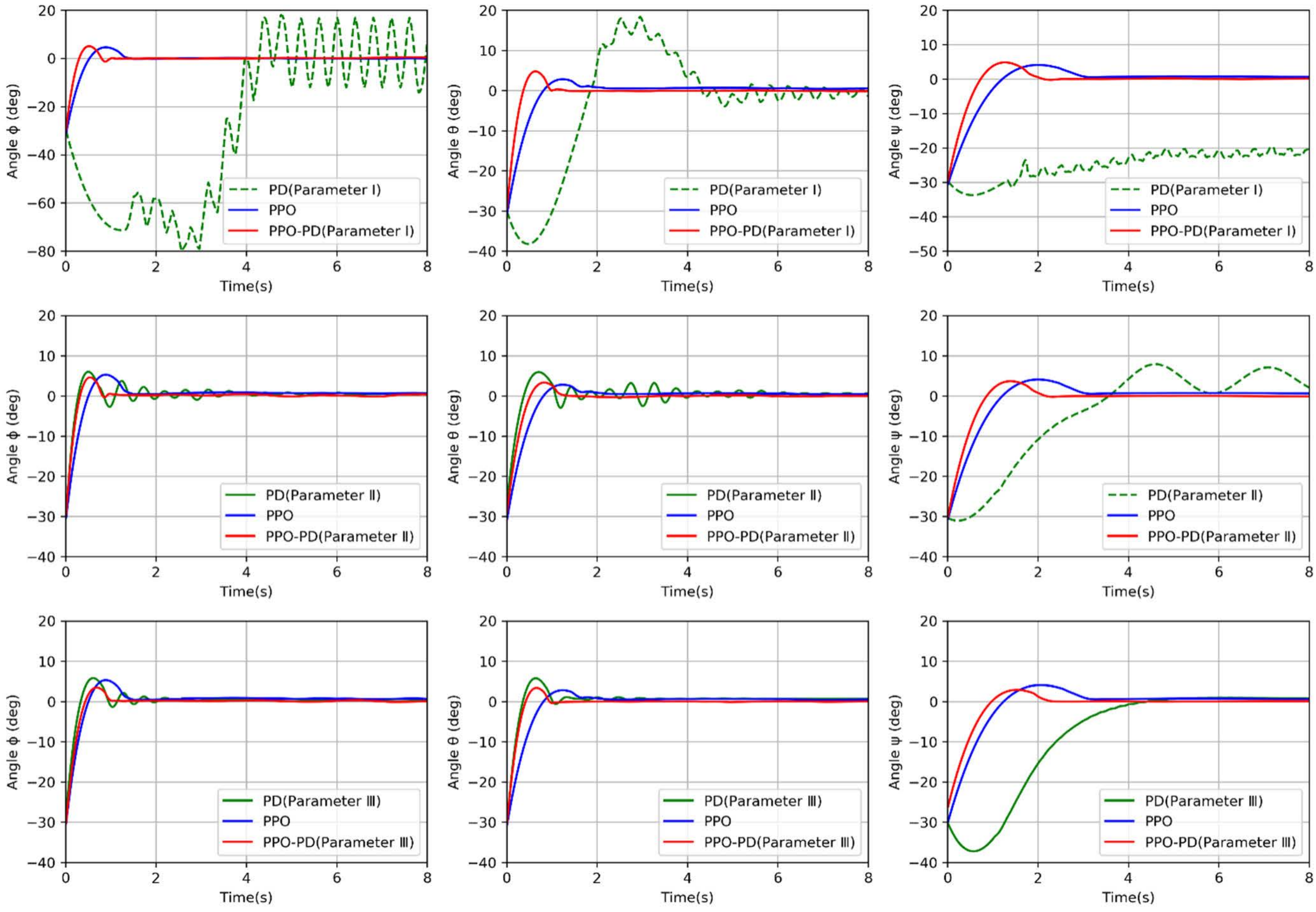


FIGURE 15. Results of stability test. Comparison of the control performance with PD controller, PPO algorithm and PPO-PD algorithm on the same quadrotor models.

reach the desired attitude and slightly oscillates. After the RL gain is added, it suggests that the quadrotor can reach the desired attitude faster than the original PPO algorithm, and the correction amplitude during flight is significantly reduced. This is due to the differentiation control increasing the damping of the system and improving the system's stability.

According to the flight results of the quadrotor under the second group of PD parameters, we slightly adjust the parameters, and thus obtain the third group of PD parameters. The quadrotor can achieve a steady state without RL gain through the stability augmentation controller under this parameter. The control policy trained by PPO with this stability augmentation controller can converge to the equilibrium point faster, and the error will be smaller.

In order to intuitively compare the importance of PD parameter change for PPO control policy, we compare the MAE between the attitude angle and the desired attitude provided by PPO-PD algorithms with three different parameters. As shown in Fig. 16, the more stable the PD control can achieve, the better performance the PPO-PD can obtain.

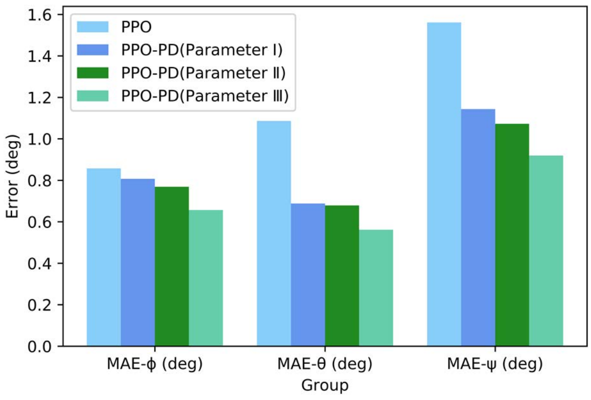


FIGURE 16. Angle errors at the steady state of quadrotor using the control policies learned by PPO and PPO-PD.

Since the parameter selection of PD control is very tedious work, it is difficult to make the system reach a steady state in a complex environment. After introducing RL gain, not only can the system be stabilized expediently, but also the control accuracy and learning efficiency are superior to ordinary RL.

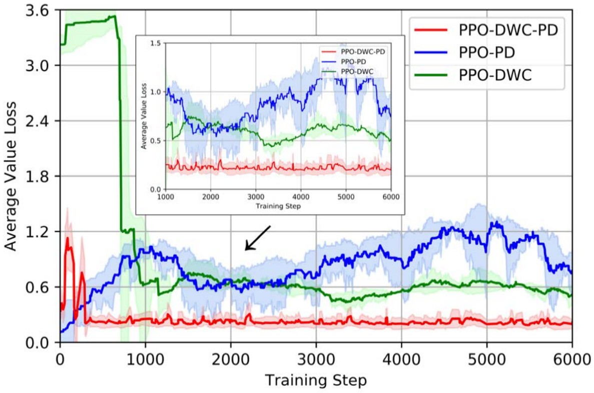


FIGURE 17. Average value loss in the evaluation of policies learned by PPO-DWC-PD, PPO-PD and PPO-DWC.

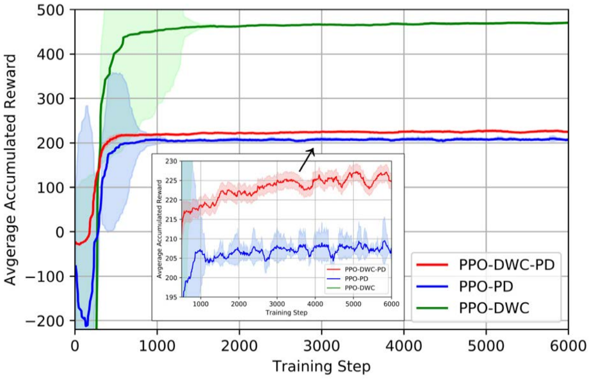


FIGURE 18. Average accumulated reward in the evaluation of policies learned by PPO-DWC-PD, PPO-PD and PPO-DWC.

C. PERFORMANCE COMPARISON OF PPO-DWC, PPO-PD, AND PPO-DWC-PD

PPO-DWC and PPO-PD improve the PPO from two different levels. PPO-DWC aims to change the structure of the algorithm to solve the problem of vanishing gradients. The sample exploration of PPO is extended to converge to the desired policy quickly while PPO-PD introduces a stability augmentation controller outside the RL policy. An accurate PD parameter can speed up the training time, thereby affecting the convergence of the policy. In this subsection, we combine the two algorithms. The stability augmentation controller uses Parameter III in Table 3. The DWC-based PPO with stability augmentation controller (PPO-DWC-PD) is brought into the same network parameters and quadrotor model parameters for training. We compare the results with PPO-DWC and PPO-PD.

1) OFFLINE TRAINING

The result comparison of the average value loss of the algorithms is shown in Figure 17. In general, PPO-DWC-PD has the most remarkable ability of quick response and the highest learning efficiency. It combines the advantages of the other two algorithms. It has both the ability of PPO-PD to converge quickly, and the ability of PPO-DWC to obtain smaller value loss under efficient exploration.

Figure 18 shows the average accumulated reward of the three algorithms, demonstrating that the learning efficiency of PPO-DWC-PD is higher than that of PPO-PD. By observing the convergence of the three algorithms, it can be concluded that PPO-DWC-PD is the fastest algorithm to obtain the maximum reward value. PPO-DWC gets the most reward because the weight of RL gain in the system has changed after the introduction of the PD controller.

2) STABILITY TEST

To further observe the control performance of PPO-DWC-PD, a worst case is selected to increase the complexity of the task. Assuming that the distance from the quadrotor rotor to the center of mass L is 1.5. The initial attitude of the quadrotor is $[-40^\circ, -40^\circ, -40^\circ]$, and the desired attitude is $[0^\circ, 0^\circ, 0^\circ]$. The final comparison result is shown in Figure 19.

The quadrotor under the PPO-PD control policy still oscillates slightly in the steady state because the change of the model parameters greatly influences the stability augmentation controller. The PPO-DWC control policy has good robustness and produces a relatively slight steady-state

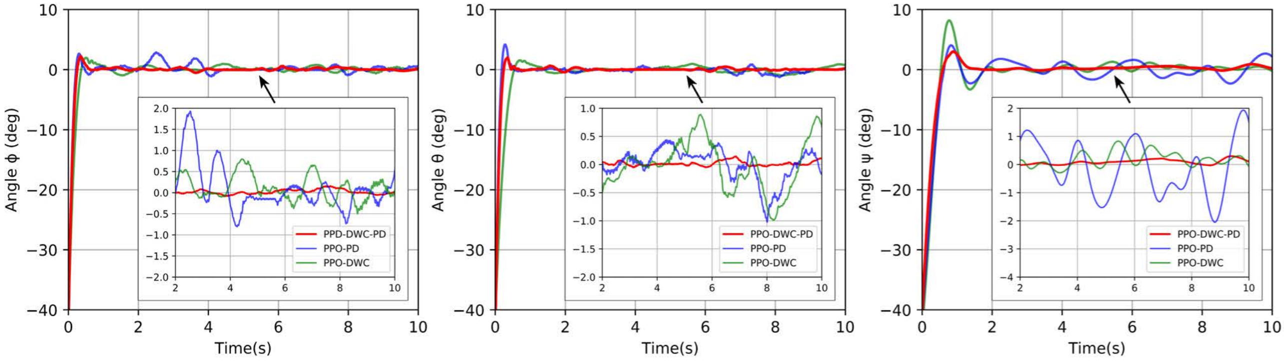


FIGURE 19. Comparison of the control performance with PPO-DWC-PD algorithm, PPO-PD algorithm and PPO-DWC algorithm on the same quadrotor models.

deviation. In contrast, the control performance of the PPO-DWC-PD policy is the fastest to converge to the steady state among the three policies, and the most stable when reaching the steady state.

V. CONCLUSION

In this paper, an improved PPO algorithm based on PPO-DWC and PPO-PD is proposed to solve the continuous motion control of the quadrotor. This is a learning-based control policy, which significantly improves the flight accuracy and reduces the steady-state error of the quadrotor attitude control. The PPO algorithm is improved in two directions. One is to optimize the algorithm structure of the PPO. For the problem of the disappearance of the sample gradient in the PPO algorithm, the dimension clipping method is used to calculate the policy function, which successfully improves the sample efficiency and the convergence of the quadrotor control policy is faster. Second, a stability augmentation controller is introduced to avoid blind exploration of the quadrotor in the initial stage. The PPO output is used as a gain term to enhance the stability of the quadrotor. The simulation results show that the control policy has good robustness and the performance of the new algorithm is better than original PPO and PID controller. In future work, we will focus on analyzing and processing the old sample batches in the PPO algorithm to enhance learning efficiency, and combine the compound reward function signal to reduce the observed steady-state error. Moreover, a more complex nonlinear stability augmentation feedback will also be considered.

REFERENCES

[1] L. Conegundes and A. C. M. Pereira, "Beating the stock market with a deep reinforcement learning day trading system," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[2] Y. Liu, H. Wang, M. Peng, J. Guan, J. Xu, and Y. Wang, "DeePGA: A privacy-preserving data aggregation game in crowdsensing via deep reinforcement learning," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4113–4127, May 2020.

[3] F. Rasheed, K.-L. A. Yau, R. M. Noor, C. Wu, and Y.-C. Low, "Deep reinforcement learning for traffic signal control: A review," *IEEE Access*, vol. 8, pp. 208016–208044, 2020.

[4] D. Chen, Q. Qi, Z. Zhuang, J. Wang, J. Liao, and Z. Han, "Mean field deep reinforcement learning for fair and efficient UAV control," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 813–828, Jan. 2021.

[5] A. Gonzalez-Garcia, D. Barragan-Alcantar, I. Collado-Gonzalez, and L. Garrido, "Adaptive dynamic programming and deep reinforcement learning for the control of an unmanned surface vehicle: Experimental results," *Control Eng. Pract.*, vol. 111, Jun. 2021, Art. no. 104807.

[6] Z.-Q. Su, M. Zhou, F.-F. Han, Y.-W. Zhu, D.-L. Song, and T.-T. Guo, "Attitude control of underwater glider combined reinforcement learning with active disturbance rejection control," *J. Mar. Sci. Technol.*, vol. 24, no. 3, pp. 686–704, Sep. 2019.

[7] B. Chen and X. Miao, "Distribution line pole detection and counting based on YOLO using UAV inspection line video," *J. Electr. Eng. Technol.*, vol. 15, no. 1, pp. 441–448, Jan. 2020.

[8] H. Shao, P. Song, B. Mu, G. Tian, Q. Chen, R. He, and G. Kim, "Assessing city-scale green roof development potential using unmanned aerial vehicle (UAV) imagery," *Urban Forestry Urban Greening*, vol. 57, Jan. 2021, Art. no. 126954.

[9] H. Zhang, L. Wang, T. Tian, and J. Yin, "A review of unmanned aerial vehicle low-altitude remote sensing (UAV-LARS) use in agricultural monitoring in China," *Remote Sens.*, vol. 13, no. 6, p. 1221, Mar. 2021.

[10] H. Liu, J. Ge, Y. Wang, J. Li, K. Ding, Z. Zhang, Z. Guo, W. Li, and J. Lan, "Multi-UAV optimal mission assignment and path planning for disaster rescue using adaptive genetic algorithm and improved artificial bee colony method," *Actuators*, vol. 11, no. 1, p. 4, Dec. 2021.

[11] F. Santoso, M. A. Garratt, and S. G. Anavatti, "State-of-the-art intelligent flight control systems in unmanned aerial vehicles," *IEEE Trans. Automat. Sci. Eng.*, vol. 15, no. 2, pp. 613–627, Apr. 2018.

[12] L. Xu, X. Shao, and W. Zhang, "USDE-based continuous sliding mode control for quadrotor attitude regulation: Method and application," *IEEE Access*, vol. 9, pp. 64153–64164, 2021.

[13] P. Tang, F. Zhang, J. Ye, and D. Lin, "An integral TSMC-based adaptive fault-tolerant control for quadrotor with external disturbances and parametric uncertainties," *Aerosp. Sci. Technol.*, vol. 109, Feb. 2021, Art. no. 106415.

[14] D. D. Dhadekar, P. D. Sanghani, K. K. Mangrulkar, and S. E. Talole, "Robust control of quadrotor using uncertainty and disturbance estimation," *J. Intell. Robot. Syst.*, vol. 101, no. 3, pp. 1–21, Mar. 2021.

[15] A. A. Najm and I. K. Ibraheem, "Altitude and attitude stabilization of UAV quadrotor system using improved active disturbance rejection control," *Arabian J. Sci. Eng.*, vol. 45, no. 3, pp. 1985–1999, Feb. 2020.

[16] D. Wang, Q. Pan, Y. Shi, J. Hu, and C. Zhao, "Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment," *IEEE Trans. Cybern.*, vol. 51, no. 10, pp. 5057–5068, Oct. 2021.

[17] S. L. Waslander, G. M. Hoffmann, J. Soon Jang, and C. J. Tomlin, "Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 3712–3717.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[19] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2096–2103, Oct. 2017.

[20] R. Polvara, M. Patacchiola, S. Sharma, J. Wan, A. Manning, R. Sutton, and A. Cangelosi, "Toward end-to-end control for UAV autonomous landing via deep reinforcement learning," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2018, pp. 115–123.

[21] C.-H. Pi, K.-C. Hu, S. Cheng, and I.-C. Wu, "Low-level autonomous control and tracking of quadrotor using reinforcement learning," *Control Eng. Pract.*, vol. 95, Feb. 2020, Art. no. 104222.

[22] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.

[23] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1587–1596.

[24] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[25] O. Elhaki and K. Shojaei, "A novel model-free robust saturated reinforcement learning-based controller for quadrotors guaranteeing prescribed transient and steady state performance," *Aerosp. Sci. Technol.*, vol. 119, Dec. 2021, Art. no. 107128.

[26] T. Chaffre, J. Moras, A. Chan-Hon-Tong, J. Marzat, K. Sammut, G. L. Chenadec, and B. Clement, "Learning-based vs model-free adaptive control of a MAV under wind gust," in *Informatics in Control, Automation and Robotics*. Paris, France: Springer, Jul. 2020, pp. 362–385.

[27] R. Polvara, M. Patacchiola, M. Hanheide, and G. Neumann, "Sim-to-real quadrotor landing via sequential deep Q-networks and domain randomization," *Robotics*, vol. 9, no. 1, p. 8, Feb. 2020.

[28] A. Rodriguez-Ramos, C. Sampedro, H. Bavle, P. De La Puente, and P. Campoy, "A deep reinforcement learning strategy for UAV autonomous landing on a moving platform," *J. Intell. Robot. Syst.*, vol. 93, nos. 1–2, pp. 351–366, Feb. 2019.

[29] G. C. Lopes, M. Ferreira, A. da Silva Simoes, and E. L. Colombini, "Intelligent control of a quadrotor with proximal policy optimization reinforcement learning," in *Proc. Latin Amer. Robot. Symp., Brazilian Symp. Robot. (SBR) Workshop Robot. Educ. (WRE)*, Nov. 2018, pp. 503–508.

[30] J. Yoo, D. Jang, H. J. Kim, and K. H. Johansson, "Hybrid reinforcement learning control for a micro quadrotor flight," *IEEE Control Syst. Lett.*, vol. 5, no. 2, pp. 505–510, Apr. 2021.

[31] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 528–535.

[32] Y. Wang, J. Sun, H. He, and C. Sun, "Deterministic policy gradient with integral compensator for robust quadrotor control," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 10, pp. 3713–3725, Oct. 2020.

[33] Z. Qingqing, T. Renjie, G. Siyuan, and Z. Weizhong, "A PID gain adjustment scheme based on reinforcement learning algorithm for a quadrotor," in *Proc. 39th Chin. Control Conf. (CCC)*, Jul. 2020, pp. 6756–6761.

[34] W. Koch, R. Mancuso, R. West, and A. Bestavros, "Reinforcement learning for UAV attitude control," *ACM Trans. Cyber-Phys. Syst.*, vol. 3, no. 2, pp. 1–21, Feb. 2019.

[35] K. W. Cobbe, J. Hilton, O. Klimov, and J. Schulman, "Phasic policy gradient," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 2020–2027.

[36] T. Kobayashi, "Proximal policy optimization with relative Pearson divergence," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 8416–8421.

[37] S. Han and Y. Sung, "Dimension-wise importance sampling weight clipping for sample-efficient reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2586–2595.

[38] J. Zhou, R. Deng, Z. Shi, and Y. Zhong, "Robust cascade PID attitude control of quadrotor helicopters subject to wind disturbance," in *Proc. 36th Chin. Control Conf. (CCC)*, Jul. 2017, pp. 6558–6563.

[39] H. Gao, C. Liu, D. Guo, and J. Liu, "Fuzzy adaptive PD control for quadrotor helicopter," in *Proc. IEEE Int. Conf. Cyber Technol. Autom., Control, Intell. Syst. (CYBER)*, Jun. 2015, pp. 281–286.

[40] K. Zheng and J. Wang, "Parameter optimization control of quadrotor aircrafts based on PD controller," *J. Hangzhou Dianzi Univ. (Natural Sci.)*, vol. 39, no. 4, pp. 58–65, Jul. 2019.

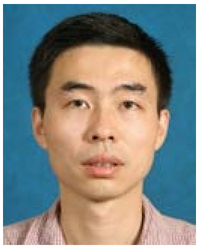
[41] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Des. Implement. (OSDI)*, 2016, pp. 265–283.

HANGXING WU received the B.S. degree in electrical engineering and automation from the School of Intelligent Manufacturing, Taizhou College, Nanjing University of Science and Technology, Taizhou, China, in 2015. He is currently pursuing the M.S. degree in control engineering with the School of Electronic Information, Jiangsu University of Science and Technology.

His current research interests include quadrotor unmanned aerial vehicle, reinforcement learning, and multi-agent control systems.

HUI YE was born in Zhenjiang, China, in 1986. He received the B.S. degree in flight vehicle propulsion engineering and the Ph.D. degree in control theory and control engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2007 and 2016, respectively.

He is currently an Associate Professor with the School of Electronics and Information, Jiangsu University of Science and Technology (JUST), Zhenjiang. His current research interests include nonlinear control systems, reinforcement learning, flight control, and underwater vehicle control.



WENTAO XUE received the M.S. degree in mechanical manufacturing and automation and the Ph.D. degree in control science and engineering from the Nanjing University of Science and Technology (NJUST), Nanjing, China, in 2002 and 2008, respectively.

He was a Visiting Scholar at the School of Engineering, University of Liverpool, U.K., from 2012 to 2013. He is currently an Associate Professor with the School of Electronics and Information, Jiangsu University of Science and Technology (JUST), Zhenjiang, China. His current research interests include intelligent control, flight control, and pattern recognition.

XIAOFEI YANG (Member, IEEE) was born in Henan, China, in 1983. He received the B.E. degree in electronical information science and technology from the Nanjing University of Technology, China, in 2005, and the M.S. degree in circuits and systems and the Ph.D. degree in automatic control from the Nanjing University of Science and Technology, China, in 2007 and 2011, respectively.

He is currently an Associate Professor with the School of Electronics and Information, Jiangsu University of Science and Technology. His research interests include control theory for autonomous systems, the Internet of Things, and RF systems.