Data-Driven Reinforcement Learning Control for Quadrotor Systems

Ngoc Trung Dang[1] and Phuong Nam Dao[2,*]

[1] Faculty of Electrical Engineering, Thainguyen University of Technology, Thai Nguyen, Vietnam
[2] School of Electrical and Electronic Engineering, Hanoi University of Science and University, Hanoi, Vietnam
Email: trungcsktd@tnut.edu.vn (N.T.D.); nam.daophuong@hust.edu.vn (P.N.D.)
*Corresponding author

Abstract—This paper aims to solve the tracking problem and optimality effectiveness of an Unmanned Aerial Vehicle (UAV) by model-free data Reinforcement Learning (RL) algorithms in both sub-systems of attitude and position. First, a cascade UAV model structure is given to establish the control system diagram with two corresponding attitude and position control loops. Second, based on the computation of the time derivative of the Bellman function by two different methods, the combination of the Bellman function and the optimal control is adopted to maintain the control signal as time converges to infinity with the addition of a discount factor. Third, according to off policy technique, the two proposed model-free RL algorithms are designed for attitude and position sub-systems in UAV control structure with a discount factor, respectively. In particular, the designed algorithms not only solve the trajectory tracking problem but also guarantee the optimality performance. Finally an illustrative system is used to verify the performance of the proposed model-free data RL algorithms in the UAV control system.

Keywords—data Reinforcement Learning (RL), Unmanned Aerial Vehicles (UAVs), quadrotor, Approximate/Adaptive Dynamic Programming (ADP), model-free based control

I. Introduction

The tracking control problem of Unmanned Aerial Vehicles (UAVs) has been extensively studied in many real-world applications, such as cargo transportation, precise agriculture, military, etc. [1-8]. In practice, the position and attitude dynamics tend to be uncertain and perturbed, which necessitates the design of nonlinear controllers with strong robustness against dynamic uncertainties and external disturbances [4]. Several classical nonlinear control methods studied for UAVs are nowadays available in the literature, such as adaptive dynamic Sliding Mode Control (SMC) [4], finite-time SMC [9], etc. However, it should be noted that the back-stepping technique laws have existed in almost all conventional nonlinear control designs [4, 9]. The majority of these methods is to handle the tracking performance of the entire UAV system while each subsystem guarantees stability. Therefore, the disadvantages of the aforementioned nonlinear control methods can be known not only in choosing the Lyapunov function candidate but also requiring complex computation. Moreover, the optimality problem has not been discussed in the unification with the tracking problem [4, 9].

Implementing the optimal control scheme for robotic systems, requires the approximate algorithms to solve Hamilton-Jacobi-Bellman (HJB) or Ricatti equations, which are difficult to directly solve by analytical method. The development of Reinforcement Learning Control (RLC) and Approximate/Adaptive Dynamic Programming (ADP) theories have significant implications for developing optimal control problems in robotic control systems [10, 11]. In particular, RLC is approached by Actor/Critic consideration [10, 11], wherein, through Neural Networks (NN) approximation and optimization solution, the learning algorithm was given to satisfy not only optimality effectiveness but also tracking problem. Nevertheless, the external disturbance and dynamic uncertainties were independently handled with the RL algorithm in [10, 11], which is considered the model-based method. Unlike the simultaneous learning in Actor/Critic in [10, 11], the work in [12-14] developed sequential training by Policy Iteration (PI) and Value Iteration (VI) for non-affine discrete-time systems and affine continuous-time systems, respectively. The convergence to optimal control and Bellman function is studied by considering the decreasing sequence to be limited. The data-driven PI in [14] was implemented with three successive phases for seeking Lipschitz admissible control and the optimal control policy. Moreover, the relation between PI and VI approaches was clarified in [12], the extension of output PI/VI ADP strategy was given with state reconstruction in [15-18]. On the other hand, data-driven learning algorithm was utilized to seek the output ADP for perturbed linear systems with exosystem based on the sampling data, which was collected from two virtual systems [15]. It is different from the off-policy technique considering the computation under the control signal being the control policy [10-15], the off-policy method was developed for model-free Hinfinity with the addition of an observer [16]. An approach to handling model-free

Manuscript received January 24, 2024; revised March 11, 2024; accepted March 22, 2024; published September 6, 2024.

requirements was introduced by using data-driven Actor/Critic Q learning for continuous linear systems [19]. On the other hand, to maintain the control signal as time comes to infinity, a model-free Q-learning algorithm was applied for linear discrete-time systems with a discount factor in the cost function [20]. Furthermore, the constrained optimal problem was addressed by integrating the Barrier function into the inverse optimal control problem [21].

Recently, particularly relevant for this paper, a data RL control strategy for a UAV was proposed in [1] to achieve the unification of the optimality problem and trajectory tracking control. An appealing feature of data RL control is that the proposed control is developed for complete dynamic uncertainties [1]. Moreover, the fault tolerant control problem was improved in the data RL algorithm [2]. However, the data RL technique was only developed for an attitude sub-system of UAV, and the cost function was mentioned but without a discount factor [1, 2, 6]. Additionally, a model-free RL control was also investigated for only an attitude sub-system of UAV to address the actuator saturation by developing an additional saturation function and saturated sliding surface [3]. On the other hand, there have been RL strategies for multiple UAVs, such as for a position-sub system [5], and an attitude sub-system [8]. Generally, the disadvantages of the recent references [2, 3, 10, 11] are to acquire the model parameters and to set the cost function being infinity, which means that the optimal control consideration is not satisfied. Thus, it is necessary to consider the addition of a discount factor in the cost function to develop the optimal control for UAV systems (see Table I).

TABLE I. THE PHYSICAL MEANINGS OF NOTATIONS

Notation	Meaning
$\mathcal{F}^i$	The inertial frame
$\mathcal{F}^b$	The Body-fixed frame
$\dot{X}$	a vector X expressed in $\mathcal{F}^j$, where $j = i, b$
e_n^i	a vector $n \times 1$ with the i^{th} element equal to 1 and 0 at the others
$0_{m,n}$	a $m \times n$ matrix with all 0 elements
R_b^i	The rotation matrix denotes a rotation from $\mathcal{F}^b$ to $\mathcal{F}^i$
c_*, s_*	cos (*), sin (*)
$\omega_1, \omega_2, \omega_3, \omega_4$	The velocities of the four rotors in Fig. 1
l	Distance between center of mass and rotor (Fig. 1)
m	Mass of the quadrotor (Fig. 1)

The rest of the paper is structured as follows. In Section II, we briefly introduce the position and attitude models of UAVs and present a control design objective. In Section III, we further present the RLC for nonlinear systems with a discount factor, and two model-free RL algorithms are developed for attitude and position-sub systems. In Section IV, the proposed algorithms are illustrated with a UAV control system to demonstrate their performance. This article concludes with a summary of the findings in Section V.

II. PRELIMINARIES AND PROBLEM STATEMENTS

Fig. 1 depicts the kinematics and dynamics of a quadrotor with two coordinate frames to be considered as a Body-fixed frame $\mathcal{F}^b$ and a NED inertial frame $\mathcal{F}^i$.

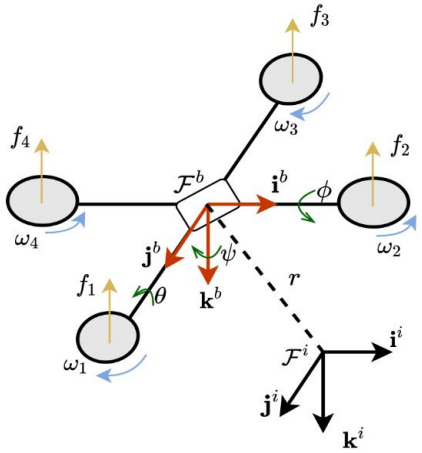

Fig. 1. Quadrotor model in NED coordinate

According to Ref. [1], the positions vector of the quadrotor and the orientation angles are defined under the NED inertial frame, the Euler angles Roll-Pitch-Yaw, respectively. It should be noted that the Euler angles are bounded such that $-\pi/2 < \phi < \pi/2$, $-\pi/2 < \theta < \pi/2$ and $-\pi < \psi < \pi$. On the other hand, the Euler angles vector is utilized to represent the following rotation matrix:

$$R = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \quad (1)$$

In Fig. 1, the angular velocities vector of the quadrotor concerning coordinate $\mathcal{F}^i$ is given by $\bar{\omega} = (p, q, r)^T$, which is expressed about Euler angles rate $\dot{\Theta} = (\dot{\phi}, \dot{\theta}, \dot{\psi})^T$ as follows:

$$\bar{\omega} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \dot{\Theta} \quad (2)$$

According to [1], the dynamics model of a quadrotor can be depicted as:

$$\begin{aligned} m\ddot{r} &= mgR e_3^3 - f R e_3^3, \\ J\dot{\bar{\omega}} &= -\bar{\omega} \times (J\bar{\omega}) + \tau. \end{aligned} \quad (3)$$

where $f \in \mathbb{R}$ indicates the total of lift forces in the frame $\mathcal{F}^i$ by $f = f_1 + f_2 + f_3 + f_4$, and τ is the torque in a frame $\mathcal{F}^b$ (see Fig. 1). Their relationship is described in Eq. (4):

$$\begin{bmatrix} f \\ T_\phi \\ T_\theta \\ T_\psi \end{bmatrix} = \begin{bmatrix} k_f & k_f & k_f & k_f \\ 0 & lk_z & 0 & -lk_z \\ -lk_z & 0 & lk_z & 0 \\ k_z & -k_z & k_z & -k_z \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} k_f & k_f & k_f & k_f \\ 0 & lk_z & 0 & -lk_z \\ -lk_z & 0 & lk_z & 0 \\ k_z & -k_z & k_z & -k_z \end{bmatrix} \begin{bmatrix} u_f \\ u_x \\ u_y \\ u_z \end{bmatrix} \quad (4)$$

with l to be the distance between the center of mass and the rotor and k_f, k_z, k_z are constant dynamics coefficients of force and torque, respectively. Similarly, we define and $u_x = \omega_2^2 - \omega_4^2$, $u_y = \omega_1^2 - \omega_3^2$, $u_z = -\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2$. Consequently, from Eqs. (2) and (3), the dynamics equations can be written as:

$$m\ddot{r} = fRe_3 - mge_3 \quad (5)$$

$$J\dot{\omega} = -\omega \times J\omega + \tau \quad (6)$$

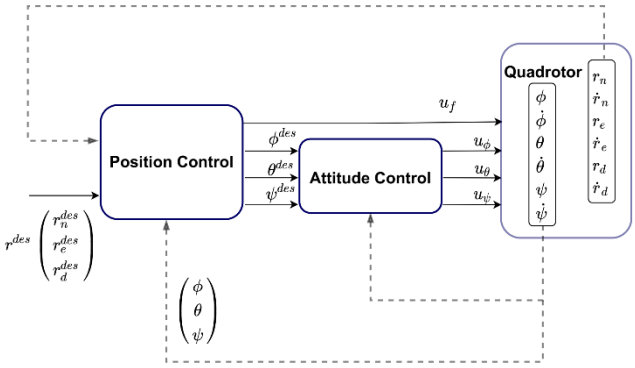

Fig. 2. Quadrotor control schematic

The objective is to seek an optimal controller for perturbed quadrotors to satisfy not only the tracking problem with a sophisticated trajectory but also minimize the discount factor-based cost function Eq. (8). Moreover, it should be noted that in the proposed control system (Fig. 2), the optimal control design approach is implemented for both subsystems in Eqs. (5) and (6) by observing system states with no prior knowledge of the object.

Remark 1. It is different from the classical trajectory tracking control objective to be solved by nonlinear control using Lyapunov stability theory in [9], the control objective in this article requires not only the trajectory tracking effectiveness but also the minimization of the given cost function. Moreover, the optimal control problem is necessary to satisfy both subsystems (Fig. 2), which is unified with trajectory tracking requirements.

III. RL Control Designs

In this section, a novel control structure illustrated in Fig. 2 is proposed with two RL-based control loops to be Position Controller and Attitude Controller after separating the Quadrotor into two subsystems as mentioned in Eqs. (5) and (6). The Position Controller is considered as an outer-loop controller to obtain the desired trajectory and its outputs are the desired Euler angles $(\phi^{des}, \theta^{des}, \psi^{des})^T$. The Attitude Controller has a duty to track the desired Euler angles. The two RL-based controllers are investigated under the influence of exponential function and data collection to satisfy the complete uncertainty.

A. RL Control Design with Discount Factor

The RL technique is investigated for the following affine system:

$$\frac{d}{dt}x(t) = F(x) + G(x)u(t). \quad (7)$$

The optimal control signal is designed to guarantee the minimization of the following infinite horizon cost function with a positive discount factor $\alpha > 0$.

$$V(x(t), u(t)) = \int_t^\infty e^{-\alpha(s-t)} U(x(s), u(s))\, ds, \quad (8)$$

where $U(x(s), u(s)) \triangleq x^T(s)Qx(s) + u^T(s)Ru(s)$, Q and $R \in \mathbb{R}^{n \times n}$ are two positive-definite symmetric constant matrices. It can be seen that the addition of discount factor in cost function Eq. (8) leads to the existence of input signals vector as time converges to infinity.

Hence, the admissible control policy is not necessary to discuss as previous references [11]. The constraint set of control signals is only necessary to satisfy the finite cost function Eq. (8). Thanks to the autonomous property of the affine model Eq. (7), the Bellman function obtained from Dynamic programming is the following static function:

$$V^*(x(t)) = \min_{u(x(t)) \in U} V(x(t), u(x)) \quad (9)$$

The modified Hamiltonian function associated with the discount factor is established by considering two different computation methods of the time derivative of the Bellman function $V^*(x(t))$. First, it is computed directly as:

$$\frac{d}{dt}V^*(x(t)) = \frac{\partial V^*}{\partial x}\frac{dx}{dt} = \frac{\partial V^*}{\partial x}F(x) + G(x)u^* \quad (10)$$

where u^* is the optimal control signal.

The second method to compute the time derivative of the Bellman function $V^*(x(t))$ is developed in the view of the dynamic programming principle with the following static Bellman function Eq. (9):

$$\begin{aligned} V^*(x(t)) &= \int_t^{t'} e^{-\alpha(s-t)} U(x(s), u^*(s))\, ds \\ &\quad + \int_{t'}^{\infty} e^{-\alpha(s-t')} U(x(s), u^*(s))\, ds \\ &= \int_t^{t'} e^{-\alpha(s-t)} U(x(s), u^*(s))\, ds + e^{-\alpha(t'-t)} V^*(x(t')). \end{aligned} \quad (11)$$

Therefore, Eq. (11) implies that:

$$\frac{V^*(t) - V^*(t-\Delta t)}{\Delta t} \approx \frac{1}{\Delta t}\int_{t-\Delta t}^{t} e^{-\alpha(s-t)} U(s, u^*(s))\, ds - \frac{e^{-\alpha \Delta t}-1}{\Delta t} V^*(t-\Delta t). \quad (12)$$

Observing Eqs. (12) and (10) as it implies that the static Bellman function $V^*(t)$ can be computed from the optimal control signal $u^*(t)$ after solving the following partial derivative equation:

$$U(t, u^*(t)) - \alpha V^*(t) + \frac{\partial V^*}{\partial t} F(t) + G(t) u^*(t) = 0. \quad (13)$$

To obtain the optimal control signal $u^*(t)$ from the Bellman function $V^*(t)$, based on the Dynamic Programming principle, it yields the following optimization problem:

$$V^*(t) = \min_{u(t) \in U} \left\{ \int_{t}^{t+\Delta t} U(t, u(t))\, ds + e^{-\alpha \Delta t} V^*(t+\Delta t) \right\} \quad (14)$$

As (14) implies the modified optimization problem is:

$$\min_{u(t) \in U} \left[U(t, u(t)) - \alpha V^*(t) + \frac{\partial V^*}{\partial t} F(t) + G(t) u(t) \right] = 0. \quad (15)$$

Denoting the modified Hamiltonian function to be associated with a discount factor $\alpha \geq 0$ as

$$H(t, u(t), \partial V / \partial t, V) = t^T Q t + u(t)^T R u(t) - \alpha V(t) + \partial V^T(t) / \partial t \left[F(t) + G(t) u(t) \right] \quad (16)$$

where it implies that the optimal control is then achieved from Eq. (15) as

$$u^*(t) = \arg\min_{u \in U(\cdot)} H(t, u(t), V^*(t)) = -\frac{1}{2} R^{-1} G^T(t) \partial V^*(t) / \partial t \quad (17)$$

Moreover, substituting the optimal control $u^*(t)$ Eq. (17) into Eq. (15) obtains the Partial Derivative Equation (PDE) as

$$H^*(t, u^*, \partial V^*/\partial t) = t^T Q t + \frac{1}{4} \partial V^{*T}(t) / \partial t\, G(t) R^{-1} G^T(t) \partial V^*(t) / \partial t - \alpha V^*(t) + \partial V^{*T}(t) / \partial t\, F(t) = 0. \quad (18)$$

However, it is impossible to analytically solve the PDE Eq. (18) to find the Bellman function from the optimal control signal $u^*(t)$. Hence, the data-driven algorithm is mobilized to seek the optimal control signal in Sections B and C.

Remark 2. The discount factor is added to the cost function Eq. (8) for keeping the control signals as the tracking problem is satisfied. However, it is necessary to address the impact of the term in Eq. (18), which establishes some modifications in actor/critic-based RL control design in Section III. B and C.

B. Data-Driven PI Position Controller

In this section, after achieving the separation of the quadrotor model, the control design of each sub-system is developed by the data-driven RL technique as follows. The first step is to rewrite the position sub-system described in Eq. (5):

$$\ddot{r} = -m^{-1} k_f u_f R e_3 + g e_3 \quad (19)$$

$$= -m^{-1} k_f u_f \quad (20)$$

where the term can be known as an offset element and eliminated by determining a bias $u_{f_0} = m k_f^{-1} g e_3$ to lift the quadrotor off the ground. To develop the RL technique as described in Section A, we employ the states vector $x_r = (r_n, \dot{r}_n, r_e, \dot{r}_e, r_d, \dot{r}_d)^T \in \mathbb{R}^6$. Thus, the position sub-system Eq. (19) can be transformed as:

$$\dot{x}_r = A_r x_r + B_r u_r \quad (21)$$

where $A_r = \mathrm{diag}(a_r, a_r, a_r) \in \mathbb{R}^{6\times 6}$, $a_r = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ and $B_r = m^{-1} k_f (e_2, e_4, e_6)$. Assume that, the desired trajectory is bounded and is the Lipschitz function. On the other hand, defining the reference $x_r^{des} = (r_n^{des}, \dot{r}_n^{des}, r_e^{des}, \dot{r}_e^{des}, r_d^{des}, \dot{r}_d^{des}) \in \mathbb{R}^6$ and $\dot{x}_r^{des}(t)$ can be completely represented as $\dot{x}_r^{des}(t) = A_{rd} x_r^{des}(t)$. Define $e_r = x_r - x_r^{des}$, Eq. (21) can be written as:

$$\dot{X}_r = \begin{bmatrix} \dot{e}_r \\ \dot{x}_r^{des} \end{bmatrix} = \begin{bmatrix} A_r & A_r - A_{rd} \\ 0_{6,6} & A_{rd} \end{bmatrix} \begin{bmatrix} e_r \\ x_r^{des} \end{bmatrix} + \begin{bmatrix} B_r \\ 0_{6,3} \end{bmatrix} u_r \quad (22)$$

The tracking performance function is chosen as:

$$V_r(X_r(t)) = \int_t^{\infty} e^{-\alpha(\tau - t)} \left[X_r^T(\tau) Q_r X_r(\tau) + u_r^T(\tau) R_r u_r(\tau) \right] d\tau \quad (23)$$

where $Q_r = \begin{bmatrix} Q_{er} & 0_{6,6} \\ 0_{6,6} & 0_{6,6} \end{bmatrix}$ with and to be both positive symmetric definite matrices. Note that, the term is added to penalize both tracking error cost and input control energy cost. It is easy to determine that although the desired trajectory is bounded, the control input does not converge to as comes to infinity. According to the work in

Section III. A, the Position Controller can be deployed as described in the following Algorithm 1:

Algorithm 1 (Fig. 3): Data-driven PI Position Control

Step 1 (Initialization): Starting the stabilizing control policy and the disturbance term to satisfy the PE condition. Implementing the data collection and establishing the threshold ϵ_p

Step 2 (Policy Evaluation): For each control signal $u_p^i(X_p)$, solve simultaneously the $V_p^{i+1}(X_p)$ and $u_p^{i+1}(X_p)$ by the following equation:

$$\begin{aligned} & V_p^{i+1}(X_p(t+\delta t)) - V_p^{i+1}(X_p(t)) \\ & = \int_t^{t+\delta t} X_p(\tau)^T Q_p X_p(\tau)\, d\tau + \int_t^{t+\delta t} u_p(X_p(\tau))^T R_p u_p(X_p(\tau))\, d\tau \\ & = \int_t^{t+\delta t} \nabla V_p^{i+1}(X_p(\tau))^T \dot{X}_p(\tau)\, d\tau + 2\int_t^{t+\delta t} u_p^{i+1}(X_p(\tau))^T R_p u_p(X_p(\tau))\, d\tau \\ & \quad - 2\int_t^{t+\delta t} u_p^{i+1}(X_p(\tau))^T R_p u_p^0(\tau)\, d\tau \end{aligned} \tag{24}$$

Step 3: (Policy Improvement): Update the control policy $u_p^i(X_p) \leftarrow u_p^{i+1}(X_p), i \leftarrow (i+1)$ and come back to Step 2 until $\| u_p^{i+1} - u_p^i \| \le \epsilon_p$

After obtaining the control signal in the control structure (Fig.1), the reference of the attitude control scheme can be obtained (u_{rn}, u_{re}, u_{rd}). According to Ref. [1], the desired yaw angle ψ^{des} can be chosen as zero and ϕ^{des}, θ^{des} can be easily solved as follows:

$$\begin{aligned} u_{rp} &= \sqrt{u_{rn}^2 + u_{re}^2 + (u_{rd} - u_{r_0})^2}, \quad \psi^{des} = 0, \\ \phi^{des} &= \arcsin\left(\frac{u_{rn}\sin(\psi^{des}) - u_{re}\cos(\psi^{des})}{u_{rp}}\right), \\ \theta^{des} &= \arctan\left(\frac{u_{rn}\cos(\psi^{des}) + u_{re}\sin(\psi^{des})}{u_{rd} - u_{r_0}}\right) \end{aligned} \tag{25}$$

C. Data-driven PI Attitude Controller

After the desired attitudes are obtained from Eq. (25), the objective of the attitude controller in Fig. 1 is to design the input signals for satisfying the optimal control problem. The model in Eq. (6) can be written as:

$$\dot{X} = J^{-1}T - J^{-1}C(\cdot,\cdot) \tag{26}$$

Based on the states vector $X = [\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T$, the attitude control scheme (Fig. 1) is similar to the Position Controller in subsection A. According to the model in Eq. (6), the attitude model can be rewritten as:

$$\dot{X}_{d\times 1} = \begin{bmatrix} 0_{6\times 6} & F_{6\times 6} \\ 0_{6\times 6} & 0_{6\times 6} \end{bmatrix} X_{d\times 1} + \begin{bmatrix} 0_{6\times 6} \\ G_{6\times 6} \end{bmatrix} u_{6\times 1} \tag{27}$$

It implies that the attitude control design can be developed in the following algorithm:

Algorithm 2 (Fig. 3): Data-driven PI Position Control

Step 1 (Initialization): Choosing the stabilizing control policy and the disturbance term $u_e(t)$, the threshold to satisfy the PE condition, and collecting the data.

Step 2 (Policy Evaluation): For each control signal $u^i(X)$, solve simultaneously the $V^{i+1}(X)$ and $u^{i+1}(X)$ by the following equation:

$$\begin{aligned} & V^{i+1}(X(t+\delta t)) - V^{i+1}(X(t)) \\ & = \int_t^{t+\delta t} X(\tau)^T Q X(\tau) + u^i(X(\tau))^T R\, u(X(\tau))\, d\tau \\ & = \int_t^{t+\delta t} \nabla V^{i+1}(X(\tau))^T \dot{X}(\tau)\, d\tau + 2\int_t^{t+\delta t} u^{i+1}(X(\tau))^T R\, u(X(\tau))\, d\tau \\ & \quad - 2\int_t^{t+\delta t} u^{i+1}(X(\tau))^T R\, u^0(\tau) + u_e\, d\tau \end{aligned} \tag{28}$$

Step 3 (Policy Improvement): Update the control policy $u^i(X) \leftarrow u^{i+1}(X), i \leftarrow (i+1)$ and come back to Step 2 until $\| u_p^{i+1} - u_p^i \| \le \epsilon_p$.

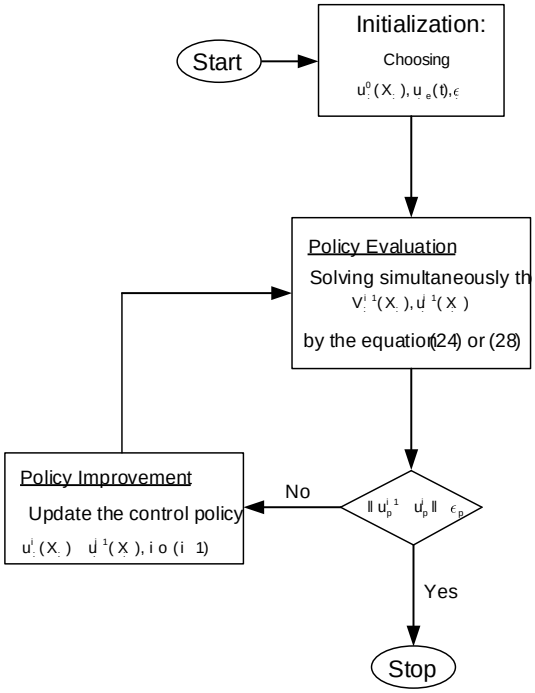

Fig. 3. The flowchart of Algorithm 1 or 2.

On the other hand, it is worth emphasizing that the training time is considered from the initial time to the time of obtaining the optimal controllers by RL Algorithms 1 and 2. Furthermore, due to the dynamic uncertainties in models in Eq. (5) and (6), the proposed RL Algorithms 1 and 2 are developed by data collection of input and state signals in the practical system, which are employed to solve the Eqs (24) (28). However, it is worth emphasizing that the existence of root for Eqs (24) (28), (Algorithm 1), (Algorithm 2) requires the Persistence of Excitation (PE) condition as shown in [10, 11].

IV. Simulation Results

Consider a perturbed quadrotor with the parameters as follows: $m = 2.0(kg), k_w = 1(Ns^2)$, and input disturbances in force are $0.1\sin Gt$ (Nm).

The desired trajectory is a spiral trajectory: $r^{des}(t) = [2\sin(at), 2\cos(at), 0.8t]^T$ where $a = 0.5$.

Initially, at the stage of collecting data, two PD (Proportion-Derivative) controllers are implemented for both position and attitude. These non-optimal controllers are tuned manually to keep the quadrotors stable with the position and attitude errors illustrated in Figs. 4 and 5. Additionally, to guarantee the PE conditions, the noises with and $u_{4e} = \sum_{m=1}^{500} 0.002\sin(w_m t)$ (w_m is a frequency chosen in range randomly), are added to the position and attitude controllers respectively. The next stage is to apply the two algorithms in the previous sections after obtaining data. The algorithms' parameters are chosen as:

$$Q_e = 100I_6, R_p = I_3, Q_{4e} = 100I_6,$$
$$R_4 = I_3, \gamma = 0.01, T_{step} = 0.01.$$

The activation functions of the critic and actor neural networks are second-order polynomials and first-order polynomials respectively. It can be seen that the convergence of the weights in Algorithms 1 and 2 is shown in Fig. 4. Moreover, it leads to the tracking problem is satisfied as shown in Fig. 5.

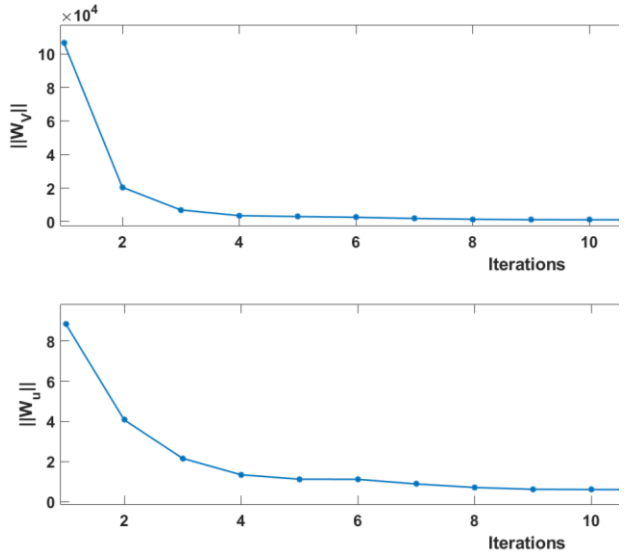

Fig. 4. The weight convergence of the learning stage

On the other hand, Fig. 6 shows the comparisons between the evolutions of tracking error depending on the discount factor. It can be seen that if the discount factor increases then the convergence speed of tracking error increases. However, it follows that the fluctuation will increase as the discount factor increases (Fig. 6). Additionally, unlike the classical controllers like Fuzzy and PID only considering the position control problem, the proposed Algorithms 1 and 2 develop not only the trajectory tracking control performance but also model-free RL strategy in the presence of dynamic uncertainties. Moreover, it can be seen that the advantage of these proposed methods is the extension of adding the discount factor and two RL algorithms, which have not been considered in the recent references [7, 10, 11].

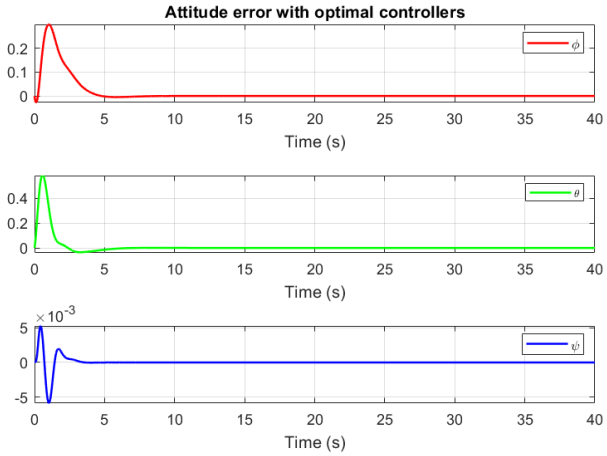

Fig. 5. The tracking of orientation angles

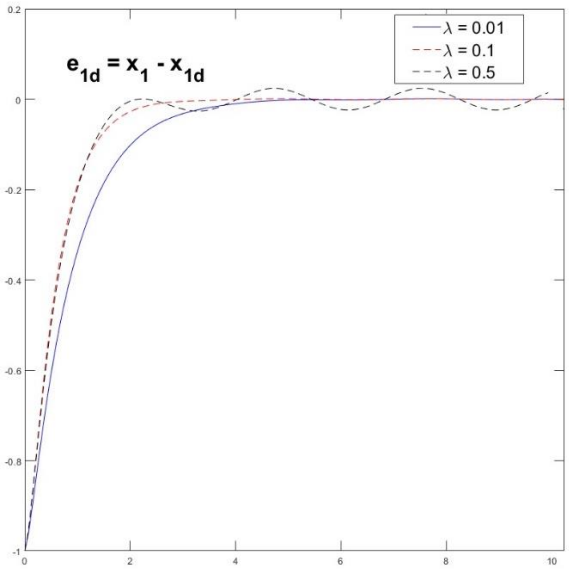

Fig. 6. The tracking error depends on the discount factor

V. Conclusion

This paper has designed the model-free data RL strategies for both attitude and position systems in cascade UAV control structure to achieve the unification of trajectory tracking problem and optimality purpose. The main idea is to establish the Off-Policy RL algorithm with a discount factor to satisfy the existence of control signal as time comes to infinity and obtain the model-free consideration without UAV model knowledge. Moreover, data collection and computation techniques are considered to achieve simultaneously the optimal value function and optimal control policy. Finally, an illustrative system is employed to validate the effectiveness of the proposed

model-free data RL algorithms in the UAV control system. In future work, we will consider extending the proposed strategies to the formation control problem of multiple UAVs.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Ngoc Trung Dang wrote the paper and conducted the simulation and Phuong Nam Dao wrote the draft paper and reviewed the paper. All authors had approved the final version.

FUNDING

This research was supported by the Research Foundation funded by the Thai Nguyen University of Technology.

REFERENCES

[1] W. Zhao et al., "Data-driven optimal formation control for quadrotor team with unknown dynamics," IEEE Transactions on Cybernetics, vol. 52, no. 8, pp. 7889–7898, August 2022.
[2] W. Zhao, H. Liu, and F. L. Lewis, "Data-driven fault-tolerant control for attitude synchronization of nonlinear quadrotors," IEEE Transactions on Automatic Control, vol. 66, no. 11, pp. 5584–5591, Nov 2021.
[3] O. Elhaki and K. Shojaei, "A novel model-free robust saturated reinforcement learning-based controller for quadrotors guaranteeing prescribed transient and steady-state performance," Aerospace Science and Technology, vol. 119, 107128, 2021.
[4] X. Wu, W. Zheng, X. Zhou, and S. Shao, "Adaptive dynamic surface and sliding mode tracking control for uncertain QUAV with time-varying load and appointed-time prescribed performance," Journal of the Franklin Institute, vol. 358, no. 8, pp. 4178–4208, 2021.
[5] L. Dou, S. Cai, X. Zhang, X. Su, and R. Zhang, "Event-triggered-based adaptive dynamic programming for distributed formation control of multi-UAV," Journal of the Franklin Institute, vol. 359, no. 8, pp. 3671–3691, 2022.
[6] J. C. B. Luo, Y. Luo, and K. Li, "Distributed UAV swarm formation and collision avoidance strategies over fixed and switching topologies," IEEE Transactions on Cybernetics, vol. 52, no. 10, pp. 10969–10979, August 2022.
[7] T. Xie, B. Xian, and X. Gu, "Fixed-time convergence attitude control for a tilt trimotor unmanned aerial vehicle based on reinforcement learning," ISA transactions, vol. 132, pp. 477–489, 2023.
[8] H. Du, W. Zhu, G. Wen, Z. Duan, and J. Lu, "Distributed formation control of multiple quadrotor aircraft based on nonsmooth consensus algorithms," IEEE Transactions on Cybernetics, vol. 49, no. 1, pp. 342–353, 2017.
[9] O. Mofid and S. Mobayen, "Adaptive sliding mode control for finite-time stability of quadrotor UAVs with parametric uncertainties," ISA Transactions, vol. 72, pp. 1–14, 2018.
[10] N. Khai, V. T. Dang, D. D. Pham, and P. N. Dao, "Formation control scheme with reinforcement learning strategy for a group of multiple surface vehicles," International Journal of Robust and Nonlinear Control, vol. 34, no. 3, pp. 2252–2279, 2024.
[11] V. T. Vu, T. L. Pham, and P. N. Dao, "Disturbance observer-based adaptive reinforcement learning for perturbed uncertain surface vessels," ISA transactions, vol. 130, pp. 277–292, 2022.
[12] B. Luo, Y. Yang, H. N. Wu, and T. Huang, "Balancing value iteration and policy iteration for discrete-time control," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 11, pp. 3948–3958, 2019.
[13] Q. Wei, D. Liu, Q. Lin, and R. Z. Song., "Discrete-time optimal control via local policy iteration adaptive dynamic programming," IEEE Transactions on Cybernetics, vol. 47, no. 10, pp. 3367–3379, 2017.
[14] C. Chen et al., "Adaptive optimal control of unknown nonlinear systems via homotopy-based policy iteration," IEEE Transactions on Automatic Control, pp. 1–8, 2023.
[15] K. Xie, Y. Xiao, and W. Y. Lan, "Optimal output regulation for unknown continuous-time linear systems by internal model and adaptive dynamic programming," Automatica, vol. 146, 110564, 2022.
[16] S. Li, P. Durdevic, and Z. Yang, "Model-free H∞ tracking control for de-oiling hydrocyclone systems via off-policy reinforcement learning," Automatica, vol. 133, 109862, 2021.
[17] F. W. Gao, Z. P. Jiang, and T. Liu, "Event-triggered adaptive optimal control with output feedback: An adaptive dynamic programming approach," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 11, pp. 5208–5221, 2021.
[18] F. Y. Zhao et al., "Adaptive optimal output regulation of linear discrete-time systems based on event-triggered output-feedback," Automatica, vol. 137, 110103, 2022.
[19] J. G. Zhao et al., "Reinforcement learning and optimal setpoint tracking control of linear systems with external disturbances," IEEE Transactions on Industrial Informatics, vol. 18, no. 11, pp. 7770–7779, 2022.
[20] P. Corrado and M. Sassano, "Q-learning for continuous-time linear systems: A data-driven implementation of the kleinman algorithm," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 52, no. 10, pp. 6487–6497, 2022.
[21] K. Lu et al., "Inverse optimal adaptive neural control for state constrained nonlinear systems," IEEE Transactions on Neural Networks and Learning Systems, pp. 1–12, Feb 2023.