IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 4, 2019, accepted February 10, 2019, date of publication February 25, 2019, date of current version March 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2901295

Supplementary Reinforcement Learning Controller Designed for Quadrotor UAVs

XIAOBO LIN[1], YAO YU[1], AND CHANG-YIN SUN[2], (Member, IEEE)
[1]School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2]School of Automation, Southeast University, Nanjing 210096, China

Corresponding author: Yao Yu (yuyao@ustb.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant FRF-GF-17-B46, and in part by the National Natural Science Foundation of China under Grant 61520106009, Grant 61533008, and Grant 61703037.

ABSTRACT The control problem for quadrotor UAVs is difficult and challenging due to the complex nonlinear dynamics and ever-changing disturbances. In this paper, a supplementary controller based on reinforcement learning (RL) is proposed to improve the control performance of quadrotor UAVs. The proposed RL method is constructed by an actor-critic structure and some improved technologies, e.g., Q-learning, temporal difference, and experience replay. With the proposed method, the speed and stability of training can be improved greatly. On one hand, the supplementary controller can work together with the traditional controller online, which can guarantee the stability of the system. On the other hand, the model uncertainties and external disturbances could be restrained through online RL training. The Lyapunov theory is used to prove the convergence of the RL controller's weights theoretically. Finally, three simulations are provided to illustrate the effectiveness of the proposed controller.

INDEX TERMS Quadrotor, UAVs, reinforcement learning, ADP, control system.

I. INTRODUCTION

Recently, quadrotor UAVs have acquired much attention in many areas [1]–[4], especially in the domain of logistics and agriculture. One of the their advantages is low-cost and low fault rates, which is based on their simple mechanical structure. Another advantage that they enjoy abilities of Vertical Take-Off and Landing (VTOL), stable hovering and high maneuverability extend application range. However, high performance control for quadrotor UAVs is a challenge for the reason that the dynamic model of them is Multi-Input Multi-Output (MIMO), strong coupling, nonlinear and underactuated; they suffer from model-uncertainties and unknown external disturbances.

Many linear or nonlinear methods were designed for quadrotor UAVs with coupling and nonlinear dynamic. The linear methods were convenient to be applied, e.g. PID and Linear Quadratic Regulator (LQR).But the liner methods simplified many nonlinear specialties and could only guaranteed the convergence near the equilibrium [5], [6]. In order to get larger convergence range and higher performance, several nonlinear methods were developed, like Nonlinear Hierarchical Control Strategy (NHCS) [7], Back-Stepping (BS) [8] and Dynamic Surface Control (DSC) [9], [10]. The NHCS and BS were efficient solutions for control problems with high-order nonlinear dynamic. Then the DSC provided a improvement for the ''explosion of complexity'' problem of the BS. However, these methods did not consider the system uncertainties and external disturbance, therefore they would lose performance when the accurate model is hard to get.

As quadrotor UAVs equipped with complex nonlinear dynamic and often worked in strange environments, it was difficult to acquire accurate model. Therefore, developing disturbance-reject methods is necessary. The robust approaches were applied to control quadrotor UAVs, which could reject external disturbances and uncertainties of system parameters, e.g. Robust Control [11] and Robust Signal Compensator [12]. However, the robust methods often needed the boundary of disturbances during controller design. Compared with robust methods, the adaptive approaches approximated the uncertain parameters and disturbances on-line, which were widely applied on quadrotor UAVs, e.g. Adaptive Back-Propagating Trajectory Tracking Control [13], Adaptive Back-Stepping Control [14] and Adaptive Control with unknown Center of Mass (COM) [15]. Although these methods could limited the

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

system uncertainties and external disturbance in some case, a basic model of quadrotor UAVs was still necessary. What's more, the optimization for the synthetic performance index, e.g. a common value function ($\int_0^\infty e^2 + u^2 \,\mathrm{d}t$, e was error and u was control value), was not satisfied.

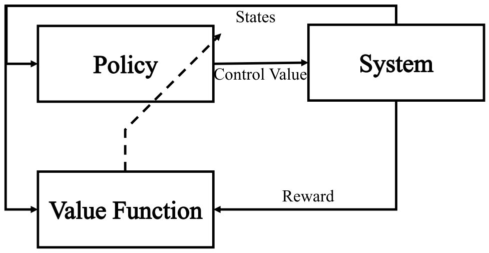


FIGURE 1. The scheme of common RL controller. There are two roles: the Policy is action used to generate control value; the Value Function is used to evaluate the performance.

With considering the above problems, the reinforcement learning (RL) controller was good candidate [16], [17]. A common RL scheme was shown in Fig. 1. The idea of RL control is to acquire a proper evaluation mechanism (called value function in follow), and then the value function is used to train the controller (called policy in follow) through experience. In virtue of powerful approximator (e.g. neural network, polynomial approximation and so on) and working mechanism, the model based RL controller could work well on many complex nonlinear systems [18], [19]. Although these conventional RL based method could satisfied common performance indexes, the full or partial knowledge of system dynamic was necessary, which limited its application. A data-based RL controller was presented to solve a class of unknown continuous-time nonlinear problems [20], [21], which was model-free and could reject system uncertainties and external disturbances. In recent year, a kind of goal-network was added to the RL structure for approximation reward function, e.g. Gr-ADP [22], [23]. The Gr-ADP not only provided more powerful reward signals, but also decreased the parameters need to be adjusted. According to whether the action information was necessary for value function, the RL controllers can be divided into action-independent ones and action-dependent ones. Compared with action-independent RL controllers, the action-dependent ones enjoyed smaller variance and higher convergence rate [24]. In addition, some other technologies were developed for improving the performance of RL controllers. A highlight one was the experience replay, which could decrease the correlations between training samples and increased the stabilities during training [25]. Considering the rapid developments of RL control and its advantages on solving high-dimension, bearing external disturbance, uncertain, coupling and nonlinear control problems, it was valuable to study its application on quadrotor UAVs.

Inspired by above discussion, in this paper, a supplementary RL controller is designed for quadrotor UAVs in this paper. The RL controller is based on actor-critic scheme, and its critic was action-dependent approximated with TD. Meanwhile, the experience replay and off-policy training technologies are applied for wider exploration, which increased robustness of controller. The advantages of the proposed including: 1) The method can run on-line with the actor-critic scheme and TD approximation. 2) Only a low-performance method rather than the system model is required for controller designed. It is a model free method and the disturbances can be rejected by training. 3) A performance index satisfying controller will be acquired after several episodes training. And the training process will be accelerated, because the original controller can restrain the state space in safe range. 4) The convergence is established by the Lyapunov method.

The rest of the paper is organized as follows. In section II, the dynamic of quadrotor is presented and the performance index is provided. In section III, the proposed reinforcement learning based supplementary method is detailed and the stability of the learning algorithm is analyzed. After that, the simulations shown the learning process and control performance in section IV. Finally, the conclusion is drawn.

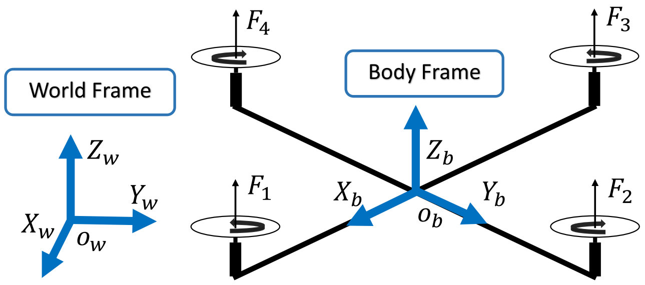


FIGURE 2. The structure of quadrotor UAVs. There are four rotors distributed in '+' shape. The $O_w - XYZ$ is the world frame and The $O_b - XYZ$ is the body frame.

II. PROBLEM DESCRIPTION

The structure of quadrotor UAVs studied in this paper is '+' type, seeing Fig. 2. There are two kinds of key state for quadrotor UAVs, position (X, Y, Z) and attitude (roll, pitch, yaw). The position describes the translation between world frame and body frame; the attitude describes rotation between these two frames. The dynamic of quadrotor UAVs is explained [26]:

$$\begin{aligned}
\ddot{p}_x &= [\cos\varphi \sin\theta \cos\psi + \sin\varphi \sin\psi]\frac{\tau_4}{m} + d_1 \\
\ddot{p}_y &= [\cos\varphi \sin\theta \sin\psi - \sin\varphi \cos\psi]\frac{\tau_4}{m} + d_2 \\
\ddot{p}_z &= \cos\theta cos\varphi\frac{\tau_4}{m} - g + d_3 \\
\ddot{\varphi} &= \dot{\varphi}\dot{\psi}(\frac{J_{zz} - J_{xx}}{J_{yy}}) + \frac{J_R}{J_{yy}}\dot{\varphi}\Omega_R + \frac{L}{J_{yy}}\tau_1 + d_4 \\
\ddot{\theta} &= \dot{\theta}\dot{\psi}(\frac{J_{yy} - J_{zz}}{J_{xx}}) - \frac{J_R}{J_{xx}}\dot{\theta}\Omega_R + \frac{L}{J_{xx}}\tau_2 + d_5 \\
\ddot{\psi} &= \dot{\theta}\dot{\varphi}(\frac{J_{xx} - J_{yy}}{J_{zz}}) + \frac{1}{J_{zz}}\tau_3 + d_6,
\end{aligned} \tag{1}$$

where p_x, p_y, p_z are the position and φ, θ, ψ are attitude of the quadrotor. The m is the mass and J_{xx}, J_{yy}, J_{zz} are the moments of inertia of the quadrotor. L is the length from the barycenter to the center of each rotor. The J_R and Ω_R donate the moments of inertia and angular velocity of the propeller blades. $d_1, \ldots, d_6$ are bounded disturbances. The τ_1, τ_2, τ_3 are thrusts in direction of roll, pitch, yaw and τ_4 is the collective thrust. The relationships between the thrusts $[\tau_1, \tau_2, \tau_3, \tau_4]^T$ and control signals $[u_1, u_2, u_3, u_4]^T$ for motors are shown as follow:

$$\begin{aligned} \tau_1 &= c_T(u_2 - u_4) \\ \tau_2 &= c_T(u_1 - u_3) \\ \tau_3 &= c_M(u_1 - u_2 + u_3 - u_4) \\ \tau_4 &= c_T(u_1 + u_2 + u_3 + u_4), \end{aligned} \tag{2}$$

where $[u_1, u_2, u_3, u_4]^T$ are control signals that range from 0 to 1. And c_T and c_M are the parameters which translate control signals to thrusts and torque.

In order to present the problem more conveniently, assume $|\theta| \leq \pi/2$, $|\varphi| \leq \pi/2$. Set $x_1 = \sin\varphi$, $x_2 = \dot{\varphi}$, $x_3 = \sin\theta$, $x_4 = \dot{\theta}$, $x_5 = \psi$, $x_6 = \dot{\psi}$, $x_7 = p_x$, $x_8 = \dot{p}_x$, $x_9 = p_y$, $x_{10} = \dot{p}_y$, $x_{11} = p_z$, $x_{12} = \dot{p}_z$. Then the system is translated into the follow form:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ f(x) &= \begin{bmatrix} \sqrt{1-x_1^2}x_2 \\ a_1x_4x_6 + d_4 \\ \sqrt{1-x_3^2}x_4 \\ a_2x_2x_6 + d_5 \\ x_6 \\ a_3x_2x_4 + d_6 \\ x_8 \\ d_1 \\ x_{10} \\ d_2 \\ x_{12} \\ -g + d_3 \end{bmatrix} \quad g(x) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & b_1 & 0 & -b_1 \\ 0 & 0 & 0 & 0 \\ b_2 & 0 & -b_2 & 0 \\ 0 & 0 & 0 & 0 \\ b_3 & -b_3 & b_3 & -b_3 \\ 0 & 0 & 0 & 0 \\ g'_X & g'_X & g'_X & g'_X \\ 0 & 0 & 0 & 0 \\ g'_Y & g'_Y & g'_Y & g'_Y \\ 0 & 0 & 0 & 0 \\ g'_Z & g'_Z & g'_Z & g'_Z \end{bmatrix} \\ u &= [u_1, u_2, u_3, u_4]^T \\ g'_X &= (\sqrt{1-x_1^2}x_3 \cos x_5 + x_1 \sin x_5)\frac{c_T}{m} \\ g'_Y &= (\sqrt{1-x_1^2}x_3 \sin x_5 + x_1 \cos x_5)\frac{c_T}{m} \\ g'_Z &= \frac{c_T\sqrt{1-x_1^2}\sqrt{1-x_3^2}}{m}, \end{aligned} \tag{3}$$

where $x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]^T$, $a_1 = \frac{J_{yy}-J_{zz}}{J_{xx}}$, $a_2 = \frac{J_{zz}-J_{xx}}{J_{yy}}$, $a_3 = \frac{J_{xx}-J_{yy}}{J_{zz}}$, $b_1 = \frac{L}{J_{xx}}$, $b_2 = \frac{L}{J_{yy}}$, $b_3 = \frac{1}{J_{zz}}$.

We assume that there exists an original controller (basic controller) $u_b(t)$ designed previously, which could stabilize the system. However, its performance index needs to be improved. Then a RL controller is developed as a supplementary to rise its performance. Note the RL controller with u_s, and the eventually controller could be shown as follow:

$$u(t) = u_b(t) + u_s(t). \tag{4}$$

The target of the eventually controller is not only to stabilize the original system, but also to optimize the performance index. The performance index used here is a long term index and is defined as follow:

$$V_{x\sim u_s(x)}(x(t)) = \int_t^{\infty} r(x(\tau))\,\mathrm{d}\tau, \tag{5}$$

where V the value function is the performance index and r is the instantaneous reward:

$$r(x(t)) = x(t)^T Wx(t) + u(t)^T Ru(t), \tag{6}$$

where W and R are a positive symmetric matrix. And the target of proposed method is to find the optimal controller u^* that minimum the optimal value function V^*.

III. SUPPLEMENTARY CONTROLLER BASED ON REINFORCEMENT LEARNING

Based on [23], the follow equations can be used to determine the optimal controller. Given an initial admissible policy u_0, these two steps are iterated until V_k and u_k converge.

(i) Policy evaluation

$$\begin{aligned} &\nabla V_k(x)(f(x) + g(x)u_{s,k}(x)) \\ &\qquad + x^T Wx + u_{s(k)}^T(x)Ru_{s,k}(x)) = 0 \\ &\qquad V_k(0) = 0. \end{aligned} \tag{7}$$

(ii) Policy improvement

$$u_{s,k+1}(x) = -1/2R^{-1}g^T(x)\nabla V_k(x)^T. \tag{8}$$

Although the above solution can acquire the optimal controller, it requires the knowledge of the system model $f(x)$ and $g(x)$. But the accurate model of quadrotor UAVs is hard to get. In order to overcome this limitation, another model-free bellman equation is shown as follow:

$$\begin{aligned} \dot{V}_k(x(t_k)) &= \nabla V_k(x)[f(x(t_k)) + g(x(t_k))u_s(t_k)] \\ &= \nabla V_k(x(t_k))[f(x(t_k)) + g(x(t_k))u_{s,k}(x(t_k))] \\ &\quad + \nabla V_k(x(t_k))g(x(t_k))(u_s(t_k) - u_{s,k}(x(t_k))) \\ &= -x^T(t_k)Wx(t_k) - u_{s,k}^T(x(t_k))Ru_{s,k}(x(t_k)) \\ &\quad - 2u_{s,k+1}^T(x(t_k))R(u_s(t_k) - u_{s,k}(x(t_k))), \end{aligned} \tag{9}$$

where t_k is the k^{th} sample instant. $u_s(t_k)$ represents the real control value in constant t_k. $u_{s,k}(x)$ notes the supplementary controller at step k and $u_{s,k+1}(x)$ notes the supplementary controller at step $k+1$.

Integrating both sides of (9) on time interval $[t_k, t_{k+1}]$, it follows that

$$\begin{aligned} &V_k(x(t_{k+1})) - V_k(x(t_k)) \\ &\quad = \int_{t_k}^{t_{k+1}} [-x(\tau)^T Wx(\tau) - u_{s,k}(x(\tau))^T Ru_{s,k}(x(\tau)) \\ &\qquad - 2u_{s,k+1}^T(x(\tau))R(u_s(\tau) - u_{s,k}(x(\tau)))]\,\mathrm{d}\tau. \end{aligned} \tag{10}$$

where the period during k and $k+1$ is fixed, noted with T.

This new equation contains no dynamic of the original system, then the following work focus on acquiring V^* and u_s^*. In order to apply a action-depended critic, a Q-function is defined:

$$Q_k(x(t_k), u_s(t_k)) = \int_{t_k}^{t_{k+1}} r(x(\tau), u_s(\tau)\,\mathrm{d}\tau + \int_{t_{k+1}}^{\infty} r(x(\tau), u_s(\tau))\,\mathrm{d}\tau, \quad (11)$$

where $Q_k(0,0) = 0$. It is found that $Q_{t_k}(x(t_k), u_s(t_k)) = V_{u_s}(x(t_k))$. Then the Q_{t_k} is rewritten as

$$\begin{aligned} Q_k(x(t_k), u_s(t_k)) &= \int_{t_k}^{t_{k+1}} r(x(\tau), u_s(\tau))\,\mathrm{d}\tau + Q_k(x(t_{k+1}, u_s(t_{k+1})) \\ &= \int_{t_k}^{t_{k+1}} r(x(\tau), u_s(\tau))\,\mathrm{d}\tau + V_k(x(t_{k+1})), \end{aligned} \quad (12)$$

where u_s is the control value. Define the optimal supplementary controller is u_s^*, then the associate optimal Q can be given by:

$$\begin{aligned} Q^*(x(t_k), u_s(t_k)) &= \int_{t_k}^{t_{k+1}} r(x(\tau), u_s(\tau))\,\mathrm{d}\tau + Q_{u^*}(x(t_{k+T}), u_s^*(t_{k+1})) \\ &= \int_{t_k}^{t_{k+1}} r(x(\tau), u_s(\tau))\,\mathrm{d}\tau + V^*(x(t_{k+1}))) \end{aligned} \quad (13)$$

Then,

$$\begin{aligned} Q^*(x(t_k), u_s^*(t_k)) &= \min_{u_s} Q^*(x(t_k), u_s(t_k)) \\ &= \min_{u_s} \int_{t_k}^{t_{k+1}} r(x(\tau), u_s(\tau)) + V^*(x(t_{t+1})) \\ &= \int_{t_k}^{t_{k+1}} r(x(\tau), u_s^*(\tau)) + V^*(x(t_{k+1})) \\ &= V^*(x(t_k)). \end{aligned} \quad (14)$$

According to the expressions (14), the supplementary controller $u_s^*(x)$ can be presented as

$$u_s^*(x) = \arg\min_{u_s} V_{u_s}(x) = \arg\min_{u_s} Q^*(x, u_s). \quad (15)$$

From above analysis, Q^* and V^* will equal when they converge. In the following design, the Q will be utilized to replace the V and the express (10) can be rewritten as:

$$\begin{aligned} Q_k(x(t_{k+1}), u_s(t_{k+1})) - Q_k(x(t_k), u_s(t_k)) &= \int_{t_k}^{t_{k+1}} [-x^T(\tau)Wx(\tau) - u_{s,k}^T(x(\tau))Ru_{s,k}(x(\tau)) \\ &\quad - 2u_{s,k+1}^T(x(\tau))R(u_s(\tau) - u_{s,k}(x(\tau)))]\,\mathrm{d}\tau. \end{aligned} \quad (16)$$

Two neural networks (NN) are used to express the Q-funciton and policy. The NN used to approximate Q-function is called as critic and the one used to approximate the policy is noted as actor. And the control scheme was shown in the Fig. 3.

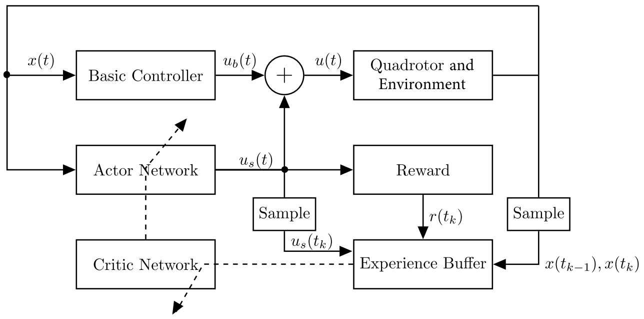


FIGURE 3. The control scheme of proposed controller. The system is divided into control subsystem and learning subsystem: the control subsystem is continues; the learning subsystem is discrete and updated at t_k.

According to the Weirstrass high-order approximation theorem, the critic and actor can be written with a compact set by a set of linearly independent basis functions.

$$Q_k(t_k) = \omega_c^T \phi_c(t_k) + \epsilon_c(t_k), \quad (17)$$

where ω_c represent the ideal coefficients, ϕ is the hidden layer neurons. ϵ_c is the approximation error.

Similarly, the actor can be expressed by

$$u_{s,k+1}(t_k) = \omega_a^T \phi_a(t_k) + \epsilon_a(t_k), \quad (18)$$

where ω_a represent the ideal coefficients, ϕ is the hidden layer neurons. ϵ_a is the approximation error. After inserting the approximation function to the (16), the following equation is acquired:

$$\begin{aligned} \epsilon_B(t_k) = \omega_c^T[\phi_c(t_{k+1}) - \phi_c(t_k)] &+ \int_{t_k}^{t_{k+1}} [x(\tau)^T Wx(\tau) + u_{s,k}(\tau)^T Ru_{s,k}(\tau) \\ &+ 2\omega_a^T \phi_a(\tau)R(u_s(\tau) - u_{s,k}(\tau))]\,\mathrm{d}\tau, \end{aligned} \quad (19)$$

where the ϵ_B is the approximation error and it is:

$$\epsilon_B = -\epsilon_c(t_{k+1}) + \epsilon_c(t_k) - \int_{t_k}^{t_{k+1}} 2\epsilon_a^T(\tau)R(u_s(\tau) - u_{s,k}(\tau))\,\mathrm{d}\tau. \quad (20)$$

The approximate critic and actor are noted with:

$$\begin{aligned} \hat{Q}_k(t_k) &= \hat{\omega}_c \phi_c(t_k) \\ \hat{u}_{s,k+1}(t_k) &= \hat{\omega}_a \phi_a(t_k), \end{aligned} \quad (21)$$

where define a residual error of for the approximation items:

$$\begin{aligned} e_1(t_k) &= \hat{\omega}_c^T[\phi_c(t_{k+1}) - \phi_c(t_k)] \\ &+ \int_{t_k}^{t_{k+1}} (x^T(\tau)Wx(\tau) + u_{s,i}(\tau)^T Ru_{s,i}(\tau))\,\mathrm{d}\tau \\ &+ \mathbf{v}(\hat{\omega}_a)^T \int_{t_k}^{t_{k+1}} 2R(u_s(\tau) - u_{s,k}(\tau)) \otimes \phi_a(\tau)\,\mathrm{d}\tau, \end{aligned} \quad (22)$$

where $\mathbf{v}$ is a operator transform the ω_a to a vector through stacking the columns one by one. The Kronecker product '⊗'

is used to translate u_s and ϕ_a to a vector which has same length with $\mathbf{v}(\hat{\omega}_a)$. Then define

$$\begin{aligned}\sigma(t_k) &= \phi_c(t_{k+1}) - \phi_c(t_k)\\ \eta(t_k) &= \int_{t_k}^{t_{k+1}} 2R(u_s(\tau) - u_{s,k}(\tau)) \otimes \phi_a(\tau)\,\mathrm{d}\tau\\ p(t_k) &= \int_{t_k}^{t_{k+1}} (x^T(\tau)Wx(\tau) + u_{s,k}^T(\tau)Ru_{s,k}(\tau))\,\mathrm{d}\tau \end{aligned} \tag{23}$$

and define the follow notes:

$$\begin{aligned}\hat{Z} &= [\hat{\omega}_c, \mathbf{v}(\hat{\omega}_a)]^T\\ \rho(t_k) &= [\sigma(t_k), \eta(t_k)]^T\end{aligned} \tag{24}$$

then the equation (22) could be note with:

$$e_1(t_k) = \hat{Z}^T\rho(t_k) + p(t_k) \tag{25}$$

Because of a experience replay is utilized to increase the train speed and diversity of samples. The samples used to train the weight are randomly selected from the experience. The states and residual errors of past data could be note as:

$$\begin{aligned}e_1(t_i) &= \hat{Z}^T\rho(t_i) + p(t_i)\\ i &= 1, 2, 3, \ldots, n_b,\end{aligned} \tag{26}$$

where n_b is the size of sample size. $[t_1, t_2, \ldots, t_{n_b}]$ is a random sequence whose element between t_0 and t_k.

Then the weigh update rule is designed:

$$\hat{Z}_{k+1} = \hat{Z}_k - \frac{\alpha}{n_b}\sum_{i=1}^{n_b}\frac{\rho(i)}{h^2(i)}[\hat{Z}_k^T\rho(i) + p(i)], \tag{27}$$

where $h = (\rho^T\rho + 1)^{\frac{1}{2}}$ is the normalization. and α is the learning rate.

Define the error between ideal weights and estimation weights as $\tilde{Z}_k = Z - \hat{Z}_k$. Then according to (25) and (19), the follow equation could be obtained:

$$\begin{aligned}e_1(i) &= (Z^T - \tilde{Z}_k^T)\rho(i) + p(i)\\ &= \epsilon_B - \tilde{Z}_k^T\rho(i)\end{aligned} \tag{28}$$

define the trend of the approximation error $S = \tilde{Z}_{k+1}^2 - \tilde{Z}_k^2$:

$$\begin{aligned}S &= (\hat{Z}_{k+1} - Z)^2 - (\hat{Z}_k - Z)^2\\
&= (\hat{Z}_k - \frac{\alpha}{n_b}\sum_{i=1}^{n_b}\frac{\rho(i)}{h^2(i)}[\hat{Z}_k^T\rho(i) + p(i)] - Z)^2 - (\hat{Z}_k - Z)^2\\
&\le \frac{\alpha^2}{n_b}\sum_{i=1}^{n_b}\frac{\rho^T(i)\rho(i)}{h^4(i)}[\hat{Z}_k^T\rho(i) + p(i)]^2\\
&\quad - 2(\hat{Z}_k^T - Z^T)\frac{\alpha}{n_b}\sum_{i=1}^{n_b}\frac{\rho(i)}{h^2(i)}[\hat{Z}_k^T\rho(i) + p(i)]\\
&\le \frac{\alpha^2}{n_b}\sum_{i=1}^{n_b}\frac{\rho^T(i)\rho(i)}{h^4(i)}e_1(i)^2 + 2\frac{\alpha}{n_b}\sum_{i=1}^{n_b}\tilde{Z}_k^T\frac{\rho(i)}{h^2(i)}\epsilon_B(i)\\
&\quad - 2\frac{\alpha}{n_b}\sum_{i=1}^{n_b}\frac{\tilde{Z}_k^T\rho(i)\rho^T(i)\tilde{Z}_k}{h^2(i)}\\
&\le \frac{\alpha^2}{n_b}\sum_{i=1}^{n_b}\frac{\rho^T(i)\rho(i)}{h^4(i)}(\epsilon_B(i) - \tilde{Z}_k^T\rho(i))^2\\
&\quad + 2\frac{\alpha}{n_b}\sum_{i=1}^{n_b}\tilde{Z}_k^T\frac{\rho(i)}{h^2(i)}\epsilon_B(i)\\
&\quad - 2\frac{\alpha}{n_b}\sum_{i=1}^{n_b}\frac{\tilde{Z}_k^T\rho(i)\rho^T(i)\tilde{Z}_k}{h^2(i)}\\
&\le \frac{\alpha^2}{n_b}\sum_{i=1}^{n_b}\frac{\rho^T(i)\rho(i)}{h^4(i)}\epsilon_B^2(i)\\
&\quad + \frac{\alpha^2}{n_b}\sum_{i=1}^{n_b}\frac{\rho^T(i)\rho(i)}{h^4(i)}\tilde{Z}_k^T\rho(i)\rho^T(i)\tilde{Z}_k\\
&\quad - 2\frac{\alpha^2}{n_b}\sum_{i=1}^{n_b}\epsilon_B(i)\frac{\rho^T(i)\rho(i)}{h^4(i)}\tilde{Z}_k^T\rho(i) + 2\frac{\alpha}{n_b}\sum_{i=1}^{n_b}\epsilon_B(i)\frac{\tilde{Z}_k^T\rho}{h^2(i)}\\
&\quad - 2\frac{\alpha}{n_b}\sum_{i=1}^{n_b}\frac{\tilde{Z}_k^T\rho(i)\rho^T(i)\tilde{Z}_k}{h^2(i)}\\
&\le \frac{\alpha^2}{n_b}\sum_{i=1}^{n_b}\frac{\rho^T(i)\rho(i)}{h^4(i)}\epsilon_B^2(i) + \frac{\alpha^2}{n_b}\sum_{i=1}^{n_b}\frac{\tilde{Z}_k^T\rho(i)\rho^T(i)\tilde{Z}_k}{h^2(i)}\\
&\quad + 2\frac{\alpha+\alpha^2}{n_b}\sum_{i=1}^{n_b}|\frac{\epsilon_B(i)\tilde{Z}_k^T\rho(i)}{h^2(i)}| - \frac{\alpha}{n_b}\sum_{i=1}^{n_b}\frac{\tilde{Z}_k^T\rho(i)\rho^T(i)\tilde{Z}_k}{h^2(i)}\\
&\quad - \frac{\alpha}{n_b}\sum_{i=1}^{n_b}\frac{\tilde{Z}_k^T\rho(i)\rho^T(i)\tilde{Z}_k}{h^2(i)}\\
&\quad - \frac{\alpha^2}{n_b}\sum_{i=1}^{n_b}\frac{\tilde{Z}_k^T\rho(i)\rho^T(i)\tilde{Z}_k}{h^2(i)} + \frac{\alpha^2}{n_b}\sum_{i=1}^{n_b}\frac{\tilde{Z}_k^T\rho(i)\rho^T(i)\tilde{Z}_k}{h^2(i)}\\
&\quad - \sum_{i=1}^{n_b}\frac{\alpha+\alpha^2}{n_b h^2(i)}\epsilon_B^2(i) + \sum_{i=1}^{n_b}\frac{\alpha+\alpha^2}{n_b h^2(i)}\epsilon_B^2(i)\\
&\le -\frac{\alpha+\alpha^2}{n_b}\sum_{i=1}^{n_b}\frac{(\tilde{Z}_k^T\rho(i) - \epsilon_B(i))^2}{n_b h^2(i)}\\
&\quad - \sum_{i=1}^{n_b}(\alpha - 2\alpha^2)\frac{\tilde{Z}_k^T\rho(t)\rho^T(t)\tilde{Z}_k}{n_b h^2(i)}\\
&\quad + \sum_{i=1}^{n_b}\frac{(\alpha+\alpha^2)\epsilon_B^2(i) + \alpha^2\epsilon_B^2(i)}{n_b h^2(i)}\\
&\le \frac{-(\alpha - 2\alpha^2)\lambda_{min}(H_\rho)\|\tilde{Z}\|^2 + (\alpha + 2\alpha^2)b_B}{n_b h^2(i)}\end{aligned} \tag{29}$$

where $\lambda_{min}(H_\rho)$ is the minimum eigenvalue of H_ρ and b_B is maximum of $\epsilon_B(i)$. The H_ρ is shown as follow:

$$H_\rho = \sum_{i=1}^{n_b}\rho(i)\rho^T(i) \tag{30}$$

So, the $\tilde{Z}_k$ will converge to the follow range with designed update rule and proper parameters:

$$\lim_{k\to\infty}\{\|\tilde{Z}_k\| \le \sqrt{\frac{(\alpha + 2\alpha^2)b_B}{(\alpha - 2\alpha^2)\lambda_{min}(H_\rho)}}\} \tag{31}$$

where $\alpha < 0.5$ is the learning rate. Meanwhile, the approximation error can be reduced by increasing the minimum eigenvalue of H_ρ. During learning process, the old data in the experience buffer will be replaced by new one when the $\lambda_{min}(H_\rho)$ increases, which make the performance better and better.

TABLE 1. Parameters of the quadrotor UAV.

Symbol	Description	Value	Units
m	Mass	1.5	kg
g	Acceleration of gravity	9.8	m/s^2
L	Radius of quadrotor	4.5	$10^{-1}m$
r	Radius of rotor	9.4	$10^{-2}m$
J_{xx}	Moment of quadrotor inertia	1.75	$10^{-2}N \cdot s^2 \cdot rad^{-1}$
J_{yy}	Moment of quadrotor inertia	1.75	$10^{-2}N \cdot s^2 \cdot rad^{-1}$
J_{zz}	Moment of quadrotor inertia	3.18	$10^{-2}N \cdot s^2 \cdot rad^{-1}$
J_R	Moment of one rotor	9.90	$10^{-5}N \cdot s^2 \cdot rad^{-1}$
C_T	Thrust coefficient	1.12	$10^{-5}N \cdot (rad/s)^{-2}$
C_M	Torque coefficient	1.47	$10^{-7}N \cdot m \cdot (rad/s)^{-2}$
C_R	Throttles coefficient	6.48	$10^2 rad/s$

IV. SIMULATION

A. SIMULATION SETUP

The quadrotor UAVs model used for validate proposed method is a '+' type one, whose physical parameters are listed in Table 1. The reference signals for the quadrotor UAV are position in XYZ and attitude in yaw. The control value is the percentage of the throttle for the motor. The simulation contain 'learning mode' and 'test mode': The 'learning mode' means the quadrotor UAV starts from a random state for better exploration and its target is stabilize the UAV to the balance point; the 'test mode' starts from a fixed state and its target may be stabilization or tracking. The critic network and the action network are updated in each step on both modes. The simulations are executed by episode. A episode means the quadrotor UAV starts from the initial state and runs max 10 seconds. Then the UAV will return to a initial state and start again. A episode will stop when the quadrotor UAV fails. The failure means that the state of quadrotor UAV goes out of the predetermined safe range. During simulation, the sample time is 0.01 second, so the max sample steps are 1000 steps. And The update method for the UAV dynamic is the classical Runge-Kutta. The training for controller weights was executed each step. In order to make zero as a reasonable balance point, the original point was set on the position where was one meter above the ground. The hardware of the simulation platform is a personal computer: CPU Intel i7-7700, Memory 16.0 GB, Display adapter AMD RadeonT 450 and SSD PM961. The software of the platform is Windows 10, Python 3.6 and tensorflow 1.5.

For the convenient to compare performance indexes between different methods, the score was defined as follow. The high score, the better performance:

$$S_k = \sum_{i=1}^{T_{max}} (c_s - \int_{t_i}^{t_{i+1}} r(x(\tau), u_s(\tau))\, d\tau) \tag{32}$$

where the S_k meant the score of the k^{th} episode and the c_s was a positive constant used to award the 'quadrotor survive'. The c_s caused that the longer quadrotor UAVs got higher score, which was reasonable because the stabilization was the first priority. The c_s was 1 in follow simulations. $r(x(i), u_s(i))$ was a positive reward function defined before. So, the max score was $c_s \cdot T_{max}$. The define of S_k was calculated with original rewards and did not depend on a specific controller, so it was fair to evaluate the performance of different methods.

Any admissible controller could be selected as the basic controller in the our proposed method. The main task of the basic controller is guaranteeing stable rather than hight performance. The eventual performance is decided by RL controller. During our simulation, a basic cascade PD method was selected. And its equation was shown as follow:

$$\begin{aligned} \theta_d &= -k_{p1}p_x - k_{d1}\dot{p}_x \\ \varphi_d &= -k_{p2}p_y - k_{d2}\dot{p}_y \\ \tau_1 &= -k_{p3}(\theta - \theta_d) - k_{d3}\dot{\theta} \\ \tau_2 &= -k_{p4}(\varphi - \varphi_d) - k_{d4}\dot{\varphi} \\ \tau_3 &= -k_{p5}p_z - k_{d5}\dot{p}_z \\ \tau_4 &= -k_{p6}\phi - k_{d6}\dot{\phi} \end{aligned} \tag{33}$$

where the $k_{p1}, \cdots, k_{p6}, k_{d1}, \cdots, k_{d6}$ were the control gains. The θ_d and φ_d were the intermediate reference signal for attitude cycle in the cascade PD controller. Then (2) was used to translate the thrust to the control value for the motor.

B. SIMULATION RESULT DURING TRAINING

The changes of scores and control performance during training are shown in this section, which presented that the control effect rise obviously with training. The initial weights of critic network and action network were random. In order to increase the explorations, the training processes were executed in 'learning mode' and a noise was added to the final control value. The variance of noise decreased over time. The algorithm update were executed all the steps and the performance will stabilize after some episodes. In order to verify the repeatability of the training, the taring process were executed 100 runs with different random initial weights and each run contained 300 episodes. The tendency of TD-error and scores was shown in Fig. 4, in which not only the average but also standard deviation of runs were given. From the result, we found that the score rise obviously and went stable after about 150 episodes; the TD-error decreased to a low value after 20 episodes. Meanwhile, the standard deviation went down with process of training, so the proposed would acquire similar final performance although it started from different random weights. The tendency of average and standard deviation shown that the proposed method enjoyed good repeatability.

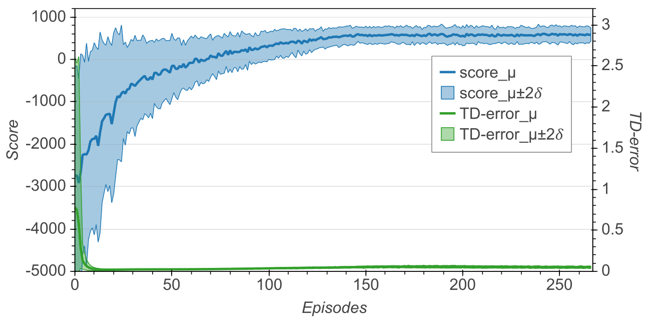


FIGURE 4. The tendency of score and TD-error during training. The bold solid lines represent the average value and the shadow areas are their 95 percent confidence interval (2 standard deviations).

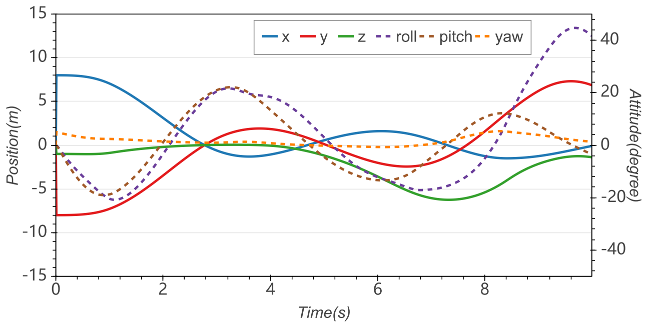


FIGURE 5. The control performance from a fixed state using proposed method with initial weights.

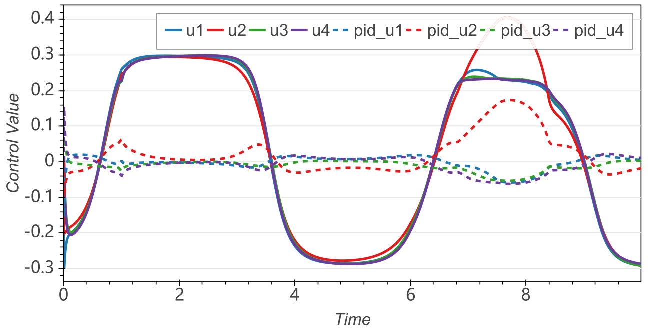


FIGURE 6. The changing of control value during position control experiments using proposed method with initial weights.

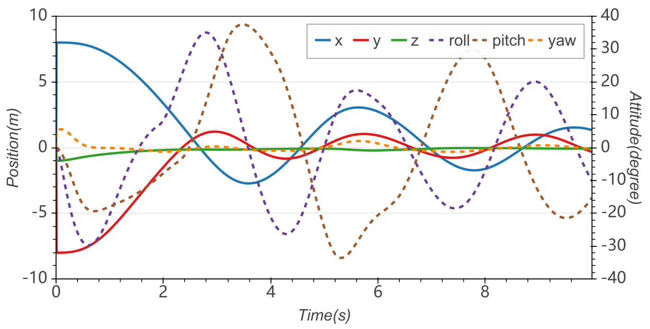


FIGURE 7. The control performance from a fixed state using proposed method with survived weights.

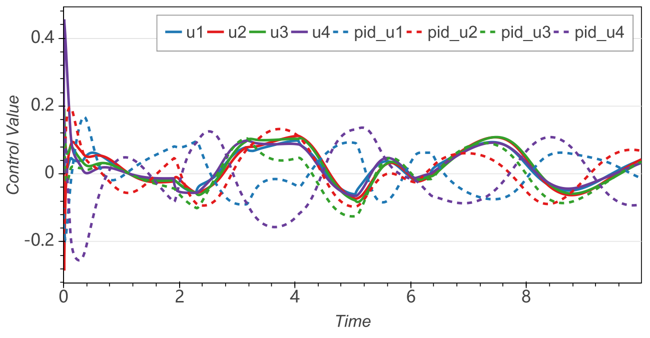


FIGURE 8. The changing of control value during position control experiments using proposed method with survived weights.

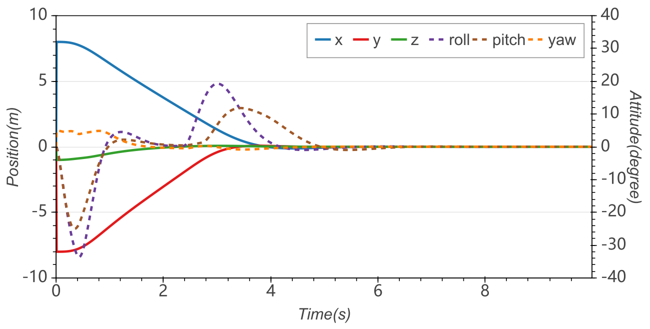


FIGURE 9. The control performance from a fixed state using proposed method with high performance weights.

In order to find the detailed changing of control performance during training, we saved the initial weights, survived weights (the quadrotor UAV did not fail before max steps) and high-performance weight (after 150 episodes). The detailed control results were shown in Fig. 5 to Fig. 10. With each group weights, two figures were provided: the one included position and attitude information, the other was about the control value. Compared between Fig. 5, Fig. 7 and Fig. 9, it could be found that the convergence rate increased and the overshoot decreased obviously. And the attitude control were more responsive, which would benefit of the position control. From Fig. 6, Fig. 8 and Fig. 10, it was obvious that the saturated output time decrease over training which was good to actuators. So the dynamic performance of quadrotor raised obviously with controller training. As the position moving of quadrotor UAVs depends on its attitude, the attitude values converge after the position is stable. Meanwhile, the position moving can be implement with any yaw when roll and pitch are regulated well. As a result, the yaw converges to 0 after 2.1s, but the roll and pitch are still changing until 5s in Fig. 9.

C. COMPARISON EXPERIMENTS WITH WIND DISTURBANCE

In this section, the effect of disturbance rejection of the proposed method was demonstrated with wind disturbance. At first, the quadrotor UAV was set on random point. The task was regulating it to the balance point and then hovering there.

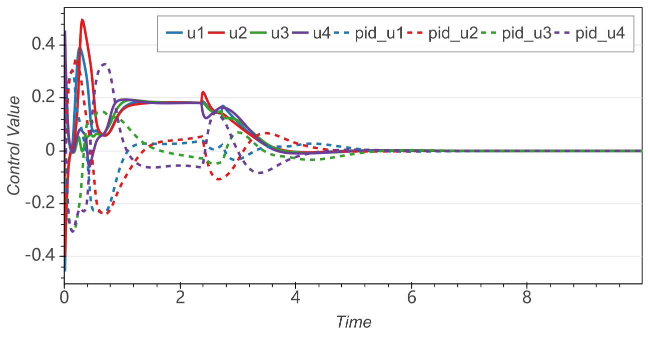


FIGURE 10. **The changing of control value during position control experiments using proposed method with high performance weights.**

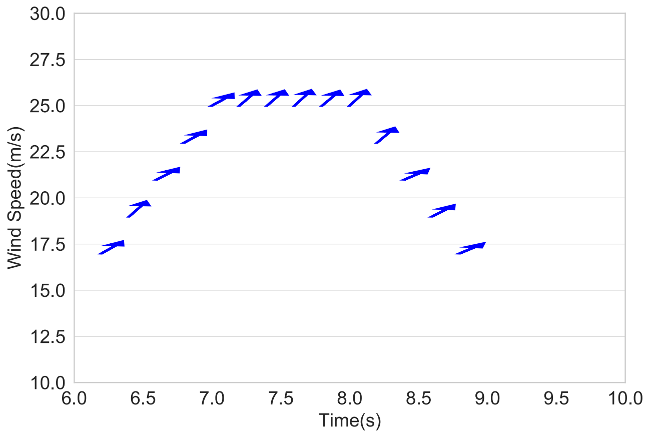


FIGURE 11. **The direct wind disturbance is changing over time. The starter of arrow is wind speed; the angle between arrow and vertical line arrow is direction of wind.**

The wind disturbance was added on the time tick on 6th second and lasted 4 seconds. The wind disturbance was assumed as follow: wind speed changes from 15 m/s to 25 m/s; wind direction varies from 40° to 60°. The wind disturbance is shown in Fig. 11. For the sake of comparison, a same mission was executed on a dynamic surface control method [10].

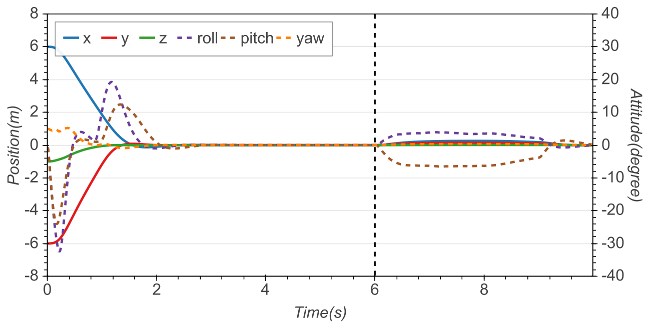


FIGURE 12. **The control performance using proposed method with wind disturbance. The black dash line showed the start of wind. The attitude changes for rejecting disturbances.**

The experiments were shown on the Fig. 12 and Fig. 13. From the result we found that, quipped with proposed method, the attitude had a swifter response when the wind

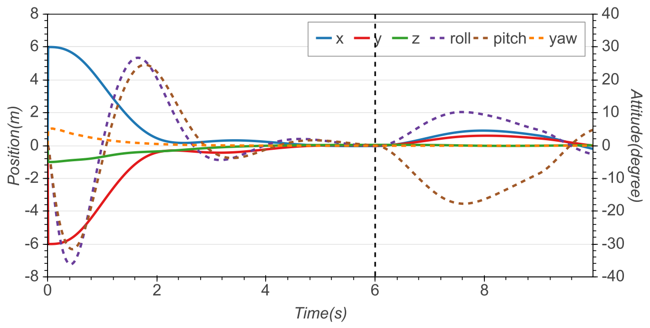


FIGURE 13. **The control performance using dynamic surface control method with wind disturbance.**

disturbance happened and enjoyed smaller range of drift. So our method can still hold high control accuracy under wind disturbance.

D. TRACKING EXPERIMENT

Although the presented method was not training with tracking cases, we found that the method could tracking a preset trajectory. Before applied on the tracking problem, it was necessary to adjust the input of action network. The errors between the reference signals r_x, r_y, r_z, r_ϕ in the trajectory and the related system states were used to replace the original controller input. The equation of new input was shown as follow:

$$x' = [x_1, x_2, x_3, x_4, x_5 - r_\phi, x_6, x_7 - r_x, x_8, x_9 - r_y, x_{10}, x_{11} - r_z, x_{12}] \quad (34)$$

where x' was the new input of system state for the action network and critic network.

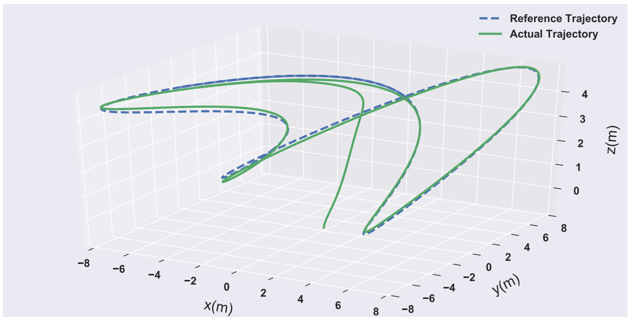


FIGURE 14. **A 3d trajectory tracking using proposed method.**

The reference trajectory was governed by $r_x = -8\sin 0.1t$, $r_y = 8\cos 0.3t$, $r_z = 2\sin(0.2t) + 3$ and $r_\phi = (\pi/20)\sin(0.2t)$ and the initial states were all zeros except $x_{11} = -1$. The simulation time was set as 70 seconds. The actual and reference trajectories in 3-D space were shown in Fig. 14, in which we found that the reinforcement based supplementary method could track the desired trajectory accurately with only the error information other than the intact trajectory dynamic. The detailed tracking performances

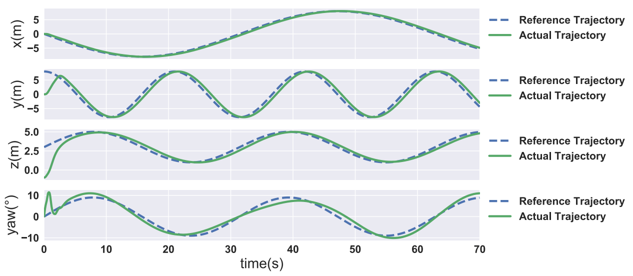


FIGURE 15. Detailed performance of trajectory tracking using proposed method.

in individual reference channels were shown in Fig. 15. From the figure, it shown that the quadrotor UAV reached the reference signal during 2 seconds and then tracked the trajectory with high accuracy.

V. CONCLUSION

In this proposed, we designed a RI based supplementary method for quadrotor to improve the performance of its basic controller. The basic controller can guarantee the stability during training and the RI controller decides the eventual performance. The actor-critic structure is employed including several RL techniques: including Q-learning, temporal difference (TD) and experience replay. The Q-learning is action-depended and off-policy, which provides smaller variance for actor training; The TD can update the algorithm each step, so it is possible to run our strategy on-line; And experience replay significantly increase the training rate. Although the method is developed for the control problem for quadrotor UAVs, we find it work well for the tracking problem during simulations.

REFERENCES

[1] P. Foehn, D. Falanga, N. Kuppuswamy, R. Tedrake, and D. Scaramuzza, "Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload," in *Proc. Robot., Sci. Syst.*, Jul. 2017, pp. 1–10.

[2] M. Hayajneh, M. Melega, and L. Marconi, "Design of autonomous smartphone based quadrotor and implementation of navigation and guidance systems," *Mechatronics*, vol. 49, pp. 119–133, Feb. 2018.

[3] B. Tian, Y. Ma, and Q. Zong, "A continuous finite-time output feedback control scheme and its application in quadrotor UAVs," *IEEE Access*, vol. 6, pp. 19807–19813, 2018.

[4] D. Shi, Z. Wu, and W. Chou, "Generalized extended state observer based high precision attitude control of quadrotor vehicles subject to wind disturbance," *IEEE Access*, vol. 6, pp. 32349–32359, 2018.

[5] A. L. Salih, M. Moghavvemi, H. A. F. Mohamed, and K. S. Gaeid, "Modelling and PID controller design for a quadrotor unmanned air vehicle," in *Proc. IEEE Int. Conf. Automat., Qual. Testing, Robot. (AQTR)*, vol. 1, May 2010, pp. 1–5.

[6] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep./Oct. 2004, pp. 2451–2456.

[7] X. Liang, Y. Fang, N. Sun, and H. Lin, "Nonlinear hierarchical control for unmanned quadrotor transportation systems," *IEEE Trans. Ind. Electron.*, vol. 65, no. 4, pp. 3395–3405, Apr. 2018.

[8] A. Das, F. Lewis, and K. Subbarao, "Backstepping approach for controlling a quadrotor using lagrange form dynamics," *J. Intell. Robotic Syst.*, vol. 56, nos. 1–2, pp. 127–151, 2009.

[9] X. Shao, J. Liu, H. Cao, C. Shen, and H. Wang, "Robust dynamic surface trajectory tracking control for a quadrotor UAV via extended state observer," *Int. J. Robust Nonlinear Control*, vol. 28, no. 7, pp. 2700–2719, 2018.

[10] J. Dou, X. Kong, X. Chen, and B. Wen, "Output feedback observer-based dynamic surface controller for quadrotor UAV using quaternion representation," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 231, no. 14, pp. 2537–2548, 2017.

[11] N. Fethalla, M. Saad, H. Michalska, and J. Ghommam, "Robust observer-based dynamic sliding mode controller for a quadrotor UAV," *IEEE Access*, vol. 6, pp. 45846–45859, 2018.

[12] H. Liu, J. Xi, and Y. Zhong, "Robust attitude stabilization for nonlinear quadrotor systems with uncertainties and delays," *IEEE Trans. Ind. Electron.*, vol. 64, no. 7, pp. 5585–5594, Jul. 2017.

[13] N. Wang, S.-F. Su, M. Han, and W.-H. Chen, "Backpropagating constraints-based trajectory tracking control of a quadrotor with constrained actuator dynamics and complex unknowns," *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published.

[14] T. X. Dinh and K. K. Ahn, "Adaptive tracking control of a quadrotor unmanned vehicle," *Int. J. Precis. Eng. Manuf.*, vol. 18, no. 2, pp. 163–173, 2017.

[15] G. Antonelli, E. Cataldi, F. Arrichiello, P. R. Giordano, S. Chiaverini, and A. Franchi, "Adaptive trajectory tracking for quadrotor MAVs in presence of parameter uncertainties and external disturbances," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 1, pp. 248–254, Jan. 2018.

[16] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, 3rd Quart., 2009.

[17] H. Zhang, H. Jiang, Y. Luo, and G. Xiao, "Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4091–4100, May 2017.

[18] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104, Feb. 2016.

[19] X. Yang, D. Liu, and Y. Huang, "Neural-network-based online optimal control for uncertain non-linear continuous-time systems with control constraints," *IET Control Theory Appl.*, vol. 7, no. 17, pp. 2037–2047, Nov. 2013.

[20] X. Yang, H. He, D. Liu, and Y. Zhu, "Adaptive dynamic programming for robust neural control of unknown continuous-time non-linear systems," *IET Control Theory Appl.*, vol. 11, no. 14, pp. 2307–2316, Sep. 2017.

[21] J. Yan, H. He, X. Zhong, and Y. Tang, "Q-learning-based vulnerability analysis of smart grid against sequential topology attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 200–210, Jan. 2017.

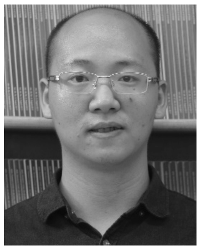
[22] X. Zhong, Z. Ni, and H. He, "Gr-GDHP: A new architecture for globalized dual heuristic dynamic programming," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3318–3330, Oct. 2017.

[23] H. He and X. Zhong, "Learning without external reward [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 48–54, Aug. 2018.

[24] C. Wu *et al.* (2018). "Variance reduction for policy gradient with action-dependent factorized baselines." [Online]. Available: https://arxiv.org/abs/1803.07246

[25] T. P. Lillicrap *et al.* (2015). "Continuous control with deep reinforcement learning." [Online]. Available: https://arxiv.org/abs/1509.02971

[26] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, "Adaptive control of quadrotor UAVs: A design trade study with flight evaluations," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1400–1406, Jul. 2013.



XIAOBO LIN received the B.S. degree in automation from the University of Science and Technology Beijing, China, in 2015, where he is currently pursuing the Ph.D. degree with the School of Automation and Electrical Engineering. His current research interests include nonlinear control for quadrotor UAVs, attitude estimation, and reinforcement learning.

YAO YU received the B.S. degree from Department of Control Science and Engineering, Huazhong University of Science and Technology, in 2004, and the M.S. and Ph.D. degrees from the Department of Automation, Tsinghua University, in 2010. She held a Postdoctoral position with Tsinghua University. She is currently an Associate Professor with the School of Automation and Electrical Engineering, University of Science and Technology Beijing. Her current research interests include nonlinear control, robust control, and time-delay systems.

CHANG-YIN SUN (M'17) received the B.S. degree from the College of Mathematics, Sichuan University, Chengdu, China, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 2001 and 2003, respectively, where he is currently a Professor with the School of Automation. His current research interests include intelligent control, flight control, pattern recognition, and optimal theory. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Neural Processing Letters*, and the IEEE/CAA JOURNAL OF AUTOMATICA SINICA.