J Syst Sci Complex (2022) 35: 502–521

Trajectory Tracking Control for Under-Actuated Hovercraft Using Differential Flatness and Reinforcement Learning-Based Active Disturbance Rejection Control*

KONG Xiangyu · XIA Yuanqing · HU Rui · LIN Min · SUN Zhongqi · DAI Li

DOI: 10.1007/s11424-022-2037-0
Received: 14 January 2022

Abstract This paper proposes a scheme of trajectory tracking control for the hovercraft. Since the model of the hovercraft is under-actuated, nonlinear, and strongly coupled, it is a great challenge for the controller design. To solve this problem, the control scheme is divided into two parts. Firstly, we employ differential flatness method to find a set of flat outputs and consider part of the nonlinear terms as uncertainties. Consequently, we convert the under-actuated system into a full-actuated one. Secondly, a reinforcement learning-based active disturbance rejection controller (RL-ADRC) is designed. In this method, an extended state observer (ESO) is designed to estimate the uncertainties of the system, and an actorcritic-based reinforcement learning (RL) algorithm is used to approximate the optimal control strategy. Based on the output of the ESO, the RL-ADRC compensates for the total uncertainties in real-time, and simultaneously, generates the optimal control strategy by RL algorithm. Simulation results show that, compared with the traditional ADRC method, RL-ADRC does not need to manually tune the controller parameters, and the control strategy is more robust.

Keywords Active disturbance rejection control, differential flatness, reinforcement learning, trajectory tracking control, under-actuated system.

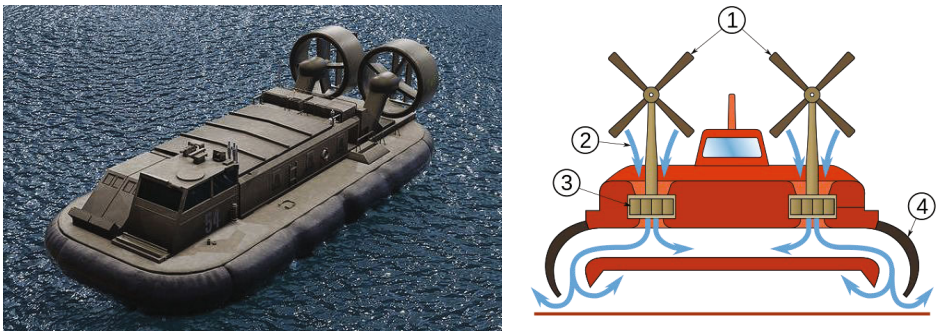
1 Introduction

A hovercraft as shown in Figure 1 (a), also known as an air-cushion vehicle, is an amphibious craft capable of traveling over land, water, mud, ice, and other surfaces. Due to its excellent amphibious performance, the hovercraft is widely used in military, rescue, shipping and other fields[1].

The air cushion, as shown in Figure 1 (b), is the key device to reduce the resistance of the hovercraft while sailing. The hovercraft uses blowers to produce a large volume of air below the air cushion, so that the air pressure below the hull is higher than the atmospheric pressure. The

KONG Xiangyu · XIA Yuanqing (Corresponding author) · HU Rui · LIN Min · SUN Zhongqi · DAI Li
School of Automation, Beijing Institute of Technology, Beijing 100190*, China.* Email: xia_yuanqing@bit.edu.cn.
*This paper was supported by the National Natural Science Foundation of China under Grant No. 61720106010.
◇ *This paper was recommended for publication by Guest Editor ZHANG Ying.*

pressure difference produces lift, as a result, the hovercraft can float above the running surface. Because of the particular working principle, different from traditional ships, hovercraft uses two high-power air propellers as power devices[1]. Therefore, the model of the hovercraft is under-actuated, nonlinear, and strongly coupled[2–6]. Moreover, the external disturbance caused by water waves and wind is also a challenge for hovercraft trajectory tracking control problem. Due to the complex structure and particular working principle of the hovercraft, manual control is extremely difficult and even impossible[7, 8]. In the last few years, various control methods have been proposed for hovercraft trajectory tracking control, such as coupled multiple sliding-mode control[2], Lyapunovs direct method[3, 4] and backstepping control method[5, 6]. Although these methods are designed to achieve the under-actuated hovercraft tracking tasks, they highly rely on an accurate hovercraft model which is difficult to be established and few of them considers modeling errors and external disturbances. In addition, the optimization of the control strategy is not considered in [2–6]. Therefore, for the hovercraft trajectory tracking control problem, under-actuated system control, disturbance rejection property and the optimization of control strategy are three challenges needed to be handled.



(a) 3D model of hovercraft (b) Schematic diagram of the air cushion

Figure 1 3D model of hovercraft and schematic diagram of the air cushion. (In subfigure (b), ①: propeller; ②: air; ③: blower; ④: soft apron; Figure source: https://en.wikipedia.org/wiki/Hovercraft)

In recent years, a mass of achievements were obtained in full-actuated system control field[9–14]. As a result, for under-actuated system control problem, some methods are proposed to transform an under-actuated system into a full-actuated one, such as differential flatness theory. Differential flatness is to find a set of flat outputs with the same dimension as the inputs, then perform dynamic feedback on the original system. Martin and Rouchon[15] demonstrate that any controllable under-actuated system with m inputs and $m+2$ states is flat and can be put into a multi-input chained form by dynamic feedback and coordinate change. This method has been successfully used to solve the tracking control problem for under-actuated systems such as quadrotor[16], aerial manipulator[17] and four-wheel steering vehicle[18].

To improve the disturbance rejection property of the control strategy, active disturbance rejection controller (ADRC) is proposed by Han in 1990s[19], because it does not rely on the accurate model and behaves excellent disturbance rejection property[20–23]. Coincidentally, based

on flat outputs, a flat under-actuated system can be put into a chained form which is the nominal model of ADRC. Therefore, ADRC combined with differential flatness theory is a classical method to solve the under-accurate system control problem. In [24], a method based on ADRC and differential flatness theory is proposed to deal with lateral path tracking control of under-actuated land vehicle. A differential flatness-based ADRC scheme for fractional-order systems is proposed in [25]. ADRC was originally given in a nonlinear gain structure to better accommodate the dynamic uncertainties and disturbances[19]. However, it occurs a technical bottleneck because of its complex analysis in the design of feedback control strategy. Therefore, in the case of reducing the optimization and adaptability, bandwidth-parameterization method is proposed for a practical convenience[26]. In [27], a linear ADRC and nonlinear ADRC switching control scheme is proposed and its stability is analyzed. A genetic algorithm based parameters tuning method for ADRC is proposed in [28]. But these methods cannot guarantee the optimality and adaptability of the control strategy.

To obtain an optimal control strategy, reinforcement learning (RL) has gained much attention from researchers. Reinforcement learning, inspired by human learning behavior, aims to learn an optimal policy to obtain a larger reward or smaller cost by interacting with the environment. Because of its good optimization performance, RL is widely used in optimal control[29–31]. However, for an environment with uncertainties and random disturbances, this method requires longer learning time and it is difficult to guarantee the optimality of the learned control strategy. Therefore, it is a natural idea to combine RL with ADRC, and it is also a complementary combination. In this integration, ADRC is used to estimate and reject disturbances to provide a new environment where RL can learn optimal feedback control policy in a relatively short period time. This is also the motivation of this paper.

Based on the above analysis, we propose a hovercraft trajectory tacking control scheme using differential flatness and reinforcement learning-based active disturbance rejection controller (RL-ADRC). The main contributions of this paper are summarized as follows: 1) We employ differential flatness method to find a set of flat outputs of the hovercraft model. Consequently, the under-actuated model is converted into a full-actuated one. 2) A reinforcement learning-based active disturbance rejection controller is designed. Compared with the traditional ADRC method, an optimal, robust and adaptive feedback control strategy is obtained.

The remainder of this paper is structured as follows. In Section 2, an under-actuated hovercraft model is established. In Section 3, flat outputs are found by differential flatness and the model is converted into a full-actuated one. In Section 4, we design the RL-ADRC scheme to solve the hovercraft trajectory tracking control problem. Simulation results are shown in Section 5. Finally, we conclude the paper in Section 6.

2 Under-Actuated Hovercraft Dynamics

With reference to several papers [32–34], in this paper, we establish the following hovercraft model.

$$\begin{cases} \tau = M\dot{\xi} + C(\xi)\xi + D\xi, \\ \dot{\eta} = J(\eta)\xi, \end{cases} \tag{1}$$

M, D, $C(\xi)$ and $J(\eta)$ are defined as following:

$$M = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_1 & 0 \\ 0 & 0 & m_2 \end{bmatrix}, \quad C(\xi) = \begin{bmatrix} 0 & 0 & -m_1 v \\ 0 & 0 & m_1 u \\ m_1 v & -m_1 u & 0 \end{bmatrix}, \tag{2}$$

$$D = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix}, \quad J(\eta) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where M is the inertia matrix, D is the hydrodynamic damping coefficient matrix, $\xi = [u, v, r]^{\mathrm{T}}$ represents the surge, sway velocity and yaw angular velocity of the hovercraft in the vehicle-fixed reference frame. $\eta = [x, y, \psi]^{\mathrm{T}}$ represents the coordinates of the hovercraft's mass center and yaw angles of the hovercraft in the earth-fixed reference frame respectively. $\tau = [\tau_1, \tau_2, \tau_3]^{\mathrm{T}}$ represents the surge, sway and yaw moment provided by the air propeller.

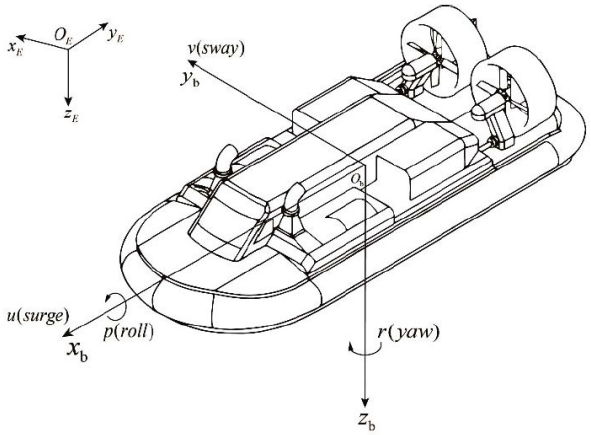


Figure 2 A brief picture of under-actuated hovercraft dynamic[33]

Remark 2.1 At the stern of the hovercraft, there are two same air propellers and two same air rudders, which provide the surge and yaw moment respectively as shown in Figure 2. Considering the structure of air propellers, to simplify the model, we ignore the sway moment, that is $\tau_2 = 0$. According to [32–34], compared to d_2, the hydrodynamic damping coefficients d_1, d_3 are very small, to simplify the model, we ignore d_1 and d_3, that is $d_1 = d_3 = 0$. The

above simplifications may not be strictly true in actual situations, which will cause modeling errors. But it is still acceptable because ADRC can estimate modeling errors and compensate them[19].

Finally, we get a simplified hovercraft model.

$$\begin{cases} \dot{x} = u\cos(\psi) - v\sin(\psi), \\ \dot{y} = u\sin(\psi) + v\cos(\psi), \\ \dot{\psi} = r, \\ \dot{u} = vr + \tau_u, \\ \dot{v} = -ur - \beta v, \\ \dot{r} = \tau_r, \end{cases} \tag{4}$$

where $\tau_u = \frac{\tau_1}{m_1}, \tau_r = \frac{\tau_3}{m_3}$ and $\beta = \frac{d_2}{m_2}$, τ_u and τ_r represent surge and yaw control inputs respectively. x, y, ψ represent x coordinate, y coordinate, yaw angle of the hovercraft in the earth coordinate system respectively and they are also the controlled variables. Therefore, the hovercraft model (4) is a two-input and three-output under-actuated system.

Remark 2.2 Considering the actual physical meaning of variables, in this paper u, v, r are bounded; ψ represents yaw angle of the hovercraft and satisfies $-\frac{\pi}{2} < \psi < \frac{\pi}{2}$; τ_u and τ_r represent surge and yaw control inputs associated with the moment of the air propeller, so τ_u, τ_r are bounded and their derivatives $\dot{\tau}_u, \dot{\tau}_r$ are also bounded.

To simplify the design of the controller, in Section 3, we will employ differential flatness method to find a set of flat outputs and convert the under-actuated system into a full-actuated one.

3 Differential Flatness System and Flat Outputs

3.1 Differential Flatness Theory

Flatness is a property of nonlinear systems, and a system with flatness property is called a flat system. For a flat system, a set of flat outputs can be found, if the system state and control input can be represented by a combination of flat outputs and their finite-order derivatives. The specific definition of the flat system is introduced below.

Consider a nonlinear system

$$\begin{cases} \dot{\chi} = f(\chi, u), & \chi \in R^n, \quad u \in R^m, \\ y = g(\chi), & y \in R^n, \end{cases} \tag{5}$$

where χ is the state variable, u is the input variable, y is the output variable and note that $n > m$.

For the system (5), if there exist smooth functions $\zeta(\cdot)$, $\sigma(\cdot)$, $\kappa(\cdot)$ and output vector F

$$F = \zeta\left(\chi, u, \dot{u}, \cdots, u^{(k)}\right), \quad k \in N, \tag{6}$$

so that the system state χ and control input u can be expressed as

$$\begin{aligned} \xi &= \sigma\left(F, \dot{F}, \cdots, F^{(k_x)}\right), \quad k_x \in N, \\ u &= \kappa\left(F, \dot{F}, \cdots, F^{(k_u)}\right), \quad k_u \in N, \end{aligned} \tag{7}$$

then the system (5) can be regarded as a differential flatness system and F is called a set of flat outputs. Considering F as the new output of the system, if the dimension of F is equal to the dimension of control inputs u, then system (5) can be regarded as a full-actuated system. However, the most important and difficult issue is to find proper flat outputs that satisfy (6) and (7), since no systematic method exists.

3.2 Flat Outputs and Proof

In this paper, we choose states x and y in (4) as flat outputs, then all states and inputs of the system (4) can be expressed by finite-order derivatives of flat outputs. The proof is shown as follows.

The second derivatives of x and y in (4) are calculated as

$$\begin{aligned} \ddot{x} =& \dot{u}\cos(\psi) - u \cdot \sin(\psi) \cdot \dot{\psi} - \dot{v}\sin(\psi) - v \cdot \cos(\psi)\dot{\psi}, \\ \ddot{y} =& \dot{u}\sin(\psi) + u \cdot \cos(\psi) \cdot \dot{\psi} + \dot{v}\cos(\psi) - v \cdot \sin(\psi)\dot{\psi}. \end{aligned} \tag{8}$$

According to (8) and (4), it is determined that

$$\begin{aligned} \ddot{x} + \beta\dot{x} =& \cos(\psi)(\dot{u} - v\dot{\psi} + \beta u) + \sin(\psi)(-u\dot{\psi} - \dot{v} - \beta v), \\ \ddot{y} + \beta\dot{y} =& \cos(\psi)(\dot{v} + u\dot{\psi} + \beta v) + \sin(\psi)(-v\dot{\psi} + \dot{u} + \beta u). \end{aligned} \tag{9}$$

Substitute (4) into (9), we can simplify (9) as

$$\begin{aligned} u\dot{\psi} + \dot{v} + \beta v &= u \cdot r - ur - \beta v + \beta v = 0, \\ \dot{u} - v\dot{\psi} + \beta u &= vr + \tau_u - vr + \beta u = \tau_u + \beta u. \end{aligned} \tag{10}$$

According to (9), it is determined that

$$\frac{\ddot{y} + \beta\dot{y}}{\ddot{x} + \beta\dot{x}} = \frac{\sin(\psi)\,(\tau_u + \beta u)}{\cos(\psi)\,(\tau_u + \beta u)} = \tan(\psi) \Rightarrow \psi = \arctan\left(\frac{\ddot{y} + \beta\dot{y}}{\ddot{x} + \beta\dot{x}}\right), \tag{11}$$

$$(\ddot{x} + \beta\dot{x})^2 + (\ddot{y} + \beta\dot{y})^2 = (\tau_u + \beta u)^2 \Rightarrow \tau_u + \beta u = \sqrt{(\ddot{x} + \beta\dot{x})^2 + (\ddot{y} + \beta\dot{y})^2}. \tag{12}$$

Considering of $r = \dot{\psi}$, therefore

$$\begin{aligned} r = \left[\arctan\left(\frac{\ddot{y} + \beta\dot{y}}{\ddot{x} + \beta\dot{x}}\right)\right]' &= \frac{1}{1 + \left(\frac{\ddot{y} + \beta\dot{y}}{\ddot{x} + \beta\dot{x}}\right)^2} \cdot \left(\frac{\ddot{y} + \beta\dot{y}}{\ddot{x} + \beta\dot{x}}\right)' \\ &= \frac{y^{(3)}(\ddot{x} + \beta\dot{x}) - x^{(3)}(\ddot{y} + \beta\dot{y}) - \beta^2(\ddot{x}\dot{y} - \ddot{y}\dot{x})}{(\ddot{x} + \beta\dot{x})^2 + (\ddot{y} + \beta\dot{y})^2}. \end{aligned} \tag{13}$$

According to (12) and (4), then

$$\begin{aligned} \dot{x}(\ddot{x} + \beta\dot{x}) &= (u\cos(\psi) - v\sin(\psi))\cos(\psi)\,(\tau_u + \beta u)\,, \\ \dot{y}(\ddot{y} + \beta\dot{y}) &= (u\sin(\psi) + v\cos(\psi))\sin(\psi)\,(\tau_u + \beta u)\,, \end{aligned} \tag{14}$$

as a result

$$\dot{x}(\ddot{x}+\beta\dot{x})+\dot{y}(\ddot{y}+\beta\dot{y})=u\left(\tau_u+\beta u\right). \tag{15}$$

Divide (15) by (12), u is expressed as

$$u=\frac{\dot{x}(\ddot{x}+\beta\dot{x})+\dot{y}(\ddot{y}+\beta\dot{y})}{\sqrt{(\ddot{x}+\beta\dot{x})^2+(\ddot{y}+\beta\dot{y})^2}}. \tag{16}$$

Combining (4) and (8), it is determined that

$$\begin{aligned}\dot{y}\ddot{x}-\dot{x}\ddot{y}=&[u\sin(\psi)+v\cos(\psi)][\dot{u}\cos(\psi)-u\sin(\psi)\dot{\psi}-\dot{v}\sin(\psi)-v\cos(\psi)\dot{\psi}]\\&-[u\cos(\psi)-v\sin(\psi)][\dot{u}\sin(\psi)+u\cos(\psi)\dot{\psi}+\dot{v}\cos(\psi)-v\sin(\psi)\dot{\psi}].\end{aligned} \tag{17}$$

Simplify Equation (17), it is obtained that

$$\begin{aligned}\dot{y}\ddot{x}-\dot{x}\ddot{y}&=-u^2\dot{\psi}-\dot{v}u+\dot{u}v-v^2\dot{\psi}\\&=-u^2r+(ur+\beta v)u+(vr+\tau_u)v-v^2r\\&=v(\beta u+\tau_u).\end{aligned} \tag{18}$$

Divide (18) by (12), v is expressed as

$$v=\frac{\dot{y}\ddot{x}-\dot{x}\ddot{y}}{\sqrt{(\ddot{x}+\beta\dot{x})^2+(\ddot{y}+\beta\dot{y})^2}}. \tag{19}$$

Similarly, according to (13), (16) and (19), τ_u is expressed as

$$\tau_u=\dot{u}-v\cdot r=\frac{d}{dt}\left\{\frac{\dot{x}(\ddot{x}+\beta\dot{x})+\dot{y}(\ddot{y}+\beta\dot{y})}{\sqrt{(\ddot{x}+\beta\dot{x})^2+(\ddot{y}+\beta\dot{y})^2}}\right\}=\frac{\ddot{x}(\ddot{x}+\beta\dot{x})+\ddot{y}(\ddot{y}+\beta\dot{y})}{\sqrt{(\ddot{x}+\beta\dot{x})^2+(\ddot{y}+\beta\dot{y})^2}}. \tag{20}$$

Finally, because of $\tau_r=\dot{r}$, according to (13), τ_r is expressed as

$$\begin{aligned}\tau_r=\dot{r}=&\frac{y^{(4)}(\ddot{x}+\beta x)-x^{(4)}(\ddot{y}+\beta\dot{y})+\beta\left(y^{(3)}\ddot{x}-x^{(3)}\dot{y}\right)-\beta^2\left(x^{(3)}\dot{y}-y^{(3)}\dot{x}\right)}{[(\ddot{x}+\beta\dot{x})^2+(\ddot{y}+\beta\dot{y})^2]}\\&-2\frac{\left[y^{(3)}(\ddot{x}+\beta\dot{x})-x^3(\ddot{y}+\beta\dot{y})-\beta^2(\ddot{x}\dot{y}-\ddot{y}\dot{x})\right]}{[(\ddot{x}+\beta\dot{x})^2+(\ddot{y}+\beta\dot{y})^2]^2}\\&\cdot\left\{(\ddot{x}+\beta\dot{x})\left(x^{(3)}+\beta\ddot{x}\right)+(\ddot{y}+\beta\dot{y})\left(y^{(3)}+\beta\ddot{y}\right)\right\}.\end{aligned} \tag{21}$$

Therefore, this system (4) is a flat system and the proof is completed. ∎

3.3 Full-Actuated Hovercraft Model

In this section, we establish the full-actuated model of the hovercraft. τ_u,τ_r are inputs and x,y are outputs .

Considering the first two equations in the model (4)

$$\begin{cases}\dot{x}=u\cos(\psi)-v\sin(\psi),\\ \dot{y}=u\sin(\psi)+v\cos(\psi).\end{cases} \tag{22}$$

Since Equation (22) does not contain τ_u and τ_r, we take the derivative of x and y continuously, until τ_u and τ_r appear simultaneously.

After continuous derivation, the fourth derivatives of x and y meet the condition and results are as follows.

$$\begin{cases} x^{(4)} = \cos(\psi)\ddot{\tau}_u + \left[-\tau_u \sin(\psi) - \beta u \sin(\psi) + \beta v \cos(\psi)\right] \tau_r + \varphi_x, \\ y^{(4)} = \sin(\psi)\ddot{\tau}_u + \left[\tau_u \cos(\psi) + \beta u \cos(\psi) + \beta v \sin(\psi)\right] \tau_r + \varphi_y, \\ \varphi_x = \{-2\dot{\tau}_u \sin(\psi) \cdot r - \tau_u r^2 \cos(\psi) - \beta \tau_u r \sin(\psi) - 2\beta v r^2 \sin(\psi) - 2\beta u r^2 \cos(\psi) \\ \qquad + \beta^2 u r \sin(\psi) - 2\beta^2 v r \cos(\psi) + \beta^3 v \sin(\psi)\}, \\ \varphi_y = \{2\dot{\tau}_u \cos(\psi) \cdot r - \tau_u r^2 \sin(\psi) + \beta \tau_u r \cos(\psi) + 2\beta v r^2 \cos(\psi) - 2\beta u r^2 \sin(\psi) \\ \qquad - \beta^2 u r \cos(\psi) - 2\beta^2 v r \sin(\psi) - \beta^3 v \cos(\psi)\}, \end{cases} \tag{23}$$

where φ_x and φ_y contain complex nonlinear terms, which can be considered as disturbances to be estimated. Note that, according to Remark 2.2, φ_x and φ_y are both bounded. Then the full-actuated hovercraft model can be expressed in a compact form as

$$\begin{aligned} \begin{bmatrix} x^{(4)} \\ y^{(4)} \end{bmatrix} &= \begin{bmatrix} -(\beta u + \tau_u)\sin\psi + \beta v \cos\psi & \cos\psi \\ (\beta u + \tau_u)\cos\psi + \beta v \sin\psi & \sin\psi \end{bmatrix} \begin{bmatrix} \tau_r \\ \ddot{\tau}_u \end{bmatrix} + \begin{bmatrix} \varphi_x \\ \varphi_y \end{bmatrix} \\ &= \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} + \begin{bmatrix} \varphi_x \\ \varphi_y \end{bmatrix}, \end{aligned} \tag{24}$$

where U_1, U_2 are defined as virtual control inputs and can be written as follows:

$$\begin{cases} U_1 = (-(\beta u + \tau_u)\sin\psi + \beta v \cos\psi)\tau_r + \cos\psi \ddot{\tau}_u, \\ U_2 = ((\beta u + \tau_u)\cos\psi + \beta v \sin\psi)\tau_r + \sin\psi \ddot{\tau}_u. \end{cases} \tag{25}$$

Assume that:

$$\begin{aligned} &x_1 = x, \quad x_2 = \dot{x}, \quad x_3 = \ddot{x}, \quad x_4 = x^{(3)}, \quad x_5 = \varphi_x, \\ &y_1 = y, \quad y_2 = \dot{y}, \quad y_3 = \ddot{y}, \quad y_4 = y^{(3)}, \quad y_5 = \varphi_y, \end{aligned} \tag{26}$$

therefore, the model (24) can be rewritten as a chained form.

$$\begin{cases} \dot{x_1} = x_2, \\ \dot{x_2} = x_3, \\ \dot{x_3} = x_4, \\ \dot{x_4} = U_1 + x_5, \\ \dot{x_5} = \varepsilon_x, \end{cases} \qquad \begin{cases} \dot{y_1} = y_2, \\ \dot{y_2} = y_3, \\ \dot{y_3} = y_4, \\ \dot{y_4} = U_2 + y_5, \\ \dot{y_5} = \varepsilon_y, \end{cases} \tag{27}$$

where ε_x and ε_y are derivatives of φ_x and φ_x respectively. Note that, according to Remark 2.2, ε_x and ε_y are both bounded in practice.

In summary, in this section, flat outputs x, y are found and proved by differential flatness theory, then a full-actuated hovercraft model (27) is built. In Section 4, a trajectory tracking control scheme will be designed based on ADRC and RL.

4 Optimal Trajectory Tracking Control Based on ADRC and RL

4.1 Problem Formulation

Assuming the sampling time is h, the model (27) can be discretized in form of

$$\begin{cases} x_1(k+1) = x_1(k) + hx_2(k), \\ x_2(k+1) = x_2(k) + hx_3(k), \\ x_3(k+1) = x_3(k) + hx_4(k), \\ x_4(k+1) = x_4(k) + h(U_1(k) + x_5(k)), \\ x_5(k+1) = x_5(k) + \varepsilon_x(k), \end{cases} \quad \begin{cases} y_1(k+1) = y_1(k) + hy_2(k), \\ y_2(k+1) = y_2(k) + hy_3(k), \\ y_3(k+1) = y_3(k) + hy_4(k), \\ y_4(k+1) = y_4(k) + h(U_1(k) + y_5(k)), \\ y_5(k+1) = y_5(k) + \varepsilon_y(k), \end{cases} \tag{28}$$

where k means the k-th sampling period. Define a reference trajectory sequence $r^*(k) = [x^*(k), y^*(k)]^{\mathrm{T}}$ $(k = 1, 2, \cdots, N)$. For the optimal trajectory tracking problem, the tracking error is defined as

$$\begin{cases} e_x(k) = x(k) - x^*(k), \\ e_y(k) = y(k) - y^*(k). \end{cases} \tag{29}$$

The goal is to design an optimal trajectory tracking controller for the hovercraft model (28) to ensure the tracking error converge to zero in an optimal manner, that is,

$$\begin{cases} \lim\limits_{k \to N} e_x(k) = 0, \\ \lim\limits_{k \to N} e_y(k) = 0. \end{cases} \tag{30}$$

Considering the complex nonlinear disturbances in (28), an active disturbance rejection controller is designed to estimate disturbances and compensate them in real-time. Generally, ADRC consists of three parts, namely tracking differentiator (TD), extended state observer (ESO), and feedback control strategy with disturbances compensation. For the general feedback control strategy[23–25] in ADRC, control parameters are adjusted manually which is complicated and the optimality cannot be guaranteed. Especially for the model (28), as a fourth-order system, there are more control parameters that need to be adjusted, which is difficult to guarantee optimality and adaptability. Therefore, a reinforcement learning algorithm is used to approximate the optimal strategy.

4.2 Active Disturbance Rejection Controller Design

Considering the four-order hovercraft model (28), to simplify the design process, linear ADRC is selected. Firstly, a tracking differentiator is designed as follows to arrange the tran-

sition process.

$$\begin{cases} x_1^*(k+1) = x_1^*(k) + hx_2^*(k), \\ x_2^*(k+1) = x_2^*(k) + hx_3^*(k), \\ x_3^*(k+1) = x_3^*(k) + hx_4^*(k), \\ x_4^*(k+1) = x_4^*(k) + f_x(k), \\ f_x(k+1) = -r_1(r_1(r_1(r_1(x_1^*(k) - x^*(k+1)) \\ \qquad\qquad +4x_2^*(k)) + 6x_3^*(k)) + 4x_4^*(k)), \end{cases} \tag{31}$$

$$\begin{cases} y_1^*(k+1) = y_1^*(k) + hy_2^*(k), \\ y_2^*(k+1) = y_2^*(k) + hy_3^*(k), \\ y_3^*(k+1) = y_3^*(k) + hy_4^*(k), \\ y_4^*(k+1) = y_4^*(k) + f_y(k), \\ f_y(k+1) = -r_2(r_2(r_2(r_2(y_1^*(k) - y^*(k+1)) \\ \qquad\qquad +4y_2^*(k)) + 6y_3^*(k)) + 4y_4^*(k)), \end{cases}$$

where $x_1^*(k), x_2^*(k), x_3^*(k), x_4^*(k), f_x(k)$ are approximations of $x^*, \dot{x}^*, \ddot{x}^*, \cdots$ and $y_1^*(k), y_2^*(k), y_3^*(k)$, $y_4^*(k), f_y(k)$ are approximations of $y^*, \dot{y}^*, \ddot{y}^*, \cdots$ respectively, r_1 and r_2 are TD parameters.

Then, an extended state observer is designed to track system states and estimate disturbances φ_x and φ_y.

$$\begin{cases} e_1(k) = z_{11}(k) - x_1(k), \\ z_{11}(k) = z_{11}(k) + h(z_{12}(k) - b_{11}e_1(k)), \\ z_{12}(k) = z_{12}(k) + h(z_{13}(k) - b_{12}e_1(k)), \\ z_{13}(k) = z_{13}(k) + h(z_{14}(k) - b_{13}e_1(k)), \\ z_{14}(k) = z_{14}(k) + h(z_{15}(k) - b_{14}e_1(k) + U_1(k)), \\ z_{15}(k) = z_{15}(k) + h(-b_{15}e_1(k)), \end{cases} \tag{32}$$

$$\begin{cases} e_2(k) = z_{21}(k) - y_1(k), \\ z_{21}(k) = z_{21}(k) + h(z_{22}(k) - b_{21}e_2(k)), \\ z_{22}(k) = z_{22}(k) + h(z_{23}(k) - b_{22}e_2(k)), \\ z_{23}(k) = z_{23}(k) + h(z_{24}(k) - b_{23}e_2(k)), \\ z_{24}(k) = z_{24}(k) + h(z_{25}(k) - b_{24}e_2(k) + U_2(k)), \\ z_{25}(k) = z_{25}(k) + h(-b_{25}e_2(k)), \end{cases}$$

where $z_{12}, z_{13}, z_{14}, z_{15}$ are approximations of x_2, x_3, x_4, x_5 in (28) and $z_{22}, z_{23}, z_{24}, z_{25}$ are approximations of y_2, y_3, y_4, y_5 in (28) respectively. And $b_{11}, b_{12}, \cdots, b_{25}$ are ESO parameters which are determined by bandwidth-parameterization method[26].

Thirdly, based on output results of TD and ESO, calculate tracking errors and design the feedback control law as below:

$$\begin{cases} e_{11}(k) = x_1^*(k) - z_{11}(k), \\ e_{12}(k) = x_2^*(k) - z_{12}(k), \\ e_{13}(k) = x_3^*(k) - z_{13}(k), \\ e_{14}(k) = x_4^*(k) - z_{14}(k), \\ U_{10}(k) = \phi_x(e_{11}(k), e_{12}(k), e_{13}(k), e_{14}(k)), \end{cases} \tag{33}$$

$$\begin{cases} e_{21}(k) = y_1^*(k) - z_{21}(k), \\ e_{22}(k) = y_2^*(k) - z_{22}(k), \\ e_{23}(k) = y_3^*(k) - z_{23}(k), \\ e_{24}(k) = y_4^*(k) - z_{24}(k), \\ U_{20}(k) = \phi_y(e_{21}(k), e_{22}(k), e_{23}(k), e_{24}(k)), \end{cases}$$

where ϕ_x and ϕ_y are nonlinear or linear functions of feedback control strategy. Note that in traditional linear ADRC, ϕ_x and ϕ_y are linear functions of the form as follows:

$$\begin{cases} \phi_x = k_{11}e_{11}(k) + k_{12}e_{12}(k) + k_{13}e_{13}(k) + k_{14}e_{14}(k), \\ \phi_y = k_{21}e_{21}(k) + k_{22}e_{22}(k) + k_{23}e_{23}(k) + k_{24}e_{24}(k), \end{cases} \tag{34}$$

where $k_{11}, k_{12}, \cdots, k_{24}$ are control parameters. However, it is extremely difficult to choose a set of robust, optimal and adaptive parameters. Therefore, optimal nonlinear functions ϕ_x and ϕ_y will be determined by RL method in Subsection 4.3. Then, compensating the disturbance, virtual control inputs $U_1(k)$ and $U_2(k)$ are defined as follows:

$$\begin{cases} U_1(k) = U_{10}(k) - z_{15}(k), \\ U_2(k) = U_{20}(k) - z_{25}(k). \end{cases} \tag{35}$$

Finally, real control inputs of the hovercraft τ_u, τ_r are solved as follows

$$\begin{bmatrix} \tau_r(k) \\ \ddot{\tau}_u(k) \end{bmatrix} = \begin{bmatrix} -(\beta u(k) + \tau_u(k))\sin(\psi(k)) + \beta v(k)\cos(\psi(k)) & \cos(\psi(k)) \\ (\beta u(k) + \tau_u(k))\cos(\psi(k)) + \beta v(k)\sin(\psi(k)) & \sin(\psi(k)) \end{bmatrix}^{-1} \cdot \begin{bmatrix} U_1(k) \\ U_2(k) \end{bmatrix}. \tag{36}$$

4.3 Optimal Control Based on RL

Define the cost function of tracking problem as

$$\begin{cases} J_x(e_x(k), U_{10}(k)) = \sum_{i=k}^{N} [e_x^{\mathrm{T}}(i) Q e_x(i) + U_{10}^{\mathrm{T}}(i) R U_{10}(i)], \\ J_y(e_y(k), U_{20}(k)) = \sum_{i=k}^{N} [e_y^{\mathrm{T}}(i) Q e_y(i) + U_{20}^{\mathrm{T}}(i) R U_{20}(i)], \end{cases} \tag{37}$$

where $e_x(k) = [e_{11}(k), e_{12}(k), e_{13}(k), e_{14}(k)]^{\mathrm{T}}$, $e_y(k) = [e_{21}(k), e_{22}(k), e_{23}(k), e_{24}(k)]^{\mathrm{T}}$, Q and R are positive definite weights matrices. The goal of the optimal control is to find optimal control sequences U_{10}^* and U_{20}^* to minimize the cost function (37).

According to the optimal control theory[35], define the value functions $V_x(k)$ and $V_y(k)$ as follows:

$$\begin{cases} V_x(k) = \sum\limits_{i=k}^{N} \gamma^{i-k}[e_x^{\mathrm{T}}(i)Qe_x(i) + U_{10}^{\mathrm{T}}(i)RU_{10}(i)], \\ V_y(k) = \sum\limits_{i=k}^{N} \gamma^{i-k}[e_y^{\mathrm{T}}(i)Qe_y(i) + U_{20}^{\mathrm{T}}(i)RU_{20}(i)], \end{cases} \tag{38}$$

where $0 < \gamma \leq 1$ is discount factor, and we require the boundary condition $V_x(0) = V_y(0) = 0$. From Bellmans optimality principle[36], the optimal control problem equivalent to find optimal value functions $V_x^*(k)$ and $V_y^*(k)$ which satisfy the Bellman equation

$$\begin{cases} V_x^*(k) = [e_x^{\mathrm{T}}(k)Qe_x(k) + U_{10}^{\mathrm{T}}(k)RU_{10}(k)] + \gamma V_x^*(k+1), \\ V_y^*(k) = [e_y^{\mathrm{T}}(k)Qe_y(k) + U_{20}^{\mathrm{T}}(k)RU_{20}(k)] + \gamma V_y^*(k+1). \end{cases} \tag{39}$$

The optimal control strategy can be determined as follows[35]:

$$\begin{cases} U_{10}^*(k) = \dfrac{1}{2}R^{-1}\dfrac{\partial V_x^*(k+1)}{\partial e_x(k+1)}, \\ U_{20}^*(k) = \dfrac{1}{2}R^{-1}\dfrac{\partial V_y^*(k+1)}{\partial e_y(k+1)}. \end{cases} \tag{40}$$

Generally, it is difficult to calculate the solution of Equation (39) directly, therefore, an actor-critic neural networks (NNs) structure is established to approximate $U_{10}^*, U_{20}^*, V_x^*$ and V_y^*. Define the critic neural network (NN) as follows

$$\widehat{V}_x(k) = \widehat{W}_{cx}^{\mathrm{T}}\phi_{cx}(e_x(k)), \qquad \widehat{V}_y(k) = \widehat{W}_{cy}^{\mathrm{T}}\phi_{cy}(e_y(k)), \tag{41}$$

where ϕ_{cx} and ϕ_{cy} are activation functions, $\widehat{W}_{cx}$ and $\widehat{W}_{cy}$ are weight matrices. $\widehat{V}_x$ and $\widehat{V}_y$ are outputs of the critic NN which is supposed to approximate to optimal value functions V_x^* and V_y^* by constant iteration. Similarly, the actor neural network (NN) is established, outputs are given as $\widehat{U}_{10}$ and $\widehat{U}_{20}$ which is supposed to approximate to U_{10}^* and U_{20}^*

$$\widehat{U}_{10}(k) = \widehat{W}_{U10}^{\mathrm{T}}\phi_{U10}(e(k)), \qquad \widehat{U}_{20}(k) = \widehat{W}_{U20}^{\mathrm{T}}\phi_{U20}(e(k)), \tag{42}$$

where ϕ_{U10} and ϕ_{U20} are activation functions, $\widehat{W}_{U10}$ and $\widehat{W}_{U20}$ are weights matrices.

We train the critic NN and actor NN to approximate the optimal value function and the optimal control strategy respectively. In other words, the critic NN should satisfy Equation (39) and the actor NN should satisfy Equation (40). Therefore, critic fitting errors are defined as

$$\begin{cases} e_{cx(i)}(k) = [e_x^{\mathrm{T}}(k)Qe_x(k) + \widehat{U}_{10}^{\mathrm{T}}(k)R\widehat{U}_{10}(k)] + \gamma\widehat{V}_x(k+1) - \widehat{V}_x(k), \\ e_{cy(i)}(k) = [e_y^{\mathrm{T}}(k)Qe_y(k) + \widehat{U}_{20}^{\mathrm{T}}(k)R\widehat{U}_{20}(k)] + \gamma\widehat{V}_y(k+1) - \widehat{V}_y(k), \end{cases} \tag{43}$$

where i represents the training iteration time, k represents the discrete-time under current training iteration time. And square critic fitting errors are defined as

$$E_{cx(i)}(k) = \frac{1}{2}e^2_{cx(i)}(k), \qquad E_{cy(i)}(k) = \frac{1}{2}e^2_{cy(i)}(k). \tag{44}$$

In order to minimize square critic fitting errors $E_{cx(i)}(k)$ and $E_{cy(i)}(k)$, weights matrices in the critic NN are updated by gradient descent. The weights learning rules for critic NN are selected as follows:

$$\begin{aligned} \widehat{W}_{cx(i+1)}(k) &= \widehat{W}_{cx(i)}(k) - r_{cx}\frac{\partial E_{cx(i)}(k)}{\partial \widehat{W}_{cx(i)}(k)}, \\ \widehat{W}_{cy(i+1)}(k) &= \widehat{W}_{cy(i)}(k) - r_{cy}\frac{\partial E_{cy(i)}(k)}{\partial \widehat{W}_{cy(i)}(k)}, \end{aligned} \tag{45}$$

where $r_{cy} > 0$ and $r_{cx} > 0$ are learning rates.

Similarly, action fitting errors and their square are defined as

$$\begin{cases} e_{U_{10}(i)}(k) = \dfrac{1}{2}R^{-1}\dfrac{\partial \widehat{V}_x(k+1)}{\partial e_x(k+1)} - \widehat{U}_{10}(k), \\ E_{U_{10}(i)}(k) = \dfrac{1}{2}e^2_{U_{10}(i)}(k), \\ e_{U_{20}(i)}(k) = \dfrac{1}{2}R^{-1}\dfrac{\partial \widehat{V}_y(k+1)}{\partial e_y(k+1)} - \widehat{U}_{20}(k), \\ E_{U_{20}(i)}(k) = \dfrac{1}{2}e^2_{U_{20}(i)}(k). \end{cases} \tag{46}$$

The weights learning rules for actor NN are selected as follows:

$$\begin{cases} \widehat{W}_{U10(i+1)}(k) = \widehat{W}_{U10(i)}(k) - r_{U_{10}}\dfrac{\partial E_{U_{10}(i)}(k)}{\partial \widehat{W}_{U10(i)}(k)}, \\ \widehat{W}_{U20(i+1)}(k) = \widehat{W}_{U20(i)}(k) - r_{U_{20}}\dfrac{\partial E_{U_{20}(i)}(k)}{\partial \widehat{W}_{U20(i)}(k)}, \end{cases} \tag{47}$$

where $r_{U_{10}} > 0$ and $r_{U_{20}} > 0$ are learning rates. Note that the stability and convergence proof of the actor-critic NNs during the training process are provided in [37, 38].

Assumed that N indicates the maximum iteration time and ρ indicates the admissible approximation error. We stop the iteration of critic NN when $E_{cx(i)}(k) < \rho, E_{cy(i)}(k) < \rho$ or $i > N$. Under this condition, $\widehat{U}_{10}, \widehat{U}_{20}$ approximate to optimal controller U^*_{10} and U^*_{20}.

5 Simulation and Result

In this section, simulation results are provided to verify the effectiveness of the control scheme proposed in this paper. In the simulation, the sampling time $h = 0.02$ s, total simulation time $t = 10.5$ s, TD parameters $r_1 = 18$ and $r_2 = 23$. Based on bandwidth-parameterization

method, ESO parameters are selected as follows:

$$\begin{cases} b_{11} = 5\sigma, & b_{12} = 10\sigma^2, & b_{13} = 10\sigma^3, & b_{14} = 5\sigma^4, & b_{15} = \sigma^5, \\ b_{21} = 5\upsilon, & b_{22} = 10\upsilon^2, & b_{23} = 10\upsilon^3, & b_{24} = 5\upsilon^4, & b_{25} = \upsilon^5, \end{cases} \tag{48}$$

where $\sigma = 45 = \upsilon = 45$. For the hovercraft model (4), we assume $\tau_u = 0.8$ and $\tau_r = 0.2$, then reference trajectory and yaw angle are generated as shown in Figure 3.

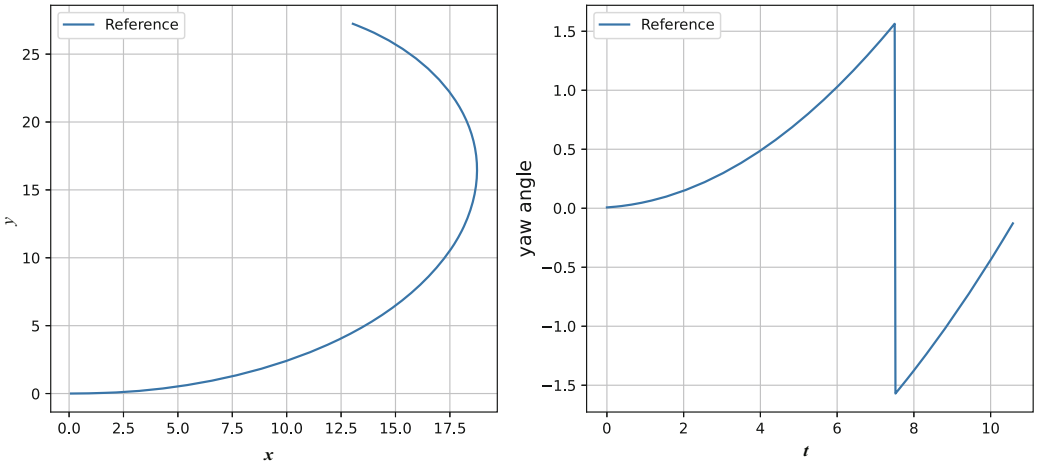


Figure 3 Reference trajectory and yaw angle

We choose fully connected NN as actor NN and critic NN with the structures of 4-16-1 (4 input nodes, 16 hidden nodes and 1 output node) and 4-16-4 (4 input nodes, 16 hidden nodes and 4 output nodes). The discount factor $\gamma = 0.95$, learning rates $r_{cy} = r_{cx} = r_{U_{10}} = r_{U_{20}} = 0.1$, maximum iteration times $N = 30$, admissible approximation error $\rho = 10^{-5}$, weights matrices Q and R are chosen as

$$Q = \begin{bmatrix} 40 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 \\ 0 & 0 & 30 & 0 \\ 0 & 0 & 0 & 20 \end{bmatrix}, \qquad R = 1. \tag{49}$$

With initial tracking errors $e_x = e_y = 0$, actor NN is trained to approximate to optimal control strategies U_{10}^* and U_{20}^*. With the same initial states, after a long period of manually tuning, a set of control parameters of linear ADRC feedback control strategy (34) is determined as follows:

$$\begin{cases} k_{11} = -9, & k_{12} = -2.16, & k_{13} = -13.6, & k_{14} = -10.8, \\ k_{21} = -10.2, & k_{22} = -4.3, & k_{23} = -15, & k_{24} = -14.7. \end{cases} \tag{50}$$

With the initial tracking errors $e_x = e_y = 0$, the tracking performance of ADRC and RL-ADRC is shown in Figure 4. We assume that there is an external disturbance at $t = 1.6$ s which causes the tracking errors $e_x = 1.2$ and $e_y = 0.5$ at $t = 1.6$ s. As shown in Figures 5 and 6,

RL-ADRC is more adaptive and robust than ADRC when there exists an external disturbance at $t = 1.6$ s. Figures 8 and 9 show that, compared with ADRC, the robustness of RL-ADRC is more obvious with tracking errors increasing.

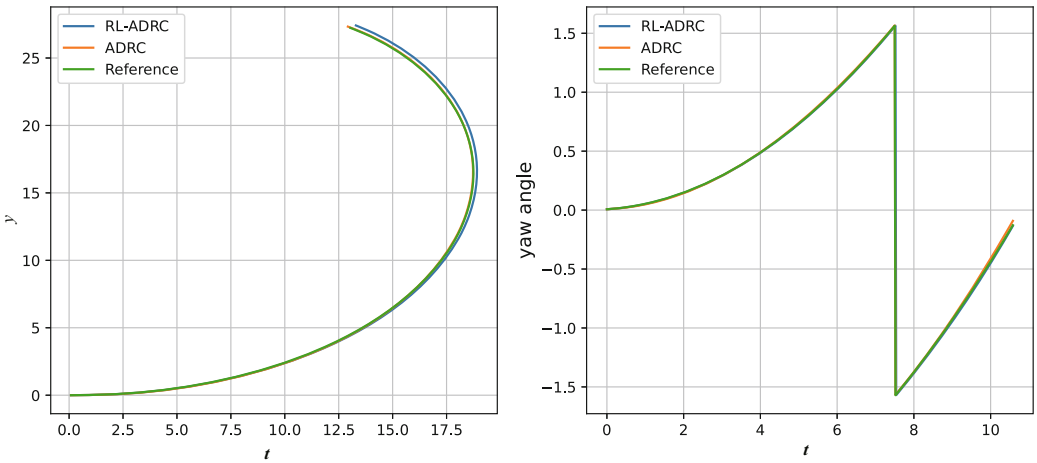


Figure 4 The tracking performance of ADRC and RL-ADRC with initial tracking errors $e_x = 0$ and $e_y = 0$

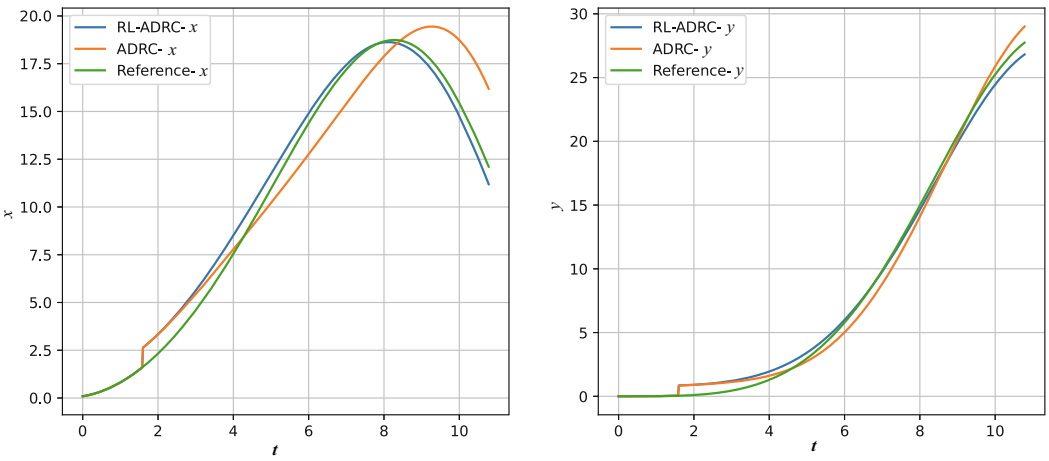


Figure 5 The tracking performance of ADRC and RL-ADRC with tracking errors $e_x = 1.2$ and $e_y = 0.5$ at $t = 1.6$ s

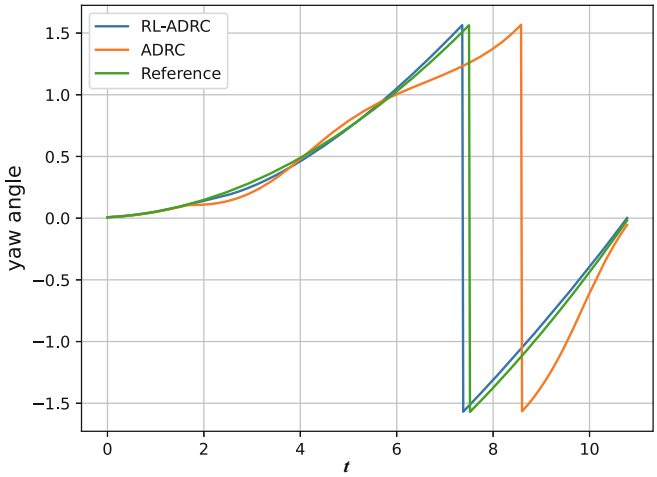


Figure 6 The yaw angle of ADRC and RL-ADRC with tracking errors $e_x = 1.2$ and $e_y = 0.5$ at $t = 1.6$ s

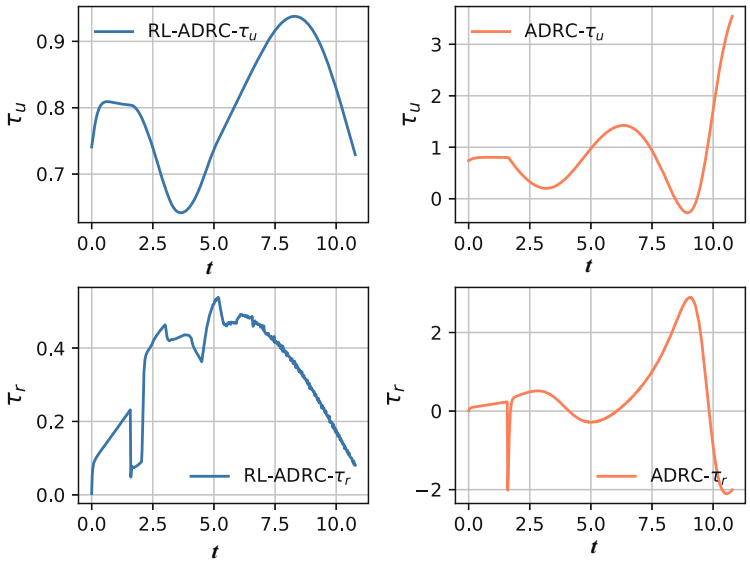


Figure 7 The control input τ_u and τ_r of ADRC and RL-ADRC with tracking errors $e_x = 1.2$ and $e_y = 0.5$ at $t = 1.6$ s

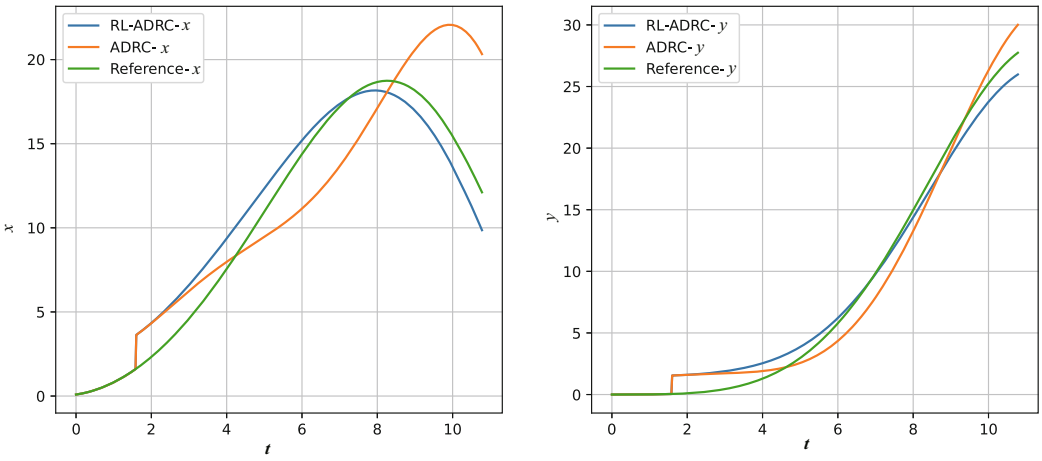


Figure 8 The tracking performance of ADRC and RL-ADRC with tracking errors $e_x = 2$ and $e_y = 1.5$ at $t = 1.6$ s

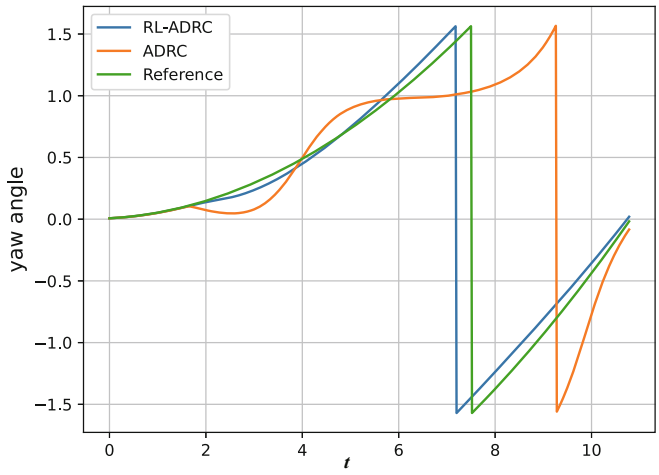


Figure 9 The yaw angle of ADRC and RL-ADRC with tracking errors $e_x = 2$ and $e_y = 1.5$ at $t = 1.6$ s

Figures 7 and 10 show that control inputs of RL-ADRC are closer to the assumed inputs $\tau_u = 0.8$ and $\tau_r = 0.2$ which generate the reference trajectory and yaw angle. On the contrary, control inputs of ADRC are oscillating and sensitive to the tracking error.

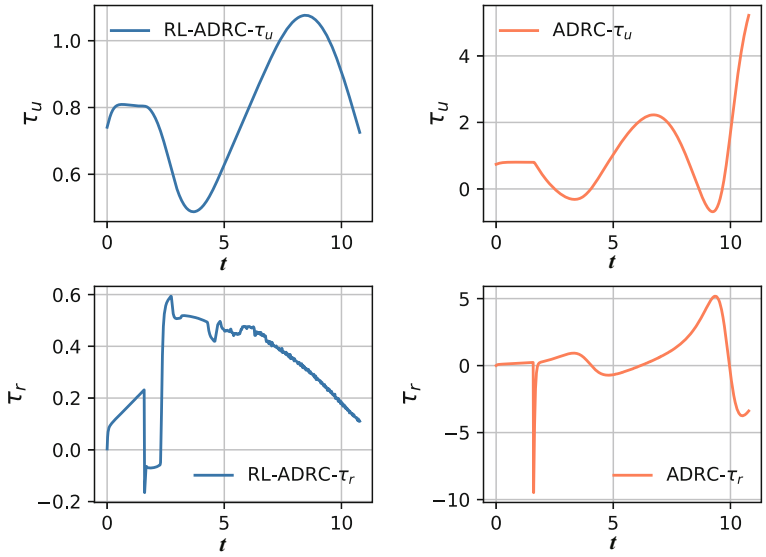


Figure 10 The control input τ_u and τ_r of ADRC and RL-ADRC with tracking errors $e_x = 2$ and $e_y = 1.5$ at $t = 1.6$ s

6 Conclusion

In this paper, a hovercraft trajectory tracking control scheme is proposed. Firstly, the underactuated model is converted to a full-actuated one by differential flatness method. Then an RL-ADRC controller is designed to generate the optimal control strategy and compensate for the total uncertainties simultaneously. Simulation studies have demonstrated that, compared with traditional linear ADRC, RL-ADRC is more robust and adaptive.

References

[1] Okafor B E, Development of a hovercraft prototype, *International Journal of Engineering and Technology*, 2013, **3**(3): 276–281.

[2] Jeong S and Chwa D, Coupled multiple sliding-mode control for robust trajectory tracking of hovercraft with external disturbances, *IEEE Transactions on Industrial Electronics*, 2017, **65**(5): 4103–4113.

[3] Jiang P, Global tracking control of underactuated ships by Lyapunov's direct method, *Automatica*, 2002, **38**(2): 301–309.

[4] Do D, Practical control of underactuated ships, *Ocean Engineering*, 2010, **37**(13): 1111–1119.

[5] Do D, Jiang P, and Pan J, Underactuated ship global tracking under relaxed conditions, *IEEE Transactions on Automatic Control*, 2002, **47**(9): 1529–1536.

[6] Dong Z, Wan L, Li Y, et al., Trajectory tracking control of underactuated USV based on modified backstepping approach., *International Journal of Naval Architecture and Ocean Engineering*, 2015, **7**(5): 817–832.

[7] Fu H, Analysis and consideration on safety of all-lift hovercraft, *Ship & Boat*, 2008, **6**: 1–3.

[8] Fu M, Gao S, Wang C, et al., Human-centered automatic tracking system for underactuated hovercraft based on adaptive chattering-free full-order terminal sliding mode control, *IEEE Access*, 2018, **6**: 37883–37892.

[9] Duan G, High-order fully actuated system approaches: Part I. Models and basic procedure, *International Journal of Systems Science*, 2021, **52**(2): 422–435.

[10] Duan G, High-order fully actuated system approaches: Part II. Generalized strict-feedback systems. *International Journal of Systems Science*, 2021, **52**(3): 437–454.

[11] Duan G, High-order fully actuated system approaches: Part III. Robust control and high-order backstepping. *International Journal of Systems Science*, 2021, **52**(5): 952–971.

[12] Tee K P and Ge S S, Control of fully actuated ocean surface vessels using a class of feedforward approximators, *IEEE Transactions on Control Systems Technology*, 2006, **14**(4): 750–756.

[13] Zheng P, Tan X, Kocer B, et al., TiltDrone: A fully-actuated tilting quadrotor platform, *IEEE Robotics and Automation Letters*, 2020, **5**(4): 6845–6852.

[14] Zhao Z, He W, and Ge S S, Adaptive neural network control of a fully actuated marine surface vessel with multiple output constraints, *IEEE Transactions on Control Systems Technology*, 2013, **22**(4): 1536–1543.

[15] Martin P and Rouchon P, Any (controllable) driftless system with m inputs and $m+2$ states is flat. *Proceedings of* 1995 34*th IEEE Conference on Decision and Control*, 1995, **3**: 2886–2891.

[16] Ma D, Xia Y, Shen G, et al., Flatness-based adaptive sliding mode tracking control for a quadrotor with disturbances, *Journal of the Franklin Institute*, 2018, **355**(14): 6300–6322.

[17] Yu Y and Lippiello V, 6D pose task trajectory tracking for a class of 3D aerial manipulator from differential flatness, *IEEE Access*, 2019, **7**: 52257–52265.

[18] Xia Y, Lin M, Zhang J, et al., Trajectory planning and tracking for four-wheel steering vehicle based on differential flatness and active disturbance rejection controller, *International Journal of Adaptive Control and Signal Processing*, 2021, **35**(11): 2214–2244.

[19] Han J, From PID to active disturbance rejection control, *IEEE transactions on Industrial Electronics*, 2009, **56**(3): 900–906.

[20] Xia Y, Dai L, Fu M, et al., Application of active disturbance rejection control in tank gun control system, *Journal of the Franklin Institute*, 2014, **351**(4): 2299–2314.

[21] Li J, Xia Y, Qi X, et al., On the necessity, scheme, and basis of the linear nonlinear switching in active disturbance rejection control, *IEEE Transactions on Industrial Electronics*, 2016, **64**(2): 1425–1435.

[22] Xue W, Bai W, Yang S, et al., ADRC with adaptive extended state observer and its application to airfuel ratio control in gasoline engines, *IEEE Transactions on Industrial Electronics*, 2015, **62**(9): 5847–5857.

[23] Lotufo A, Colangelo L, Perez-Montenegro C, et al., UAV quadrotor attitude control: An ADRC-EMC combined approach, *Control Engineering Practice*, 2019, **84**: 13–22.

[24] Xia Y, Pu F, Li S, et al., Lateral path tracking control of autonomous land vehicle based on ADRC and differential flatness, *IEEE Transactions on Industrial Electronics*, 2016, **63**(5): 3091–3099.

[25] Li Z, Wei Y, Zhou X, et al., Differential flatness based ADRC scheme for underactuated fractional

order systems, *International Journal of Robust and Nonlinear Control*, 2020, **30**(7): 2832–2849.

[26] Gao Z, Scaling and bandwidth-parameterization based controller tuning, *Proceedings of the American Control Conference*, 2006, **6**: 4989–4996.

[27] Li J, Xia Y, Qi X, et al., On the necessity, scheme, and basis of the linear nonlinear switching in active disturbance rejection control, *IEEE Transactions on Industrial Electronics*, 2016, **64**(2): 1425–1435.

[28] Zhou X, Gao H, Zhao B, et al., A GA-based parameters tuning method for an ADRC controller of ISP for aerial remote sensing applications, *ISA Transactions*, 2018, **81**: 318–328.

[29] Duan J, Yi Z, Shi D, et al., Reinforcement-learning-based optimal control of hybrid energy storage systems in hybrid ACDC microgrids, *IEEE Transactions on Industrial Informatics*, 2019, **15**(9): 5355–5364.

[30] Kiumarsi B, Vamvoudakis K G, Modares H, et al., Optimal and autonomous control using reinforcement learning: A survey, *IEEE Transactions on Neural Networks and Learning Systems*, 2017, **29**(6): 2042–2062.

[31] Jiang Y, Fan J, Chai T, et al., Data-driven flotation industrial process operational optimal control based on reinforcement learning, *IEEE Transactions on Industrial Informatics*, 2017, **14**(5): 1974–1989.

[32] Fu M, Gao S, Wang C, et al., Human-centered automatic tracking system for underactuated hovercraft based on adaptive chattering-free full-order terminal sliding mode control, *IEEE Access*, 2018, **6**: 37883–37892.

[33] Fu M, Zhang T, and Ding F, Adaptive finite-time PI sliding mode trajectory tracking control for underactuated hovercraft with drift angle constraint, *IEEE Access*, 2019, **7**: 184885–184895.

[34] Sira-Ramrez H and Ibez C A, On the control of the hovercraft system, *Dynamics and control*, 2000, **10**(2): 151–163.

[35] Al-Tamimi A, Lewis F L, and Abu-Khalaf M, Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof, *IEEE Transactions on Systems, Man, and Cybernetics, Part B* (*Cybernetics*), 2008, **38**(4): 943–949.

[36] Lin F, An optimal control approach to robust control design, *International Journal of Control*, 2000, **73**(3): 177–186.

[37] Luo B, Wu H N, Huang T, et al., Data-based approximate policy iteration for affine nonlinear continuous-time optimal control design, *Automatica*, 2014, **50**(12): 3281–3290.

[38] Bian T, Jiang Y, and Jiang Z P, Adaptive dynamic programming and optimal control of nonlinear nonaffine systems, *Automatica*, 2014, **50**(10): 2624–2632.