

## Research Article

# Adaptive Navigating Control Based on the Parallel Action-Network ADHDP Method for Unmanned Surface Vessel

Zhijian Huang <sup>1,2,3</sup> Xinze Liu,<sup>1</sup> Jiayi Wen,<sup>1</sup> Guichen Zhang,<sup>1</sup> and Yihua Liu <sup>1</sup>

<sup>1</sup>Lab of Intelligent Control and Computation, Shanghai Maritime University, Shanghai 201306, China

<sup>2</sup>Department of Computing, Hong Kong Polytechnic University, Hung Hom, KL 999077, Hong Kong

<sup>3</sup>Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881, USA

Correspondence should be addressed to Zhijian Huang; [zjhuang@shmtu.edu.cn](mailto:zjhuang@shmtu.edu.cn) and Yihua Liu; [liuyh@shmtu.edu.cn](mailto:liuyh@shmtu.edu.cn)

Received 12 June 2019; Accepted 28 October 2019; Published 25 November 2019

Academic Editor: Aniello Riccio

Copyright © 2019 Zhijian Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The feedback PID method was mainly used for the navigating control of an unmanned surface vessel (USV). However, when the intelligent control era is coming now, the USV can be navigated more effectively. According to the USV character in its navigating control, this paper presents a parallel action-network ADHDP method. This method connects an adaptive controller parallel to the action network of the ADHDP. The adaptive controller adopts a RBF neural network approximation based on the Lyapunov stability analysis to ensure the system stability. The simulation results show that the parallel action-network ADHDP method has an adaptive control character and can navigate the USV more accurately and rapidly. In addition, this method can also eliminate the overshoot of the ADHDP controller when navigating the USV in various situations. Therefore, the adaptive stability design can greatly improve the navigating control and effectively overcome the ADHDP algorithm limitation. Thus, this adaptive control can be one of the intelligent ADHDP control methods. Furthermore, this method will be a foundation for the development of an intelligent USV controller.

## 1. Introduction

The development of science and technology is speeding up the informatization to intellectualization. Many systems are facing the new problems of intelligent control. The intelligent control can also be widely used in the unmanned surface vessel (USV) for its learning and adaptive ability. At present, the control research for the USV mainly focuses on the path-following, path planning, formation control, and experiment etc.

Among the above researches, the path-following has been researched at most. In 2017, Shin et al. proposed a path-following control for the USV based on an identified dynamic model [1]. In 2018, Qin et al. solved the path-tracking control of the USV with input saturation and full-state constraints [2]; Qu et al. presented an exponential path-tracking control of the USV with external disturbance, system uncertainties [3], etc. These methods generally require the precise model of the USV or external environment, which are difficult to establish.

The path planning is also one of the major research studies. In 2017, Kim et al. fulfilled the path optimization of the USV under environmental loads using the genetic algorithm [4]. In 2018, Lyu and Yin used a path-guided potential-field method to achieve the path planning in restricted waters [5]. In 2019, Wang et al. adopted an improved grey wolf optimizer to optimize the USV trajectory [6]. This is a new intelligent method, but it still has some shortcomings.

In addition, in 2017, Klinger et al. evaluated the intelligent controller performance through a USV experiment [7]. In 2018, Conte et al. developed a ROS of multi-agent structure for the USV, and tested its path planning [8]. Dai et al. combined many methods to achieve the UAV formation control [9] and so on. For more detailed development and trend analysis, please refer to these review papers [10–12].

However, the nonlinear adaptive optimal control of the USV has not been fully investigated. This is an important task in the intelligent USV control, and further research is

needed. For the adaptive optimal control of the USV, it usually needs solving the Hamilton–Jacobi–Behrman (HJB) equation. This equation is difficult to solve in many cases except for a linear quadratic system. The classical dynamic programming method still suffers to the curse of dimensionality. Fortunately, in recent years, the approximate dynamic programming (ADP) method has arisen with neural network approximation [13]. Liu et al. gives the most recent developments of the ADP theory and its advances in industrial control [14]. Yang et al. presented a guaranteed cost neural tracking control for a class of uncertain nonlinear systems using the ADP [15]. These researches make the ADP suitable for the optimal control of minimizing a performance index. Thus, the ADP can also be used to learn and adjust the navigating according to the USV conditions.

However, in practice, simply relying on the ADP cannot achieve enough accurate adjustment, especially under an external influence, while the adaptive control has the ability to adapt to different environment. If we combined the advantages of these two kinds of control, the USV navigating can be adjusted more quickly and accurately. Therefore, in this paper, these two methods are jointed for the navigating USV. Then, based on the analysis of the USV, this paper presents a parallel action-network ADP method. This method connects an adaptive controller parallel to the action network of the ADP. This parallel adaptive controller adopts a RBF neural network approximation based on the Lyapunov stability analysis. The simulation results show that the parallel action-network ADP has an adaptive control character and can navigate the USV more accurately and rapidly. In addition, this method can also eliminate the overshoot of the ADP controller when navigating the USV in various situations. Thus, this adaptive control can be one of the intelligent ADP control methods. This method will also be a foundation to the development of an intelligent USV controller.

## 2. The ADHDP Control Method

The action-dependent heuristic dynamic programming (ADHDP) is one of the main approaches in the ADP family. The critic and action networks of the ADHDP are usually established based on the BP neural network [16, 17]. As shown in Figure 1, the main function of the critic network uses the system state to approximate the cost function  $J(t)$ . The correctness of the cost function will guide the action network to the optimal  $u(t)$  approximation. If the output of the action network is received as part of the input to the critic network, it forms a new critic network. The new critic network conceals a controlled object model and approximates the new cost function  $Q(t)$ . The output  $Q(t)$  is an estimation of the cost function in the next time. Thus,  $Q(t) = J(t + 1)$ , and the ADHDP can perform control without the mathematical plant model.

**2.1. The Problem Formulation.** The action-critic scheme of the ADHDP and its symbols are both shown in Figure 2. Assuming that a nonlinear system is set up as

$$x(t + 1) = F_t[x(t), u(t), t], \quad (1)$$

where  $x(t) \in R^n$  is the system state of time  $t$ ,  $u(t) \in R^m$  is the system action of time  $t$ , and  $F_t$  is the state transition equation of the nonlinear system.

In order to let the system run in an optimal state for the dynamic programming algorithm, the cost function must be defined as [18]

$$J[x(i), i] = \sum_{k=i}^{\infty} \alpha^{k-i} \gamma[x(k), u(k), k], \quad (2)$$

where  $\alpha$  is the discount factor ( $0 < \alpha \leq 1$ ), and  $\gamma$  is the utility function of each time step. The objective is to choose the control action,  $u(k)$ ,  $k = i, i + 1, \dots$ , so that the cost function  $J(t)$  defined in equation (2) is minimized [19].

**2.2. The Critic Network.** As shown in Figure 2, the input vector to the critic network is

$$\mathbf{s} = [x_1(t), x_2(t), \dots, x_n(t), u_1(t), u_2(t), \dots, u_m(t)]. \quad (3)$$

The input  $q_i(t)$  to the hidden layer is calculated as

$$q_i(t) = \sum_{j=1}^{n+m} w_{c_{ij}}^{(1)} s_j. \quad (4)$$

$\mathbf{s}$  in the hidden layer of Figure 2 means the nonlinear Sigmoidal function. Thus, the output of the hidden layer is

$$p_i(t) = \frac{1 - \exp(-q_i(t))}{1 + \exp(-q_i(t))}. \quad (5)$$

The  $/$  in the output layer means adopting a linear function, so the output layer is

$$Q(t) = \sum_{i=1}^{N_{h1}} w_{c_i}^{(2)} p_i(t). \quad (6)$$

The ADHDP approximates the optimal solution by minimizing the new cost function; that is,  $Q(t) = J[x(t + 1), t + 1]$ . This is achieved by training the new critic network to minimize the following error measured over time:

$$\|E_c(t)\| = \sum_t \frac{1}{2} e_c^2(t) = \sum_t \frac{1}{2} (\alpha Q(t) - [Q(t - 1) - \gamma(t)])^2, \quad (7)$$

where  $Q(t) = Q[x(t), u(t), t, w_c]$ ,  $w_c$  is the critic network weights. The output  $Q(t)$  of time  $t$  is an estimation of the dynamic programming at time  $t + 1$ , which is the cost function in the next time.

Then, the gradient descent algorithm is used to update the critic network weights and minimize  $E_c(t)$  at each time step by

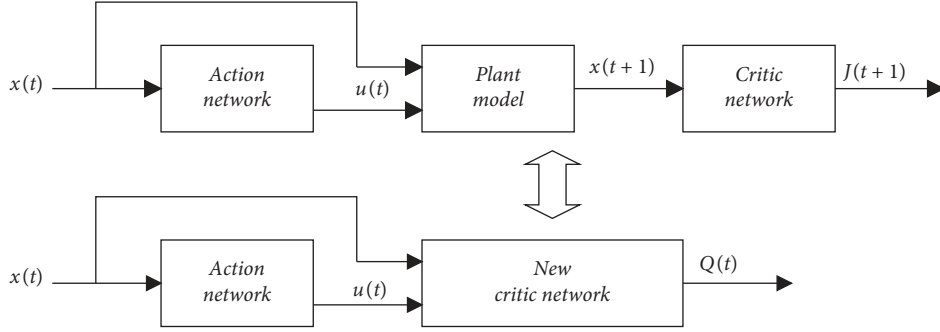


FIGURE 1: A new critic network established in the ADHDP.

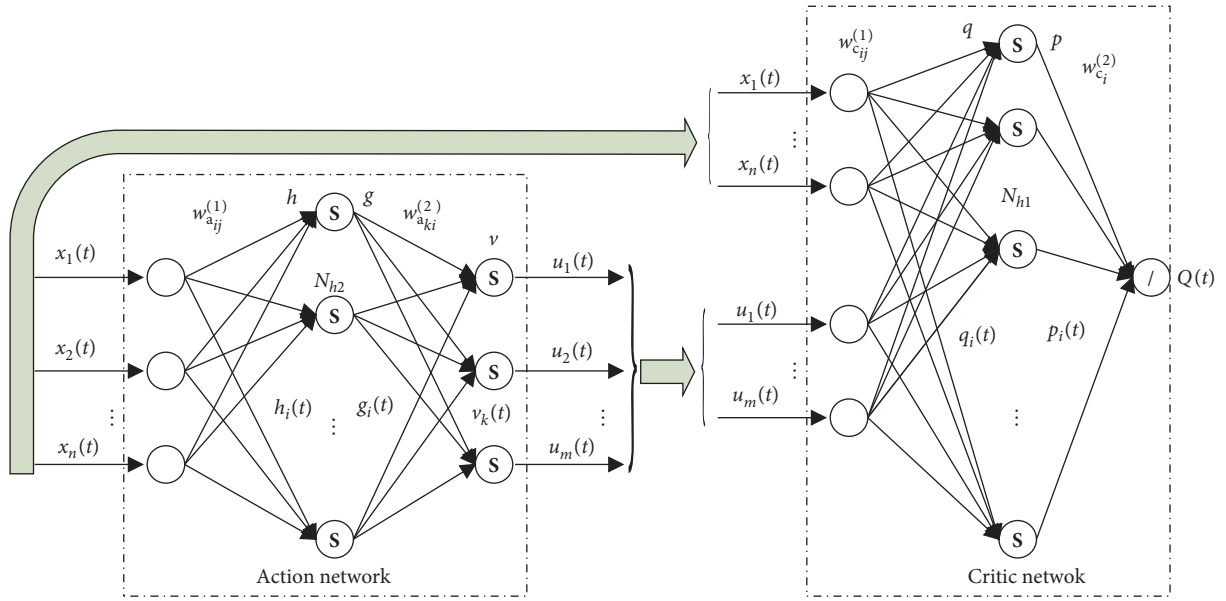


FIGURE 2: The action-critic principle and structure of the ADHDP.

$$w_c(t+1) = w_c(t) + \Delta w_c(t),$$

$$\begin{aligned} \Delta w_c(t) &= l_c(t) \left[ \frac{\partial E_c(t)}{\partial w_c(t)} \right] = l_c(t) \left[ \frac{\partial E_c(t)}{\partial e_c(t)} \cdot \frac{\partial e_c(t)}{\partial Q(t)} \cdot \frac{\partial Q(t)}{\partial w_c(t)} \right] \\ &= -l_c(t) \cdot e_c(t) \cdot \alpha \cdot \frac{\partial Q(t)}{\partial w_c(t)}, \end{aligned} \quad (8)$$

where  $l_c(t) > 0$  is the learning rate of the critic network at time  $t$ , which decreases to a very small value with time  $t$  [20].

**2.3. The Action Network.** As shown in Figure 2, the input to the action network is

$$\mathbf{x} = [x_1(t), x_2(t), \dots, x_n(t)]. \quad (9)$$

The input  $h_i(t)$  to the hidden layer of the action network is calculated as

$$h_i(t) = \sum_{j=1}^n w_{a_{ij}}^{(1)} x_j. \quad (10)$$

The hidden layer uses the nonlinear Sigmoidal function; thus, the output of the hidden layer is

$$g_i(t) = \frac{1 - \exp(-h_i(t))}{1 + \exp(-h_i(t))}. \quad (11)$$

The input  $v_k(t)$  to the output layer is calculated as

$$v_k(t) = \sum_{i=1}^{N_{h2}} w_{a_{ki}}^{(2)} g_i(t). \quad (12)$$

The output layer also uses the nonlinear Sigmoidal function; thus, the output of the action network is

$$u_k(t) = \frac{1 - \exp(-v_k(t))}{1 + \exp(-v_k(t))}. \quad (13)$$

When the critic network has been trained, the objective of training the action network is to minimize the following index:

$$E_a(t) = \frac{1}{2} e_a^2(t) = \frac{1}{2} Q^2(t). \quad (14)$$

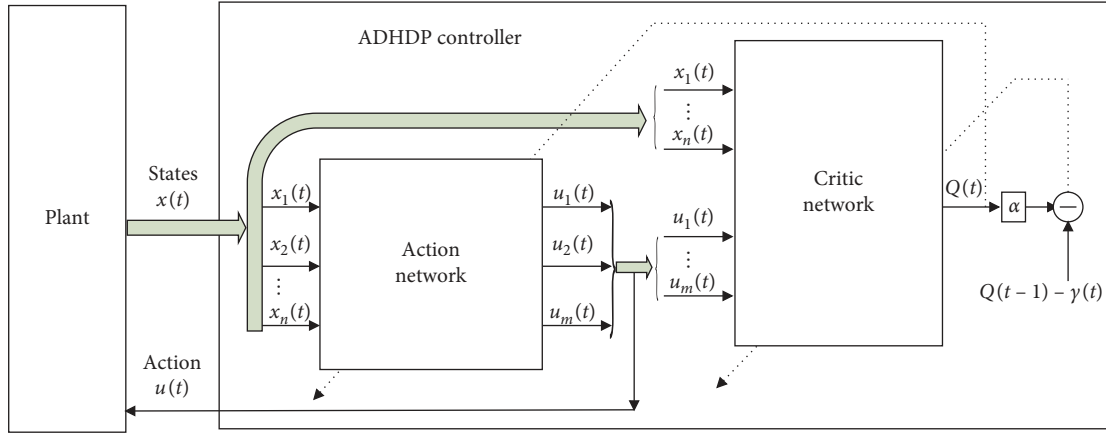


FIGURE 3: The schematic diagram of the ADHDP control method.

Then, the gradient descent algorithm is also used to update the weight of the action network by minimizing  $E_a(t)$  with

$$w_a(t+1) = w_a(t) + \Delta w_a(t),$$

$$\begin{aligned} \Delta w_a(t) &= l_a(t) \left[ \frac{\partial E_a(t)}{\partial w_a(t)} \right] = l_a(t) \left[ \frac{\partial E_a(t)}{\partial e_a(t)} \cdot \frac{\partial e_a(t)}{\partial Q(t)} \cdot \frac{\partial Q(t)}{\partial w_a(t)} \right] \\ &= -l_a(t) \cdot e_a(t) \cdot \frac{\partial Q(t)}{\partial w_a(t)}, \end{aligned} \quad (15)$$

where  $l_a(t) > 0$  is the learning rate of the action network at time  $t$ , which also decreases to a very small value with time  $t$  [20].

After getting the optimal  $Q(t)$  function, the optimal control  $u^*(t)$  can be obtained by the following equation:

$$u^*(t) = \arg \min_{u(t)} Q^*(x(t), u(t)), \quad (16)$$

that is, to obtain the optimal control output of the action network, it is necessary to make the cost function  $Q(t) = 0$ . This will also make the critic network output equal to zero as close as possible.

As shown in Figure 3, the dashed line means that the error should be eventually reduced to zero by calculating the error of each operation and the increment of the critic and action network weights.

### 3. The Parallel Action-Network ADHDP: The Adaptive ADHDP Method

The addition of an adaptive control to the navigating of the USV is to ensure its system stability. This is because the gradient descent algorithm cannot guarantee the system stability [21] when updating the ADHDP parameters. In addition, it will lead to a local optimum in some cases. Therefore, the stability design based on the Lyapunov theory will greatly increase the stability of the navigating control and effectively overcome the limitation of the ADHDP algorithm.

Based on the error minimization in equation (18), the error and its differential are used as the inputs to both the

action network and adaptive controller, as shown in Figure 4. After getting the error and its differential inputs, the adaptive control algorithm is calculated separately.

When different control signals  $u(t)$  are obtained, the litter between the adaptive control and action network outputs is selected as input to the controlled object. This will also help the stability of the navigating USV system. In this way, the advantage of the AHDHP and adaptive control can be integrated to achieve the navigating stability of the USV.

The adaptive control process is as follows: Supposing that the nonlinear system is

$$\ddot{\mathbf{x}} = f(\mathbf{x}, \dot{\mathbf{x}}) + g(\mathbf{x}, \dot{\mathbf{x}})\mathbf{u}, \quad (17)$$

$$e = \text{rin} - y, \quad (18)$$

where rin is the tracking navigation command, the error  $e$  is the difference between the command and the system output  $y$ ,  $x$  is the system state,  $u$  is the control input, and  $f(\mathbf{x}, \dot{\mathbf{x}})$  and  $g(\mathbf{x}, \dot{\mathbf{x}})$  are the nonlinear state functions of the USV system.

Supposing that  $\mathbf{E} = [e \ \dot{e}]^T$ , and the optimal control  $\mathbf{u}^*$  is designed as

$$\mathbf{u}^* = \frac{1}{g(\mathbf{x}, \dot{\mathbf{x}})} \left[ -f(\mathbf{x}, \dot{\mathbf{x}}) + \ddot{\text{rin}} + \mathbf{K}^T \mathbf{E} \right], \quad (19)$$

where  $\mathbf{K}$  is a coefficient. By substituting equations (18) and (19) in equation (17), and simplifying its form, the system error satisfies

$$\ddot{e} + k_2 \dot{e} + k_1 e = 0. \quad (20)$$

To design  $\mathbf{K} = [k_1 \ k_2]^T$ , the root of its polynomial function  $s^2 + k_2 s + k_1 = 0$  must be in the left half plane of the phase-plane. When  $t$  tends to infinity, there should be  $e(t) \rightarrow 0$  and  $\dot{e}(t) \rightarrow 0$ .

For the adaptive control, the RBF neural network is used for approximation. The input to the network is  $\mathbf{x} = [e \ \dot{e}]^T$ , and the output of the network is

$$\hat{f}(x) = \hat{\mathbf{W}}^T \mathbf{h}(x), \quad (21)$$

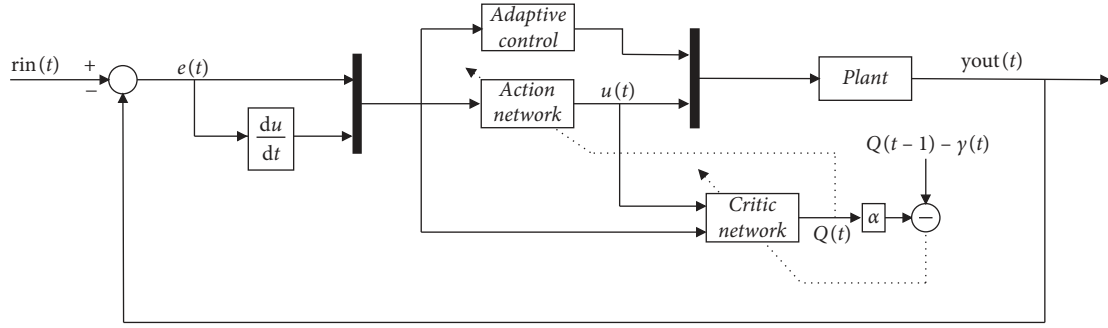


FIGURE 4: The scheme of the parallel action-network ADHDP method for the navigating control of the USV.

TABLE 1: The related USV parameters for simulation.

| $L$     | $B$      | $V$       | $T$      | $C_b$ | $m$                      | $A_\delta$           |
|---------|----------|-----------|----------|-------|--------------------------|----------------------|
| 180 (m) | 27.8 (m) | 9.0 (m/s) | 10.3 (m) | 0.64  | 30,000 (m <sup>3</sup> ) | 38 (m <sup>2</sup> ) |

where  $\hat{W}$  is the estimation of the network weight,  $h_j = g(\|\mathbf{x} - \mathbf{c}_{ij}\|^2/b_j^2)$  is a Gauss basis function of No.  $j$  weight, and  $\mathbf{h}(\mathbf{x})$  is a vector of the Gauss basis function, please see [22]. Suppose that the matrix  $\mathbf{P}$  is symmetric positive-definite and satisfies the following Lyapunov equation:

$$\Lambda^T \mathbf{P} + \mathbf{P} \Lambda = -\mathbf{I}, \quad (22)$$

where  $\Lambda = \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix}$  and  $\mathbf{I}$  is a unit matrix. After obtaining the matrix  $\mathbf{P}$ , the expected RBF network weights are

$$\hat{W} = -\varphi \mathbf{E}^T \mathbf{P} \mathbf{B} \mathbf{h}(\mathbf{x}), \quad (23)$$

where  $\mathbf{B} = [0 \ 1]^T$ .

After the weights of the RBF neural network are obtained, the output of this network can be obtained by equation (21). Finally, the optimal control signal can be obtained by equation (19).

#### 4. The USV Model for Navigating Control

The ADHDP is based on the state system and does not require a controlled object model. Thus, the ADHDP is also known as a data-driven control method [23], which enables an online learning and control [24]. Thus, in this study, the USV model is only used as a simulation object, not for the controller design. However, the choice of a suitable controlled object model is also very important for numerical experiment.

The motion model and state space of the USV used in this paper are summarized in [25]. This model was proposed by Abkowitz and other scholars with mathematical integrity and rigor. The standard state space of the USV is expressed as follows

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\delta, \quad (24)$$

where  $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 1 & 0 \end{bmatrix}$ ,  $\mathbf{B} = \begin{bmatrix} b_{11} \\ b_{21} \\ 0 \end{bmatrix}$ , and  $\delta$  is the rudder angle of the USV.

In it,

$$a_{11} = \frac{[(I'_{zz} - N'_r)Y'_v - (m'x'_C - Y'_r)N'_v]V}{S_1},$$

$$a_{12} = \frac{[(I'_{zz} - N'_r)(Y'_r - m') - (m'x'_C - Y'_r)(N'_r - m'x'_C)]LV}{S_1},$$

$$a_{21} = \frac{[-(m'x'_C - N'_v)Y'_v + (m' - Y'_v)N'_v]V/L}{S_1},$$

$$a_{22} = \frac{[-(m'x'_C - N'_v)(Y'_r - m') + (m' - Y'_v)(N'_r - m'x'_C)]V}{S_1},$$

$$b_{11} = \frac{[(I'_{zz} - N'_r)Y'_\delta - (m'x'_C - Y'_r)N'_\delta]V^2}{S_1},$$

$$b_{21} = \frac{[-(m'x'_C - N'_v)Y'_\delta + (m' - Y'_v)N'_\delta]V^2/L}{S_1},$$

$$S_1 = [(I'_{zz} - N'_r)(m' - Y'_v) - (m'x'_C - N'_v)(m'x'_C - Y'_r)]L, \quad (25)$$

$m$  is the mass of the USV,  $m'$  is its dimensionless mass during calculation, which is calculated with  $m' = (1/2)m\rho L^3$ ,  $\rho$  is the sea water density,  $L$  is the length between the perpendiculars;  $I'_{zz}$  is the inertial distance, which is calculated with  $I'_{zz} = I_{zz}((1/2)\rho L^2)$ , and  $I_{zz} = (mL^2/16)$ ;  $V$  is the USV speed;  $S_1$  is the reference area. For other related intermediate variables, please see [25], and they can be calculated with

$$\begin{aligned}
Y'_v &= -\left[1 + \frac{0.16C_b B}{T} - 5.1\left(\frac{B}{L}\right)^2\right]\pi\left(\frac{T}{L}\right)^2, \\
Y'_r &= -\left[\frac{0.67B}{L} - 0.0033\left(\frac{B}{T}\right)^2\right]\pi\left(\frac{T}{L}\right)^2, \\
N'_v &= -\left[\frac{1.1B}{L} - \frac{0.041B}{T}\right]\pi\left(\frac{T}{L}\right)^2, \\
N'_r &= -\left[\frac{1}{12} + \frac{0.017C_b B}{T} - \frac{0.33B}{L}\right]\pi\left(\frac{T}{L}\right)^2, \\
Y'_v &= -\left[1 + \frac{0.40C_b B}{T}\right]\pi\left(\frac{T}{L}\right)^2, \\
Y'_r &= -\left[-\frac{1}{2} + \frac{2.2B}{L} - \frac{0.080B}{T}\right]\pi\left(\frac{T}{L}\right)^2, \\
N'_v &= -\left[\frac{1}{2} + \frac{2.4T}{L}\right]\pi\left(\frac{T}{L}\right)^2, \\
N'_r &= -\left[\frac{1}{4} + \frac{0.039B}{T} - \frac{0.56B}{L}\right]\pi\left(\frac{T}{L}\right)^2, \\
Y'_\delta &= \frac{3.0A_\delta}{L^2}, \\
N'_\delta &= -\left(\frac{1}{2}\right)Y'_\delta,
\end{aligned} \tag{26}$$

where  $B$  is the width of the USV,  $C_b$  is the block coefficient,  $A_\delta$  is the area of the rudder leaf, and  $T$  is the draught of the USV.

Considering the marine environment effect to the movement of the USV, the interference of a white noise  $\omega = [\omega_1 \ \omega_2 \ \omega_3]$  is added. Thus,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\delta + \omega. \tag{27}$$

## 5. The Numerical Experiment

**5.1. The Parameters of the Control System.** The utility function of the adaptive ADHDP controller can be designed as

$$\gamma(t) = (4 \cdot e(t) + 4 \cdot \dot{e}(t))^2, \tag{28}$$

where  $e(t)$  is the error in equation (18) at time  $t$ , and  $\dot{e}(t)$  is its differential with respect to time  $t$ .

The MATLAB/Simulink is used for the simulation. The S-function is used to establish the related model and algorithm because there is no ADHDP toolbox in MATLAB. The parameters of the simulating USV model are shown in Table 1.

### 5.2. Simulation and Results

- (1) The navigating target is set as east ( $90^\circ$ ) and the initial navigation is north ( $0^\circ$ ). The simulation time is 50 seconds. The navigating responses of a PID and the adaptive ADHDP control are shown in Figure 5. The

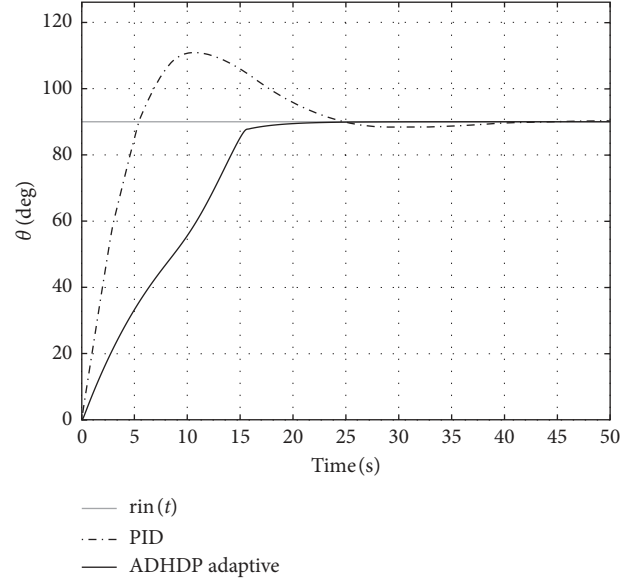


FIGURE 5: A  $90^\circ$  navigating control response of the USV under a PID and the adaptive ADHDP controller.

TABLE 2: The comparison of a  $90^\circ$  navigating control response of the USV under a PID and the adaptive ADHDP controller.

| Comparing items               | PID      | Adaptive ADHDP |
|-------------------------------|----------|----------------|
| Maximum response ( $^\circ$ ) | 111.1079 | 90             |
| Overshoot (%)                 | 23.29    | 0              |
| Delay time (s)                | 2.1400   | 7.5535         |
| Rise time (s)                 | 5.4600   | 23.9995        |
| Peak time (s)                 | 10.8650  | 23.9995        |
| Adjustment time (s)           | 22.5850  | 16.3405        |

detailed comparison information of result curves in Figure 5 is shown in Table 2.

From the data in Figure 5 and Table 2, it can be seen that the combination of the ADHDP and adaptive control can shorten the adjusting time and eliminate overshoot. The data in figures and tables also prove that the combination of the ADHDP and adaptive control can adjust the navigating USV quickly and accurately, and it only needs one navigating change.

- (2) When navigating the USV in a narrow river, the adjusting frequency and time are significantly increased. For the simulation in this condition, the navigating target is set as east ( $90^\circ$ ) for the first 50 seconds, and the initial heading is north ( $0^\circ$ ). Then, the navigating target is set as north ( $0^\circ$ ) in the 50th seconds, and the navigating should turn around  $90^\circ$ . This stage runs for another 50 seconds. Then, the navigating target is set as east ( $90^\circ$ ) again in the 100th second, and the navigating should turn back  $90^\circ$ . This stage also runs for 50 seconds. The simulation results are shown in Figure 6.

According to the simulation results, it can be seen that the PID controller has a long time to adjust the navigating USV, and its overshoot is very big. However, the navigating USV control with the

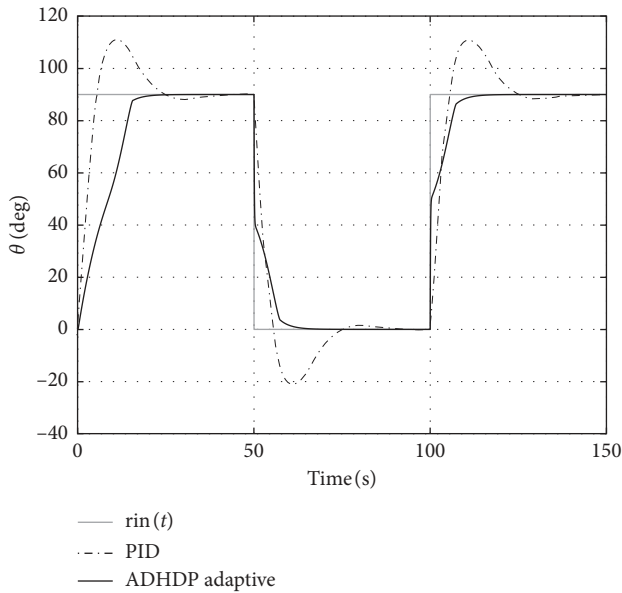


FIGURE 6: The PID and adaptive ADHDP controller response to a continue sharp navigating process of the USV.

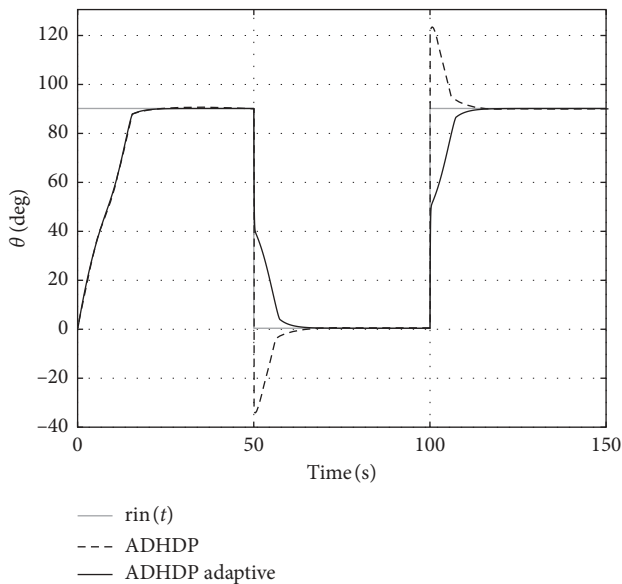


FIGURE 7: The ADHDP and adaptive ADHDP controller response to a continuous and sharp navigating process of the USV.

adaptive ADHDP is significantly enhanced in overshoot and rapidity compared with the PID controller.

- (3) Figure 7 shows the control effect and response under the above continuous and sharp navigating with the individual ADHDP, and compares to that of the adaptive ADHDP. It can be seen from Figure 7 that the ADHDP controller has a larger overshoot in a continuous and sharp navigating change. This is because the ADHDP is based on the data-driven and previous learning effects, which is difficult to make a right response in a short period of time. After

combining the adaptive control, the adaptive ADHDP can avoid an overshoot in mutation state and can ensure a safe driving for the USV.

## 6. Discussion and Conclusion

The ADHDP control is a data-driven based method, which can be used without a plant model. In the simulation, it can be seen that the adjusting time of the ADHDP controller is much smaller than that of the PID controller, but the overshoot is larger than that of the PID. Therefore, an adaptive control is added to eliminate the overshoot of the ADHDP controller under the condition of a continuous and sharp navigating change.

In addition, the simulation proves the feasibility of the adaptive ADHDP control method. This control method also does not need the model of the controlled object, which can simplify the controller design although it needs combining the adaptive method. This method can also increase the speed and stability of the controller. Thus, these methods can lay a foundation for the development of the intelligent USV and have a practical effect for a further enhancing of the intelligent USV control.

## Data Availability

The data used to support the findings of this study have been deposited in the Adaptive Navigating Control Based on the Parallel Action-Network ADHDP for Unmanned Surface Vessel repository (7697143).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

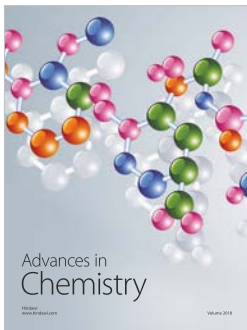
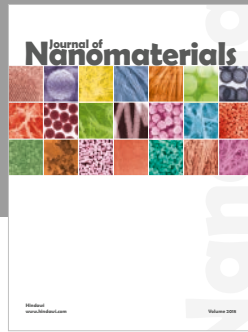
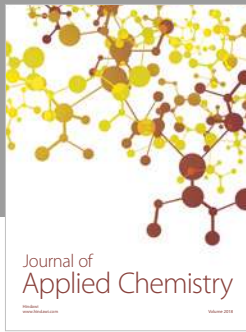
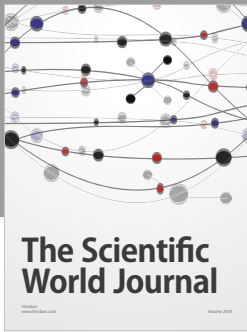
This work was supported by the NSFC Projects of China under grant nos. 61403250, 51779136, and 51509151.

## References

- [1] J. Shin, D. J. Kwak, and Y.-i. Lee, "Adaptive path-following control for an unmanned surface vessel using an identified dynamic model," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 3, pp. 1143–1153, 2017.
- [2] H. Qin, C. Li, Y. Sun, Z. Deng, and Y. Liu, "Trajectory tracking control of unmanned surface vessels with input saturation and full-state constraints," *International Journal of Advanced Robotic Systems*, vol. 15, no. 5, pp. 1–9, 2018.
- [3] Y. Qu, B. Xiao, Z. Fu, and D. Yuan, "Trajectory exponential tracking control of unmanned surface ships with external disturbance and system uncertainties," *ISA Transactions*, vol. 78, pp. 47–55, 2018.
- [4] H. Kim, S.-H. Kim, M. Jeon, J. Kim, S. Song, and K.-J. Paik, "A study on path optimization method of an unmanned surface vehicle under environmental loads using genetic algorithm," *Ocean Engineering*, vol. 142, pp. 616–624, 2017.
- [5] H. Lyu and Y. Yin, "Fast path planning for autonomous ships in restricted waters," *Applied Sciences*, vol. 8, no. 12, p. 2592, 2018.

- [6] Y. Wang, P. Yao, and Y. Dou, "Monitoring trajectory optimization for unmanned surface vessel in sailboat race," *Optik*, vol. 176, pp. 394–400, 2019.
- [7] W. B. Klinger, I. R. Bertaska, K. D. von Ellenrieder, and M. R. Dhanak, "Control of an unmanned surface vehicle with uncertain displacement and drag," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 2, pp. 458–476, 2017.
- [8] G. Conte, D. Scaradozzi, D. Mannocchi, P. Raspa, L. Panebianco, and L. Screpanti, "Development and experimental tests of a ros multi-agent structure for autonomous surface vehicles," *Journal of Intelligent & Robotic Systems*, vol. 92, no. 3-4, pp. 705–718, 2018.
- [9] S.-L. Dai, S. He, H. Lin, and C. Wang, "Platoon formation control with prescribed performance guarantees for usvs," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4237–4246, 2018.
- [10] G. Eudeline, "Developing and evolving unmanned surface vessels," *Sea Technology*, vol. 58, no. 12, pp. 10–12, 2017.
- [11] A. V. Titov and L. Barakat, "Prospects for technological development and implementation of unmanned ships," *Marine Intellectual Technologies*, vol. 1, no. 3, pp. 94–103, 2018.
- [12] Y. Liu and R. Bucknall, "A survey of formation control and motion planning of multiple unmanned vehicles," *Robotica*, vol. 36, no. 7, pp. 1019–1047, 2018.
- [13] F. He, J. Yang, and M. Li, "Vehicle scheduling under stochastic trip times: an approximate dynamic programming approach," *Transportation Research Part C: Emerging Technologies*, vol. 96, pp. 144–159, 2018.
- [14] D. Liu, Q. Wei, D. Wang, X. Yang, and H. Li, *Adaptive Dynamic Programming with Applications in Optimal Control*, Springer, London, UK, 2017.
- [15] X. Yang, D. Liu, Q. Wei, and D. Wang, "Guaranteed cost neural tracking control for a class of uncertain nonlinear systems using adaptive dynamic programming," *Neurocomputing*, vol. 198, pp. 80–90, 2016.
- [16] L. Hager, K. R. Uren, G. van Schoor, and A. J. van Rensburg, "Adaptive neural network control of a helicopter system with optimal observer and actor-critic design," *Neurocomputing*, vol. 302, pp. 75–90, 2018.
- [17] D. Wang, D. Liu, Y. Zhang, and H. Li, "Neural network robust tracking control with adaptive critic framework for uncertain nonlinear systems," *Neural Networks*, vol. 97, pp. 11–18, 2018.
- [18] Q. Wei, D. Liu, and H. Lin, "Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 840–853, 2016.
- [19] F. Y. Wang, N. Jin, D. Liu, and Q. Wei, "Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with  $\varepsilon$ -error bound," *IEEE Transactions on Neural Networks*, vol. 22, no. 1, pp. 24–36, 2011.
- [20] A. Heydari, "Revisiting approximate dynamic programming and its convergence," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2733–2743, 2014.
- [21] Z. Q. Wu, W. J. Jia, L. R. Zhao, and C. H. Wukey, "Maximum wind power tracking for PMSG chaos systems-ADHDP method," *Applied Soft Computing*, vol. 36, pp. 204–209, 2015.
- [22] X. Shi, Y. Cheng, C. Yin, X. Huang, and S.-m. Zhong, "Design of adaptive backstepping dynamic surface control method with RBF neural network for uncertain nonlinear system," *Neurocomputing*, vol. 330, pp. 490–503, 2019.
- [23] Z. Peng, Y. Zhao, J. Hu, and B. K. Ghosh, "Data-driven optimal tracking control of discrete-time multi-agent systems with two-stage policy iteration algorithm," *Information Sciences*, vol. 481, pp. 189–202, 2019.
- [24] R. Panigrahi, P. R. Srivastava, and D. Sharma, "Online learning: adoption, continuance, and learning outcome—a review of literature," *International Journal of Information Management*, vol. 43, pp. 1–14, 2018.
- [25] P. Borkowski, "Ship course stabilization by feedback linearization with adaptive object model," *Polish Maritime Research*, vol. 21, no. 1, pp. 14–19, 2013.





**Hindawi**  
Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

