Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Optimal Tracking Control Based on Integral Reinforcement Learning for An Underactuated Drone

Li, Shaobao; Durdevic, Petar; Yang, Zhenyu

Published in:
IFAC-PapersOnLine

DOI (link to publication from Publisher):
10.1016/j.ifacol.2019.08.048

Creative Commons License
CC BY-NC-ND 4.0

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Li, S., Durdevic, P., & Yang, Z. (2019). Optimal Tracking Control Based on Integral Reinforcement Learning for An Underactuated Drone. *IFAC-PapersOnLine*, *52*(8), 194-199. https://doi.org/10.1016/j.ifacol.2019.08.048

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 52-8 (2019) 55–60

Optimal Tracking Control Based on Integral Reinforcement Learning for An Underactuated Drone

Shaobao Li, Petar Durdevic, Zhenyu Yang

Department of Energy Technology, Aalborg University, Esbjerg Campus, Niels Bohus Vej 8, Esbjerg, 6700, Denmark (e-mail: shl@et.aau.dk;pdl@et.aau.dk;yang@et.aau.dk).

Abstract: A drone is desirable to perform various flying missions with different loads while always guaranteeing optimal flying performance. In this paper, an integral reinforcement learning algorithm is developed for a drone such that it can learn optimal control policy online. The drone is described by an underactuated nonlinear model and the inner-outer loop control strategy is applied for the navigation control. In the outer loop an optimal controller is designed to minimize a cost function with input saturation, and a policy iteration based integral reinforcement learning (IRL) algorithm is proposed. Critic-actor neural networks (NNs) are further applied for online implementation of the IRL algorithm. In the inner loop a quaternion based feedback attitude controller is designed to guarantee system stability. A simulation study is finally provided to demonstrate the effectiveness of the proposed IRL algorithm.

© 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Reinforcement learning, optimal control, neural network, inner-outer loop control.

1. INTRODUCTION

Drones have been widely employed in a number of applications such as surveillance, aerial inspection, goods delivery, and so on (Kaleem et al. (2018); Seo et al. (2018)). A good navigation control is necessary to guarantee flying performance of drones during various missions. Many feedback control approaches have been developed for navigation of drones with underactuated dynamics, e.g., (Sun and Cheng (2017); Viswanathan et al. (2018); Naldi et al. (2017)), just to name a few. However, the aforementioned control approaches are not designed for control performance optimization.

Recently, for the importance of energy efficiency in flying control of drones, optimal navigation control of drones has attracted much attention. Gandolfo et al. (2017) proposed a state feedback controller to minimize energy consumption during trajectory tracking. Although this paper suggests a way to lower energy consumption by well selecting a desired velocity, the proposed method is conservative and not optimal. Linear-Quadratic Regulation (LQR) is a mature technique for optimal control of drones. Alaimo et al. (2016) designed an LQR-PID controller for optimal tracking performance using the linearization model of a drone. Saha and Wadoo (2017) studied the control performance of LQG (Linear-Quadratic-Gaussian) based optimal control for a drone. However, LQR approach requires to linearize the nonlinear dynamics of drones and is not easy to handle input saturation directly. Hernandez et al. (2014) and Dentler et al. (2016) proposed a model predictive controller to achieve trajectory tracking of drones without overshoot, but they only considered drones modeled by linear dynamics without input saturation. More recently, as the fast development of machine learning techniques for optimal control, reinforcement learning as a typical machine learning has been appeared for drones to learn unknown environment and get better adaption, e.g., (Vankadari et al. (2018); Loquercio et al. (2018)). The salient feature of the studies in (Vankadari et al. (2018); Loquercio et al. (2018)) is that algorithm training can be model-free, but it is off-line and retraining is necessary once the dynamics of the drones have been changed.

In this paper, a policy iteration based integral reinforcement learning (IRL) algorithm is developed to optimize control performance of drones online. General underactuated nonlinear dynamics of drones and input saturation constraint are considered in this work. Critic-actor NNs are designed to implement the IRL algorithm online. The salient features of the proposed approach are: 1) a stabilizable controller is exerted on the drone to generate training data such that the learning process is on-line; and 2) an extra advantage is that the proposed approach can be extended to solve the optimal control problem of drones with time-varying mass because the optimal process is model-free. It is noted that the translational model of the drone is only used to determine the throttle control force.

The remainder of the paper is organized as follows: Section 2 describes the nonlinear system model of a drone. Section 3 presents the policy iteration based IRL algorithm. Section 4 provides a simulation example to demonstrate the effectiveness of the proposed algorithm and Section 5 concludes the paper.

2. SYSTEM DESCRIPTION

Through this paper, $\{\mathcal{I}\}$ and $\{\mathcal{B}\}$ are used to denote the inertial reference frame fixed to the earth surface and the reference coordinate frame attached to the drone's gravity

2405-8963 © 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.08.048

center, respectively. The drone is a Vertical Take-Off and Landing (VTOL) aerial vehicle and can be described by the following underactuated dynamic model (Naldi et al. (2017)):

$$\dot{p} = v \tag{1}$$

$$m\dot{v} = u_f \Omega e_3 - mge_3 \tag{2}$$

$$\dot{\Omega} = \Omega S(\omega) \tag{3}$$

$$J\dot{\omega} = S(J\omega)\omega + u_\tau \tag{4}$$

where $p = [x, y, z]^T \in \mathcal{R}^3$ and $v = [v_x, v_y, v_z]^T \in \mathcal{R}^3$ are the position and velocity of the drone in $\{\mathcal{I}\}$ frame, respectively, $\omega = [\omega_x, \omega_y, \omega_z]^T$ is the angular velocity of the drone in $\{\mathcal{B}\}$ frame, $m > 0 \in \mathcal{R}$ is the mass, $J = J^T > 0 \in \Re^{3\times3}$ is the inertia matrix, $g = 9.8N/Kg$ is the acceleration of gravity, $e_3 = [0, 0, 1]^T$, $u_f \in \mathcal{R}$ is the throttle control force, $u_\tau \in \mathcal{R}^3$ is the attitude control torque, $S(\omega) \in \mathcal{R}^{3\times3}$ is a skew symmetric matrix for a vector $\omega = [\omega_1, \omega_2, \omega_3]^T \in \mathcal{R}^3$:

$$S(\omega) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{5}$$

and Ω defined in equation (6) with ϕ, θ and ψ being the Euler angles is the rotation matrix from frame $\{\mathcal{B}\}$ to $\{\mathcal{I}\}$. It is noted that (6) follows the "$x-y-z$" convention.

Equations (1) and (2) describe the translational motion while equations (3) and (4) describe the attitude control of the drone. In Engineering, a unit quaternion is usually used to describe the attitude of drones, denoted as $q = [\eta, \epsilon^T]^T \in \mathcal{S}_3$, where η is a scalar, $\epsilon \in \mathcal{R}^3$ is a vector, and $\mathcal{S}_3 := \{x \in \mathcal{R}^4 : |x| = 1\}$ is an 3-dimensional unit sphere. Applying the Rodrigues formula (Shuster (1993)):

$$\Omega(q) = I_3 + 2\eta S(\epsilon) + 2S(\epsilon)^2, \tag{7}$$

equation (3) can be rewritten in the quaternion form as

$$\dot{q} = \frac{1}{2} q \otimes \nu(\omega) = \frac{1}{2} \begin{bmatrix} -\epsilon^T \\ \eta I_3 + S(\epsilon) \end{bmatrix} \omega, \tag{8}$$

where $\nu(\omega) = \begin{bmatrix} 0 & \omega^T \end{bmatrix}^T$ and $\otimes$ is an operator defined on two quaternions $q_i = \begin{bmatrix} \eta_i & \epsilon_i^T \end{bmatrix}^T$, $i = 1, 2$, as follows:

$$q_1 \otimes q_2 = \begin{bmatrix} \eta_1 & -\epsilon_1^T \\ \epsilon_1 & \eta_1 I_3 + S(\epsilon_1) \end{bmatrix} \begin{bmatrix} \eta_2 \\ \epsilon_2 \end{bmatrix}. \tag{9}$$

3. INTEGRAL REINFORCEMENT LEARNING BASED CONTROL DESIGN

An optimal control is always desirable for a drone to track a desired trajectory while minimizing power consumption. However, as the change of load, the traditional control laws (Naldi et al. (2017); Zhao et al. (2015)) with fixed control parameters cannot always guarantee optimal control performance. Towards this end, in this paper an optimal control law based on reinforcement learning is developed aiming at tuning the control parameters of the control law online to achieve optimal tracking performance whenever the change of load.

In this section, an inner-outer loop control strategy is applied to design the trajectory tracking controller of the drone, where the position control system (1)-(2) and the attitude control system (3)-(4) are the outer loop and inner loop, respectively. For the outer loop control, an integral reinforcement learning (IRL) algorithm will be presented such that the control force u_f can be tuned to be optimal automatically. Then, an attitude controller is designed for the inner loop to guarantee system stability.

3.1 Optimal Outer Loop Control

The objective of this paper is to find an optimal control policy u_f^* such that the drone can track a desired reference trajectory $p_d(t)$ while minimizing a given cost function. It is assumed that $\dot{p}_d$ and $\ddot{p}_d$ are known.

Let $\tilde{p} = p - p_d$ and $\tilde{v} = v - \dot{p}_d$. Now, we can formulate the trajectory tracking problem as follows:

$$\begin{cases} \dot{\tilde{p}} = \tilde{v} \\ m\dot{\tilde{v}} = u_f \Omega e_3 - mge_3 - m\ddot{p}_d \end{cases} \tag{10}$$

It is noted that system (10) does not satisfy the standard form of IRL as stated in (Lewis et al. (2012)). Towards this end, a virtual controller u_c in the form of

$$u_c = u_f \Omega e_3 - mge_3 - m\ddot{p}_d \tag{11}$$

is defined. It is noted that such design follows the *vectored-thrust control* paradigm (Hua et al. (2013)). The control force can be calculated by

$$u_f = \|mge_3 + m\ddot{p}_d + u_c\|, \tag{12}$$

where $\|\cdot\|$ denotes the Euclidean norm. Then, the desired Euler angles $\gamma_d \triangleq [\phi_d, \theta_d, \psi_d]^T$ for the inner loop can be determined by

$$\Omega_d e_3 = \frac{mge_3 + m\ddot{p}_d + u_c}{u_f}, \tag{13}$$

To avoid singularity of (13), it is assumed that $\|mge_3 + m\ddot{p}_d\| > \underline{m}$ and $\|u_c\| \le \bar{\kappa}$ with $\bar{\kappa} < \underline{m}$ such that

$$\|mge_3 + m\ddot{p}_d + u_c\| \ge \underline{m} - \bar{\kappa} > 0.$$

Therefore, it is naturally define a saturation constraint for the virtual controller u_c. Towards this end, we define a cost function as follows with respect to $\xi(t) = [\tilde{p}(t)^T, \tilde{v}(t)^T]^T$:

$$V(\xi(t)) = \int_t^\infty \left(\xi(\tau)^T Q \xi(\tau) + U(u_c(\tau))\right) d\tau, \tag{14}$$

where $Q \in \mathcal{R}^{6\times6} > 0$, and $U(u_c(t))$ is a positive-definite integrand function defined as

$$U(u_c) = 2\int_0^{u_c} (\kappa \tanh^{-1}(\chi/\kappa))^T R d\chi, \tag{15}$$

where κ is the saturation bound of u_c, i.e. $\|u_c\| \le \kappa$, and $R \in \mathcal{R}^{3\times3} > 0$ is assumed to be a diagonal matrix for simplicity of analysis. Then, the control problem is to find an optimal control u_c^* to minimize cost function (14).

Definition 1. (Admissible Controller). A control policy $u_c(\xi(t))$ is said to be admissible with respect to (14) if $u_c(\xi(t))$ is continuous, $u_c(0) = 0$, $u_c(\xi(t))$ stabilizes the error system (10), and $V(\xi(t))$ is finite.

$$\Omega = \begin{bmatrix} \cos(\theta)\cos(\psi) & -\cos(\theta)\sin(\psi) & \sin(\theta) \\ \cos(\phi)\sin(\psi) + \cos(\psi)\sin(\phi)\sin(\theta) & \cos(\phi)\cos(\psi) - \sin(\phi)\sin(\theta)\sin(\psi) & -\cos(\theta)\sin(\phi) \\ \sin(\phi)\sin(\psi) - \cos(\phi)\cos(\psi)\sin(\theta) & \cos(\psi)\sin(\phi) + \cos(\phi)\sin(\theta)\sin(\psi) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{6}$$

In the following, the design process of the reinforcement learning control will be described.

Differentiating $V(\xi(t))$ along the error system (10) gives the following Bellman equation:

$$H(\xi, u_c, \nabla V) := \xi^T Q\xi + U(u_c) + \nabla V^T (A\xi + Bu_c) = 0. \tag{16}$$

where $\nabla V = \partial V/\partial \xi$, and

$$A = \begin{bmatrix} \mathcal{O}_3 & I_3 \\ \mathcal{O}_3 & \mathcal{O}_3 \end{bmatrix}, B = m^{-1} \begin{bmatrix} \mathcal{O}_3 \\ I_3 \end{bmatrix}, \tag{17}$$

where $\mathcal{O}_3$ is an 3rd-order square zero matrix and I_3 is an 3rd-order identity matrix.

The optimal solution V^* of (16) satisfies the following Hamilton-Jacobi-Bellman (HJB) equation:

$$\min_{u_c} \left(\xi^T Q\xi + U(u_c) + \nabla V^T (A\xi + Bu_c)\right) = 0. \tag{18}$$

Differentiating the HJB equation with respect to u_c, one can obtain the optimal controller as follows:

$$u_c^* = -\kappa \tanh\left(\frac{1}{2\kappa} R^{-1} B^T \nabla V\right). \tag{19}$$

Substituting (19) back into (18) gives

$$\xi^T Q\xi + U(u_c^*) + \nabla V^T (A\xi + Bu_c^*) = 0, \tag{20}$$

which is an equation with respect to ∇V. If ∇V can be solved, the optimal controller u_c^* described by (19) can be determined. However, the HJB equation (20) is nonlinear and is difficult to obtain analytical solution. Thus we develop an integral reinforcement learning algorithm to solve the nonlinear HJB equation (20).

Remark 1. It is noted that the cost function V in (14) is actually a Lyapunov function for u_c. If u_c^* solves the HJB equation (20), we have $\dot{V} = -\xi^T Q\xi - U(u_c^*) \leq 0$, i.e., u_c^* stabilizes the error system (10).

3.2 Policy Iteration Based IRL Algorithm

In this subsection, a policy iteration based IRL algorithm is developed to solve the nonlinear HJB equation (20).

We define an *updating policy* u_c^j and rewrite the error system (10) as

$$\dot{\xi} = A\xi + Bu_c^j + B(u_c - u_c^j), \tag{21}$$

where u_c is an admissible control policy called as *behavior policy* which is continuously exerted on the drone to generate data and u_c^j learns from the data to update its parameters towards optimal control policy. The superscript j denotes the iteration number of the *updating policy*.

Keeping (16) in mind and differentiating the cost function V^{j+1} corresponding to $(j+1)$th iteration along system (21), one can obtain that

$$\begin{aligned} \dot{V}^{j+1} &= (\nabla V^{j+1})^T \left(A\xi + Bu_c^j + B(u_c - u_c^j)\right) \\ &= -\xi^T Q\xi - U(u_c^j) + (\nabla V^{j+1})^T B(u_c - u_c^j) \end{aligned} \tag{22}$$

Based on (22), we can give the following IRL algorithm for solving the nonlinear HJB equation (20).

It is noted that equation (23) is obtained by integrating (22) over the time interval $[t, t+T]$ with T being the reinforcement learning time interval. Then, we can obtain Theorem 1.

Algorithm 1: IRL Algorithm

1. Initialization: Fix a stabilizing control policy u_c and select any admissible control policy u_c^0. Do for $j = 0, 1, \cdots$ until convergence.

2. Policy evaluation step: Solve for $V^{j+1}(\xi(t))$ using

$$\begin{aligned} &V^{j+1}(\xi(t+T)) - V^{j+1}(\xi(t)) \\ &= \int_t^{t+T} \left(-\xi^T Q\xi - U(u_c^j) + (\nabla V^{j+1})^T B(u_c - u_c^j)\right) d\tau \end{aligned} \tag{23}$$

3. Policy is updated by

$$u_c^{j+1} = -\kappa \tanh\left(\frac{1}{2\kappa} R^{-1} B^T \nabla V^{j+1}\right). \tag{24}$$

Theorem 1. For any admissible control policy u_c^j, the updated policy u_c^{j+1} is still an admissible policy for system (10). Moreover, u_c^j converges to the optimal control policy u_c^* as $j \to \infty$.

Proof: Differentiating V^{j+1} along the system $\dot{\xi} = A\xi + Bu_c^{j+1}$ yields

$$\dot{V}^{j+1}(\xi, u_c^{j+1}) = (\nabla V^{j+1})^T \left(A\xi + Bu_c^{j+1}\right). \tag{25}$$

According to the HJB equation (20), one has

$$\nabla (V^{j+1})^T A\xi = -\xi^T Q\xi - U(u_c^j) - (V^{j+1})^T Bu_c^j. \tag{26}$$

The updating policy (24) implies

$$\nabla (V^{j+1})^T B = -2\kappa (\tanh^{-1}(u_c^{j+1}/\kappa))^T R. \tag{27}$$

Substituting (26) and (27) into (25) yields

$$\begin{aligned} \dot{V}^{j+1}(\xi, u_c^{j+1}) &= -\xi^T Q\xi - U(u_c^j) \\ &\quad + 2\kappa \left(\tanh^{-1}(u_c^{j+1}/\kappa)^T R(u_c^j - u_c^{j+1})\right). \end{aligned} \tag{28}$$

Applying mean value theorem to $U(u_c^j)$, there exists a $\bar{u}_c \in (0, u_c^j)$ such that $U(u_c^j) = 2\kappa \tanh^{-1}(\bar{u}_c/\kappa)^T R u_c^j$. Then, the following inequality always holds:

$$\begin{aligned} &2\kappa \left(\tanh^{-1}(u_c^{j+1}/\kappa)^T R(u_c^j - u_c^{j+1})\right) - U(u_c^j) \\ &= 2\kappa \left(\tanh^{-1}(u_c^{j+1}/\kappa)^T R(u_c^j - u_c^{j+1}) \right. \\ &\quad \left. - \tanh^{-1}(\bar{u}_c/\kappa)^T R u_c^j\right) \leq 0 \end{aligned} \tag{29}$$

due to the fact that $\tanh^{-1}(\cdot)$ is an monotone and odd function. Therefore, one has $\dot{V}^{j+1}(\xi, u_c^{j+1}) < 0$ for any $\xi \neq 0$. According to Definition 1, u_c^{j+1} is also an admissible control policy.

Next, we prove that u_c^j converges to the optimal control policy u_c^*, which minimizes the cost function. Consider the following equation along the trajectory $\dot{\xi} = A\xi + Bu_c^{j+1}$.

$$\begin{aligned} &V^{j+1}(\xi) - V^j(\xi) \\ &= -\int_0^\infty \left((\nabla V^{j+1} - \nabla V^j)^T (A\xi + Bu_c^{j+1})\right) d\tau \\ &= 2\kappa \int_0^\infty \left(\tanh^{-1}(u_c^{j+1}/\kappa)^T R(u_c^j - u_c^{j+1}) \right. \\ &\quad \left. + \int_{u_c^j}^{u_c^{j+1}} \left(\tanh^{-1}(\chi/\kappa)^T R d\chi\right)\right) d\tau \end{aligned} \tag{30}$$

Applying mean value theorem, there exists a $\hat{u}_c \in (u_c^j, u_c^{j+1})$ such that

$$\begin{aligned}&\tanh^{-1}(u_c^{j+1}/\kappa)^T R(u_c^j-u_c^{j+1})+\int_{u_c^j}^{u_c^{j+1}}\left(\tanh^{-1}(\chi/\kappa)^T R d\chi\right)\\&=\tanh^{-1}(u_c^{j+1}/\kappa)^T R(u_c^j-u_c^{j+1})\\&\quad-\tanh^{-1}(\hat{u}_c/\kappa)^T R(u_c^j-u_c^{j+1})\le 0\end{aligned}, \tag{31}$$

which implies that $V^{j+1}(\xi)\le V^j(\xi)$. It is noted that $V^{j+1}(\xi)=V^j(\xi)$ holds for $\xi\neq 0$ if and only if $u_c^j=u_c^{j+1}=u^*$. Therefore, u_c^j will converge to the optimal control policy u_c^*. The proof is thus completed. □

Remark 2. It is noted from (30) that $V^{j+1}(\xi)=V^j(\xi)$ holds if $\xi=0$, as a result, u_c^j will stop converging to the optimal control policy. Therefore, a probing noise should be added into the control loop to guarantee persistence of excitation. The persistent excitation (PE) condition can refer to Theorem 3 in (Modares and Lewis (2014)).

Remark 3. It is noted from Algorithm 1 that the actual controller is u_c during the reinforcement learning process and the updating policy u_c^j does not affect the control loop before it converges to the optimal control policy. Therefore, Algorithm 1 has the following two properties for system control: 1) even if the initially admissible control policy u_c^0 is bad for control performance, it will not exert any influence to the closed-loop control; 2) u_c is replaced by u_c^j only when u_c^j reaches optimal.

Remark 4. It is worth mentioning that the learning process could continue to be implemented to monitor if the obtained policy is optimal. When the obtained policy deviates from the learning results due to parameter changes, e.g., mass of the drone, a new optimal control policy can be obtained by implementing Algorithm 1 using the obtained control policy as the initial control policy.

3.3 Online Implementation of Algorithm 1

In this subsection, critic-actor NNs are developed to implement Algorithm 1 online. The cost function V^{j+1} is approximated by a critic NN as follows:

$$V^{j+1}(\xi)=(\hat{W}_1^{j+1})^T\varphi_1(\xi) \tag{32}$$

where $\hat{W}_1^{j+1}\in\mathcal{R}^l$ is the weight matrix and $\varphi_1(\xi)\in\mathcal{R}^l$ is the activation function.

Define $D^{j+1}=\frac{1}{2\kappa}R^{-1}B^T\nabla V^{j+1}$, which is approximated by an actor NN as follows:

$$D_i^{j+1}(\xi)=(\hat{W}_{2i}^{j+1})^T\varphi_{2i}(\xi), \tag{33}$$

where $i=1,2,3$ is the index of the elements of D^{j+1}, $\hat{W}_{2i}^{j+1}\in\mathcal{R}^l$ and $\varphi_{2i}(\xi)\in\mathcal{R}^l$.

Substituting (32) and (33) into (23), one has

$$\begin{aligned}&e^{j+1}\triangleq(\hat{W}_1^{j+1})^T(\varphi_1(\xi(t))-\varphi_1(\xi(t+T)))\\&+2\kappa r_i\sum_{i=1}^{3}\int_t^{t+T}\left((\hat{W}_{2i}^{j+1})^T\varphi_{2i}(\xi))^T R(u_{ci}-u_{ci}^j)\right)d\tau\\&-\int_t^{t+T}\left(\xi^TQ\xi+U(u_c^j)\right)d\tau\end{aligned}, \tag{34}$$

where r_i, $i=1,\cdots,3$, is the ith diagonal element of R, and u_{ci} and u_{ci}^j are the ith element of u_c and u_c^j, respectively. Denote

$$\hat{W}^{j+1}=\left[(\hat{W}_1^{j+1})^T,(\hat{W}_{21}^{j+1})^T,(\hat{W}_{22}^{j+1})^T,(\hat{W}_{23}^{j+1})^T\right]^T \tag{35}$$

$$\Phi(t)=\begin{bmatrix}\varphi_1(\xi(t))-\varphi_1(\xi(t+T))\\2\kappa r_1\int_t^{t+T}\varphi_{21}(\xi)(u_{c1}-u_{c1}^j)d\tau\\2\kappa r_2\int_t^{t+T}\varphi_{22}(\xi)(u_{c2}-u_{c2}^j)d\tau\\2\kappa r_3\int_t^{t+T}\varphi_{23}(\xi)(u_{c3}-u_{c3}^j)d\tau\end{bmatrix} \tag{36}$$

$$Y(t)=\int_t^{t+T}\left(\xi^TQ\xi+U(u_c^j)\right)d\tau \tag{37}$$

Then, (34) can be rewritten as follows

$$e^{j+1}(t)=(\hat{W}^{j+1})^T\Phi(t)-Y(t). \tag{38}$$

Based on gradient descent approach, the weight matrix $\hat{W}^{j+1}$ is updated by

$$\dot{\hat{W}}^{j+1}=-\Gamma\frac{\Phi(t)}{(1+\Phi^T(t)\Phi(t))^2}\left((\hat{W}^{j+1})^T\Phi(t)-Y(t)\right), \tag{39}$$

where $\Gamma>0$ is the learning rate matrix and $(1+\Phi^T(t)\Phi(t))^2$ is used for normalization. Therefore, in each iteration, the control policy in (24) is updated by

$$u_c^{j+1}=-\kappa\tanh((\hat{W}_2^{j+1})^T\varphi(\xi(t))), \tag{40}$$

where $\hat{W}_2^{j+1}=[(\hat{W}_{21}^{j+1})^T,(\hat{W}_{22}^{j+1})^T,(\hat{W}_{23}^{j+1})^T]^T$.

3.4 Inner Loop Control

For simplicity, the left side of (13) is described by a vector $[\nu_x,\nu_y,\nu_z]^T$. According to the definition of rotation matrix (6), the desired Euler angles can be determined by

$$\phi_d(t)=\arctan(\frac{-\nu_y}{\nu_z}),\ \theta_d(t)=\arcsin(\nu_x). \tag{41}$$

It is noted that $\Omega_d e_3$ has nothing to do with ψ_d, which implies ψ_d can be selected randomly.

Now, we have the desired Euler angles $\gamma_d=[\phi_d,\theta_d,\psi_d]^T$. We further need the higher-order information of the desired Euler angles, but they can hardly be obtained through differentiating (13) directly because of the unknown higher-order derivatives of u_c. Considering that γ_d is always bounded, we apply an high-gain observer to estimate the higher-order derivatives of γ_d as follows:

$$\begin{cases}\delta\dot{\pi}_1=\pi_2\\\delta\dot{\pi}_2=-\lambda_1\pi_2-\pi_1+\gamma_d\end{cases}. \tag{42}$$

where δ is any small positive constant, $\pi_{1,2}$ are the observer states, and λ_1 is a constant satisfying that the polynomial $s^2+\lambda_1 s+1$ is Hurwitz. According to Lemma 6.4 of (Ge et al. (2013)), one can directly obtain

$$\pi_1-\gamma_d=-\delta\varepsilon_1,\ \frac{\pi_2}{\delta}-\omega_d=-\delta\varepsilon_2, \tag{43}$$

where ω_d is the desired angular velocity, and ε_k, $k=1,2$, are the bounded estimation errors. Then, there exist constants t^* and b_k, only depending on ε_k, $k=1,2$, δ and λ_1 such that for all $t>t^*$,

$$\|\varepsilon_1\|\le b_1,\ \|\varepsilon_2\|\le b_2. \tag{44}$$

Therefore, we can define the estimation of the desired Euler angles and its derivatives as follows:

$$\hat{\gamma}_d=\pi_1,\ \hat{\omega}_d=\frac{\pi_2}{\delta},\ \dot{\hat{\omega}}_d=\frac{\dot{\pi}_2}{\delta}. \tag{45}$$

The estimated Euler angles $\hat{\gamma}_d$ can be described by a quaternion q_d. Then, we can define the following attitude errors:

$$\tilde{q} = q_d^{-1} \otimes q \tag{46}$$
$$\tilde{\omega} = \omega - \bar{\omega}_d, \tag{47}$$

where $\bar{\omega}_d = \Omega(\tilde{q})^T \hat{\omega}_d$. According to (8), the error attitude dynamics can be described by

$$\dot{\tilde{q}} = \frac{1}{2}\tilde{q} \otimes \begin{bmatrix} 0 \\ \tilde{\omega} \end{bmatrix}, \tag{48}$$
$$J\dot{\tilde{\omega}} = \Sigma(\tilde{\omega}, \bar{\omega}_d)\tilde{\omega} + S(J\bar{\omega}_d)\bar{\omega}_d - J\Omega(\tilde{q})^T\dot{\omega}_d + u_\tau. \tag{49}$$

where

$$\Sigma(\tilde{\omega}, \bar{\omega}_d) = S(J\tilde{\omega}) + S(J\bar{\omega}_d) - S(\bar{\omega}_d)J - JS(\bar{\omega}_d). \tag{50}$$

We refer the attitude controller design proposed in (Naldi et al. (2017)) and use the following attitude controller:

$$u_\tau = -k_p\tilde{\epsilon} - k_d\tilde{\omega} - S(J\bar{\omega}_d)\bar{\omega}_d - J\Omega(\tilde{q})^T\dot{\omega}_d. \tag{51}$$

4. SIMULATION STUDIES

In this section, a simulation example is provided to demonstrate the effectiveness of the proposed algorithm. The drone's mass is $m = 1.121$ kg and the inertial matrix is

$$J = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.0082 & 0 \\ 0 & 0 & 0.0148 \end{bmatrix}.$$

Positive definite matrices Q and R of the cost function are chosen to be identical matrices. Let $\kappa = 10$, $T = 0.1$s and the learning rate for $\hat{W}^{j+1}$ is $\Gamma = \text{diag}\{50I_9, 5I_{15}\}$. The optimal outer-loop control policy is automatically learned using Algorithm 1. The drone is driven to track an "8" shaped trajectory described by

$$p_d = \left[\cos(\frac{\pi}{60}t), \sin(\frac{\pi}{30}t), 0.5\right]^T. \tag{52}$$

To guarantee persistence of excitation, a small probing noise in the form of

$$n(t) \triangleq \begin{bmatrix} 0.15 \\ 0.15 \\ 0.075 \end{bmatrix}^T (\sin(8t)^2\cos(2t) + \sin(20t)^4\cos(7t)) \tag{53}$$

is added to the feedforward position and velocity signals, respectively.

The activation functions are defined as follows:

$$\varphi_1 = [\xi_1^2, \xi_2^2, \xi_3^2, \xi_4^2, \xi_5^2, \xi_6^2, \xi_1\xi_4, \xi_2\xi_5, \xi_3\xi_6]^T, \tag{54}$$
$$\varphi_{21} = [\xi_1, \xi_4, \xi_1^2, \xi_1\xi_4, \xi_4^2], \tag{55}$$
$$\varphi_{22} = [\xi_2, \xi_5, \xi_2^2, \xi_2\xi_5, \xi_5^2], \tag{56}$$
$$\varphi_{23} = [\xi_3, \xi_6, \xi_3^2, \xi_3\xi_6, \xi_6^2]. \tag{57}$$

For inner-loop control, choose $\delta = 20$, $\lambda_1 = 2$, $k_p = 4$, $k_d = 0.5$.

The simulation results are shown in Figs. 1-5. During the first 100s, a fixed control law $u_c = -\tilde{p} - \tilde{v}$ is exerted on the drone for data generation and probing noise (53) is added to guarantee persistence of excitation. From Figs. 1-2, one can observe that the drone tracks the reference trajectory with some small bounds which is induced by the added probing noise. From Fig. 3, one can observe that the weights of the critic-actor NNs converge to some

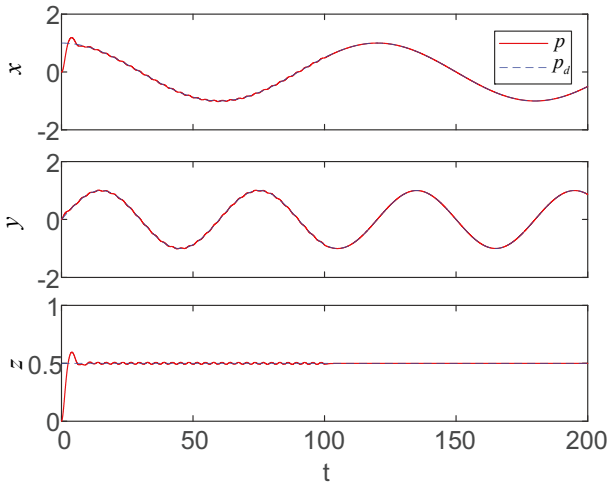

Fig. 1. The position trajectory of the drone.

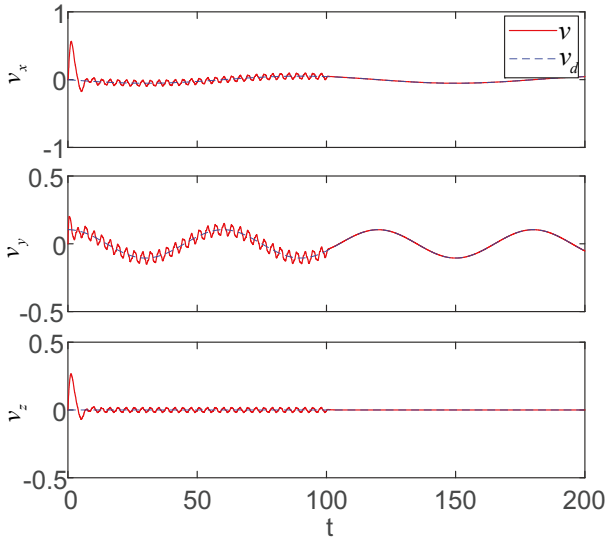

Fig. 2. The velocity of the drone.

constants at 100s, which means the learning process is almost completed. Then, at 100s the probing noises is turned off while the fixed control law u_c is replaced by the obtained control law u_c^j. One can further observe that the drone perfectly tracks the reference trajectory. Fig. 4 intuitively shows the trajectory tracking of the drone. Further to show the effectiveness of the proposed reinforcement learning based algorithm, we compare the control performance between the initial control law u_c, the updated control law u_c^j and the optimal control law without considering input saturation as shown in Fig. 5. It is shown that the updated control law significantly reduces the cost value comparing with the initial control law u_c and is very close to the cost value of the optimal control law. It is worth mentioning that the error between the updated control law and the optimal control law is induced by the approximation errors of the critic-actor NNs and the high-gain observer (42).

5. CONCLUSIONS

In this paper, the optimal control problem of underactuated drones with input saturation is studied. Inner-

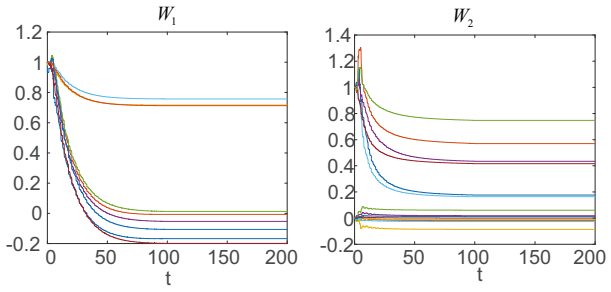

Fig. 3. The weights of the critic NN and actor NN.

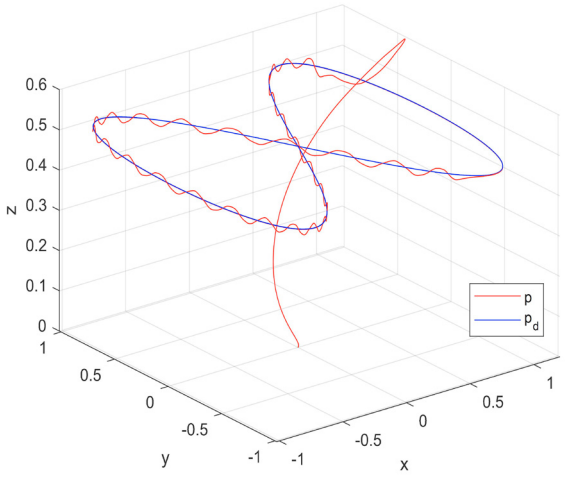

Fig. 4. The tracking trajectory of the drone.

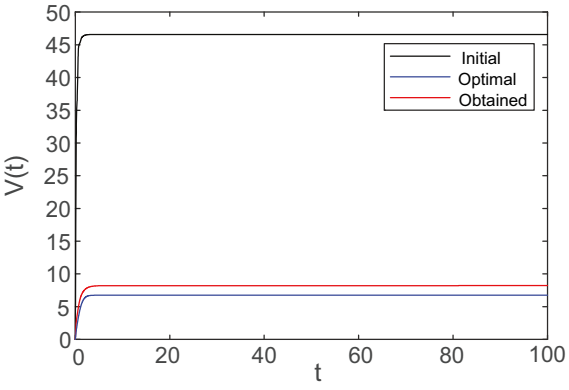

Fig. 5. Comparison of cost functions using initial, optimal and obtained control policies.

outer loop control strategy is applied to design navigation control algorithm, where in the outer loop an policy iteration based IRL algorithm is developed to tune the controller towards minimizing a given cost function and in the inner loop a quaternion based feedback control is adopted to design attitude controller. The salient feature of the proposed IRL algorithm is that the updating policy is independent of the closed-loop control. In the future, it would be very interesting to develop a value iteration based IRL algorithm, which is totally model-free and does not require admissible initial updating policy.

REFERENCES

Alaimo, A., Artale, V., Barbaraci, G., Milazzo, C., Orlando, C., and Ricciardello, A. (2016). LQR-PID control applied to hexacopter flight. *J Numer Anal, Ind Appl Math*, 9-10(3-4), 47–56.

Dentler, J., Kannan, S., Mendez, M.A.O., and Voos, H. (2016). A tracking error control approach for model predictive position control of a quadrotor with time varying reference. In *IEEE Intl Conf Robotics and Biomimetics*, 2051–2056.

Gandolfo, D.C., Salinas, L.R., Brandão, A., and Toibero, J.M. (2017). Stable path-following control for a quadrotor helicopter considering energy consumption. *IEEE Trans Contr Syst Tech*, 25(4), 1423–1430.

Ge, S.S., Hang, C.C., Lee, T.H., and Zhang, T. (2013). *Stable adaptive neural network control*, volume 13. Springer Science & Business Media.

Hernandez, A., Murcia, H., Copot, C., and De Keyser, R. (2014). Model predictive path-following control of an AR. drone quadrotor. In *Proc the XVI Latin American Contr Conf*, 14–17.

Hua, M.D., Hamel, T., Morin, P., and Samson, C. (2013). Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles. *IEEE Contr Syst*, 33(1), 61–75.

Kaleem, Z., Rehmani, M.H., Ahmed, E., Jamalipour, A., Rodrigues, J.J., Moustafa, H., and Guibene, W. (2018). Amateur drone surveillance: Applications, architectures, enabling technologies, and public safety issues: Part 2. *IEEE Commun Mag*, 56(4), 66–67.

Lewis, F.L., Vrabie, D., and Syrmos, V.L. (2012). *Optimal control*. John Wiley & Sons.

Loquercio, A., Maqueda, A.I., del Blanco, C.R., and Scaramuzza, D. (2018). Dronet: Learning to fly by driving. *IEEE Robot Auto Lett*, 3(2), 1088–1095.

Modares, H. and Lewis, F.L. (2014). Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning. *Automatica*, 50(7), 1780–1792.

Naldi, R., Furci, M., Sanfelice, R.G., and Marconi, L. (2017). Robust global trajectory tracking for underactuated VTOL aerial vehicles using inner-outer loop control paradigms. *IEEE Trans Auto Contr*, 62(1), 97–112.

Saha, S. and Wadoo, S. (2017). Linear optimal control of a parrot AR drone 2.0. In *IEEE MIT Undergraduate Res Techn Conf*, DOI: 10.1109/URTC.2017.8284178.

Seo, J., Duque, L., and Wacker, J. (2018). Drone-enabled bridge inspection methodology and application. *Auto in Construction*, 94, 112–126.

Shuster, M.D. (1993). A survey of attitude representations. *Navigation*, 8(9), 439–517.

Sun, J.H. and Cheng, L. (2017). Robust pid controller for AR drone. In *Proc Intl Conf Comput Sci Tech*, 1213–1221. World Scientific.

Vankadari, M.B., Das, K., Shinde, C., and Kumar, S. (2018). A reinforcement learning approach for autonomous control and landing of a quadrotor. In *Intl Conf Unmanned Aircraft Syst*, 676–683.

Viswanathan, S.P., Sanyal, A.K., and Samiei, E. (2018). Integrated guidance and feedback control of underactuated robotics system in SE(3). *J Intell Robot Syst*, 89(1-2), 251–263.

Zhao, B., Xian, B., Zhang, Y., and Zhang, X. (2015). Nonlinear robust adaptive tracking control of a quadrotor UAV via immersion and invariance methodology. *IEEE Trans Ind Elec*, 62(5), 2891–2902.