Letter

Straight-Path Following and Formation Control of USVs Using Distributed Deep Reinforcement Learning and Adaptive Neural Network

Zhengqing Han, Yintao Wang, and Qi Sun

Dear Editor,

This letter presents a distributed deep reinforcement learning (DRL) based approach to deal with the path following and formation control problems for underactuated unmanned surface vehicles (USVs). By constructing two independent actor-critic architectures, the deep deterministic policy gradient (DDPG) method is proposed to determine the desired heading and speed command for each USV. We consider the realistic dynamical model and the input saturation problem. The radial basis function neural networks (RBF NNs) are employed to approximate the hydrodynamics and unknown external disturbances of USVs. Simulation results show that our proposed method can achieve high-level tracking control accuracy while keeping a desired stable formation.

Employing multiple USVs as a formation fleet is essential for future USV operations [1]. To achieve formation, the vehicles can be driven to the individual paths first, then the formation is obtained by synchronizing the motion of the vehicles along the paths. Therefore, the path following and formation control problems are simultaneously studied in this letter. For the path following (PF) task, a USV is assumed to follow a path without temporal constraints [2]. Since the route is typically specified by waypoints, following the straight path between waypoints is a fundamental task to USVs. It is a challenging issue considering the highly nonlinear systems with time-varying hydrodynamic coefficients, external disturbances and underactuated characteristic. Extensive research has been undertaken to address above problems, such as backstepping control [3], sliding mode control [4], NN-based control [5] and model predictive control [6]. Among these methods, RBF NNs [5] have been proven to be a powerful solution for handling model uncertainties and disturbances.

Recently, researchers have shown an increased interest in artificial intelligence (AI) methods, e.g., DDPG method is used to solve the PF problem. DDPG plays a role of both guidance law and low-level controller in [7], or only acts as a low-level controller in [8]. However, these approaches are vulnerable to dynamic environment and cannot realize satisfactory tracking control accuracy. To overcome above problems, a DDPG-based guidance law is presented with an adaptive sliding mode controller in [9], which proves the benefit of DRL strategies in guidance. Therefore, a promising idea is taking advantage of DRL methods to obtain efficient guidance law, then using RBF NNs to achieve high-level control accuracy, while giving the DRL-based approach the ability to deal with model uncertainties and disturbances. That is the first motivation of our work.

Several attempts have been made to the formation control of USVs. In [10], all vehicles are coordinated by consensus tracking control law, but a fixed communication topology is required. To reduce the information exchanges among vehicles, an event-based approach is presented in [11] such that the periodic transmission is avoided. In [12], each robot follows its leader by visual servoing, where some information only needs to be exchanged by an acoustic sensor at the beginning. The study in [13] provides new insights into USV formations by proposing a distributed DRL algorithm, where the adaptive formation is achieved by observing the relative angle and distance between follower and leader, and the plug-and-play capability is obtained by applying a trained agent onto any newly added USVs. However, it does not consider the problems of model uncertainties, unknown disturbances and input saturation. Hence, the second motivation of our work is using a distributed DRL method to achieve adaptive and extendable formation control, while reducing the communication frequency. Similarly, by combining DRL method with RBF NNs, the high-level control accuracy can be achieved.

Based upon the discussions above, our main contributions are: 1) A DDPG-based guidance law is employed to make USVs converge to the desired path, where transfer learning (TL) is utilized to increase the tracking performance; 2) The desired formation position of each vehicle is achieved by using a distributed DRL method, and a potential function is presented to realize smooth control; 3) RBF NNs are proposed to design the low-level controller for underactuated USVs that subject to the unknown external disturbances, and an adaptive compensating approach is presented to address the input saturation problem. In this letter, the DRL-based approach is utilized to determine the desired yaw rate command r_d and speed command u_d, and the NN-based controller is expected to produce the control input forces and torques for tracking the reference signals. The algorithm architecture is depicted in Fig. 1 . It is shown through numerical examples that our proposed method can achieve satisfactory performance for both path following and formation tasks.

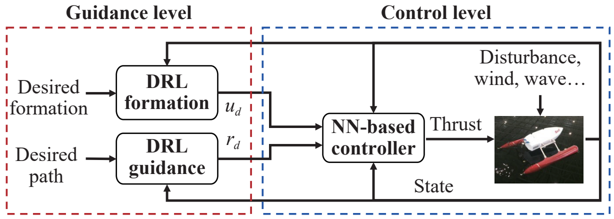


Fig. 1. The architecture of the proposed algorithm.

Preliminaries: 1) The dynamic model concerned in this work is motivated by [5]. The vector $\eta=[x,y,\psi]^T$ denotes the position (x,y) and the yaw angle ψ in the earth-fixed frame. The vector $\nu=[u,\upsilon,r]^T$ describes the linear velocities (u,υ) and the yaw rate r in the body-fixed frame. We consider a group of USVs with the same structure. Then, the three-degree-of-freedom model of each vehicle can be described as

$$\begin{aligned}&\dot{\eta}=J(\psi)\nu\\&M\dot{\nu}+C(\nu)\nu+D(\nu)\nu=\delta(\tau)+d\end{aligned}\tag{1}$$

where $J(\psi)=[\cos\psi,-\sin\psi,0;\sin\psi,\cos\psi,0;0,0,1]$ denotes a rotation matrix; $M\in\mathbb{R}^{3\times3}$ is the inertia matrix, $C\in\mathbb{R}^{3\times3}$ and $D\in\mathbb{R}^{3\times3}$ are the Coriolis and Centripetal matrix, and damping matrix, respectively; $d=[d_u,d_v,d_r]^T$ represents the unknown disturbance; $\tau=[\tau_u,0,\tau_r]^T$, where τ_u,τ_r are the surge force and the yaw moment, respectively; $\delta(\tau)=[\delta_1(\tau_u),0,\delta_3(\tau_r)]^T$ isthe saturated control vector adopted from [5].

2) For any nonlinear continuous function f, the estimation by RBF NNs can be defined as $f(Z)=\hat{f}(Z,W^*)+\varepsilon(Z)$, where $Z\in\Omega$ denotes the input vector; $\hat{f}(Z,W^*)=W^{*T}\boldsymbol{S}(Z)$ denotes the estimation of the continuous function $f(Z)$; $\varepsilon(Z)$ represents the bounded approximation error satisfying $|\varepsilon(Z)|\leq\varepsilon^*$; $W^*=\text{argmin}[\sup|f(Z)-\hat{f}(Z,\hat{W})|]$ is the optimal NNs weights, where the weight vector $\hat{W}$ is defined as $\hat{W}=[\hat{W}_1,\ldots,\hat{W}_m]^T$, and m is the number of NNs nodes. $\boldsymbol{S}(Z)=[S_1(Z),\ldots,S_m(Z)]^T$ is the nonlinear regressor vector of the inputs adopted from [5].

Controller design: The system matrices M, C and D are divided into nominal part and bias part, i.e., $M=M^*+\Delta M$, $C=C^*+\Delta C$ and $D=D^*+\Delta D$, where $(\cdot)^*$ denotes the nominal value obtained from

Corresponding author: Yintao Wang.

Citation: Z. Q. Han, Y. T. Wang, and Q. Sun, "Straight-path following and formation control of USVs using distributed deep reinforcement learning and adaptive neural network," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 2, pp. 572–574, Feb. 2023.

The authors are with the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: hanzq@mail.nwpu.edu.cn; wangyintao@nwpu.edu.cn; sunqi@nwpu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.123255

the experiment, and $\Delta(\cdot)$ denotes the unknown bias part. Thus, the model in (1) can be rewritten as $M^*\dot{v}+C^*(v)v+D^*(v)v=\delta(\tau)+d_{\text{sum}}$, where $d_{\text{sum}}=[d_{\text{sum},u},d_{\text{sum},v},d_{\text{sum},r}]^T=-\Delta M\dot{v}-\Delta C(v)v-\Delta D(v)v+d$. Then, the model can be further rewritten as

$$\begin{aligned}\dot{u}&=\phi_u+\phi_{d_u}+\delta_1(\tau_u)/m^*_{11}\\ \dot{v}&=\phi_v+\phi_{d_v}\\ \dot{r}&=\phi_r+\phi_{d_r}+m^*_{22}\delta_3(\tau_r)/m_r\end{aligned}\tag{2}$$

where $\phi_u=\frac{m^*_{22}}{m^*_{11}}vr+\frac{m^*_{23}}{m^*_{11}}r^2-\frac{d^*_{11}}{m^*_{11}}u$, $\phi_v=-\frac{m^*_{23}}{m^*_{22}}\dot{r}-\frac{m^*_{11}}{m^*_{22}}ur-\frac{d^*_{22}}{m^*_{22}}v-\frac{d^*_{23}}{m^*_{22}}r$, $\phi_r=\frac{1}{m_r}\{-(m^{*2}_{22}-m^*_{11}m^*_{22})uv-(m^*_{22}m^*_{23}-m^*_{23}m^*_{11})ur-(m^*_{22}d^*_{32}-m^*_{23}d^*_{22})v-(m^*_{22}d^*_{33}-m^*_{23}d^*_{23})r\}$, $\phi_{d_u}=d_{\text{sum},u}/m^*_{11}$, $\phi_{d_v}=d_{\text{sum},v}/m^*_{22}$, $\phi_{d_r}=\frac{1}{m_r}(m^*_{22}d_{\text{sum},r}-m^*_{23}d_{\text{sum},v})$, $m_r=m^*_{22}m^*_{33}-m^{*2}_{23}$; m^*_{ij} and d^*_{ij} denote the ith row and jth column of M^* and D^*, respectively. Then, we define that $W^{*T}_1\boldsymbol{S}_1(Z)+\varepsilon_1=-\phi_u-\phi_{d_u}$, and $W^{*T}_3\boldsymbol{S}_3(Z)+\varepsilon_3=-\phi_r-\phi_{d_r}$. The surge speed tracking error is defined as $u_e=u_d-u-\alpha_1\tanh\beta_1$, and the yaw rate tracking error is defined as $r_e=r_d-r-\alpha_3\tanh\beta_3$, where α_1 and α_3 are positive constants. Define that $[\Delta_{\tau_u},0,\Delta_{\tau_r}]^T=\tau-\delta(\tau)$, thus β_1 is designed as $\dot{\beta}_1=\cosh^2\beta_1\{-\mu_u\beta_1+\Delta_{\tau_u}/m^*_{11}\}/\alpha_1$, and β_3 is designed as $\dot{\beta}_3=\cosh^2\beta_3\{-\mu_r\beta_3+m^*_{22}\Delta_{\tau_r}/m_r\}/\alpha_3$, where μ_u and μ_r are positive constants. Then, by differentiating u_e and r_e, the surge force and the yaw moment can be addressed by

$$\begin{aligned}\tau_u&=m^*_{11}(K_1u_e+\hat{W}^T_1\boldsymbol{S}_1(Z)+\dot{u}_d+\mu_u\beta_1)\\ \tau_r&=m_r(K_3r_e+\hat{W}^T_3\boldsymbol{S}_3(Z)+\dot{r}_d+\mu_r\beta_3)/m^*_{22}\end{aligned}\tag{3}$$

where $K_1,K_3>0\in\mathbb{R}$. The weight update law is designed as

$$\begin{aligned}\dot{\hat{W}}_1&=\Gamma_1(\boldsymbol{S}_1(Z)u_e-\kappa_1\hat{W}_1)\\ \dot{\hat{W}}_3&=\Gamma_3(\boldsymbol{S}_3(Z)r_e-\kappa_3\hat{W}_3)\end{aligned}\tag{4}$$

where $\kappa_1,\kappa_3>0\in\mathbb{R}$, $\Gamma_1,\Gamma_3>0\in\mathbb{R}^{m\times m}$.

Theorem 1: Consider the underactuated USV model (2) in the presence of model uncertainties, unknown environmental disturbances and input saturation, together with the controller in (3), and the weight update law in (4), if the appropriate design parameters are chosen, the surge speed tracking error and the yaw rate tracking error converge to a small neighborhood of the origin, the signals in the closed loop system are uniformly ultimately bounded.

Proof: The proof is omitted due to page limitation. ■

Path following and formation tasks: As shown in Fig. 2, a start-point $\boldsymbol{p}_k=[x_k,y_k]^T$ and an endpoint $\boldsymbol{p}_{k+1}=[x_{k+1},y_{k+1}]^T$ are chosen to construct a straight path in the earth frame. $\gamma_p=\text{atan2}(y_{k+1}-y_k,x_{k+1}-x_k)$ is the angle of the path. $x_e=(x-x_{k+1})\cos\gamma_p+(y-y_{k+1})\sin\gamma_p$ is the along-tracking error, and $y_e=-(x-x_{k+1})\sin\gamma_p+(y-y_{k+1})\cos\gamma_p$ is the cross-tracking error. Then, for the PF task, the control objective is to make y_e converge to zero, i.e., $\lim_{t\to\infty}y_e(t)=0$. For the formation task, the leader-follower strategy is selected, and the speed of leader is assumed to be constant. The desired angle is defined as $\alpha_{pd}=\alpha_d+\gamma_p$, where α_d is decided by the formation shape. The relative angle tracking error is defined as $\alpha_e=\alpha_p-\alpha_{pd}$ for followers on the left, and as $\alpha_e=\alpha_{pd}-\alpha_p$ for followers on the right. Then, the control objective of the formation task is to make α_e converge to zero, i.e., $\lim_{t\to\infty}\alpha_e(t)=0$.

Implementation: In this work, we implement the DDPG algorithm [14]. A path following actor-critic network (PFACN) is proposed to solve the path following control problem. The state space is presented as $\mathcal{X}_{\text{PF}}=[\psi_e,\dot{\psi}_e,y_e,\dot{y}_e,x_e,\dot{x}_e]^T$, where $\psi_e=\gamma_p-\psi$ is the heading error relative to the path angle, and x_e is used for encouraging the vehicle to approach the endpoint. We define the action space as $\mathcal{A}_{\text{PF}}=[r_d]$. If $|\psi_e|<90°$, the reward function is designed as $r_{\text{PF}}=\lambda_x r_{x_e}+\lambda_y r_{y_e}-c_{\dot{r}_d}\dot{r}^2_d$, where $\dot{r}_d$ is used to make the control actions smoother, and $r_{y_e}=e^{-0.5y^2_e}$, $r_{x_e}=e^{-0.001x^2_e}$. Then, if $|\psi_e|\geq 90°$, the reward is designed as $r_{\text{PF}}=r_{\psi_e}$, where $r_{\psi_e}=-e^{0.1(|\psi_e|-180)}$. A formation actor-critic network (FACN) is presented to solve the formation control problem. The state space is presented as $\mathcal{X}_{\text{F}}=[\alpha_e,\dot{\alpha}_e]^T$, and the action space is defined as $\mathcal{A}_{\text{F}}=[u_d]$. Then, if $|\alpha_e|\geq 1°$, the reward function is designed as $r_{\text{F}}=e^{-0.5\alpha^2_e}$, otherwise $r_{\text{F}}=e^{-0.5\alpha^2_e}-0.2|\dot{\alpha}_e|$.

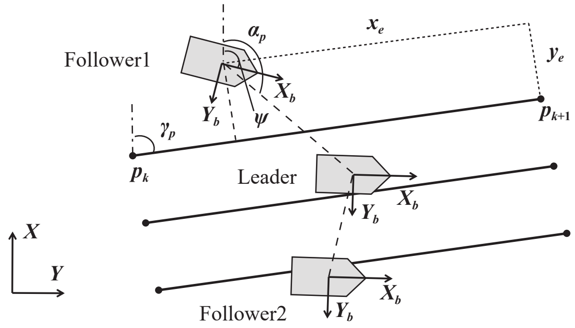


Fig. 2. Geometry of the path following and formation tasks.

For each task, both actor and critic have two hidden layers with 400 and 300 units, respectively. The activation functions for all of the hidden layers are rectified linear units (ReLU). The output layer of the actor has a hyperbolic tangent activation function, and the output layer of the critic has a linear activation function. During training, each episode has 600 training steps, with a timestep of 0.05 s. The training of the PFACN and the FACN are run with 3000 episodes and 2000 episodes, respectively. The learning rate is set to 10^{-4} for the actor and to 10^{-3} for the critic. Batches of 128 transitions are drawn randomly from a buffer of size 10^6, with the discount factor $\gamma=0.99$ and the soft target update rate $\tau=0.001$.

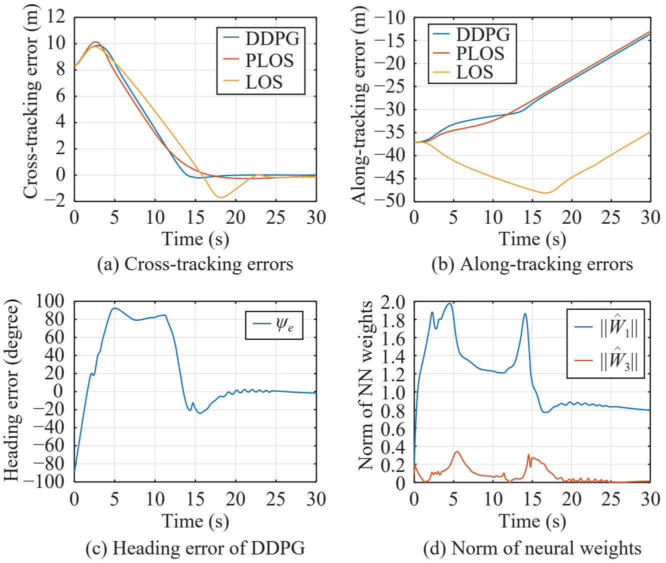
Results and discussions: The model uncertainty of USVs can be supposed as $\Delta(\eta,v)=[0.8,0.2r^2,0.2u^2+\sin(v)+0.1vr]^T$. The disturbances in the earth frame can be defined as $\omega(t)=[0.6\sin(0.7t)+1.8\cos(0.05t)-2,1.5\sin(0.06t)+0.4\cos(0.6t)-3,0]^T$. For controller design, each $\boldsymbol{S}_l(Z)$ has 11 NN nodes. The gain matrices are defined as $\Gamma_l=4I_{11\times11}$, and the initial weights W_l are zero, where $l=1,3$. The input vector of the RBF NNs is designed as $Z=[u,v,r]^T$. The control gains are chosen as $K_1=2$, $K_3=10$, $\kappa_1=0.003$, $\kappa_3=0.5$, $\alpha_1=5$, $\alpha_3=5$, $\mu_u=20$, $\mu_r=20$, $\beta_1(0)=0$, $\beta_3(0)=0$.

For training the PFACN, random constant speed is chosen for each episode to learn from different dynamic scenarios. We train a model on the source task with the reward parameters $\lambda_x=1$, $\lambda_y=1$, $c_{\dot{r}_d}=0$ first, then we adjust the reward parameters to $\lambda_x=0.5,\lambda_y=1.5$, $c_{\dot{r}_d}=0.02$, and other conditions remain the same. We transfer the whole network parameters and continue training model. For comparison, we choose a standard line-of-sight (LOS) guidance law [2] with look-ahead distance $\delta=5$, and a pure pursuit and LOS guidance law (PLOS) [15] with $k_1=0.9$, $k_2=0.18$. Fig. 3 (a) shows the DDPG approach achieves less overshoot and reaches steady-state faster. After 25 s, the root mean squared error (RMSE) values of cross-tracking are shown in Table 1, which indicates that DDPG also performs better in RMSE. Another illustration is in Fig. 3(b). Due to the reward r_{x_e} and r_{ψ_e}, DDPG quickly moves towards the endpoint, which is similar to PLOS. The low scores of LOS in r_{x_e} and r_{ψ_e} also lead to the low reward value in Table 1 . Note that tuning control parameters may affect the results of LOS guidance, these tasks still validate the performance of the DRL-based approach. From Fig. 3(c), the heading error of DDPG converges to a small neighborhood of the origin. The norms of NN weights are bounded as shown in Fig. 3(d).

By equipping with the learned PFACN, the leader and Follower1 are chosen to train the FACN. Fig. 4 (a) shows the desired relative angle is achieved, then in this case no communication is needed. However, the action in Fig. 4 (b) is very aggressive even with the penalty term in r_{F}. Instead of tuning the reward function, we propose a potential function ζ_u (motivated by [16]) to smooth the action

$$\zeta_u(\alpha_e)=\begin{cases}\rho_h(-\alpha_e)\phi(-\alpha_e), & \text{if}\quad -1°<\alpha_e<0°\\ -\rho_h(\alpha_e)\phi(\alpha_e), & \text{if}\quad 0°<\alpha_e<1°\\ 0, & \text{if}\quad |\alpha_e|\geq 1°\end{cases}\tag{5}$$

where $\rho_h(\alpha_e)=\frac{1}{5}\left(1+\cos\left(\pi\frac{(\alpha_e-1)}{4}\right)\right)$, $\phi(\alpha_e)=\alpha_e/\sqrt{1+\alpha^2_e}$. Then, if $|\alpha_e|\geq 1°$, the speed command is designed as $u_d=u_{\text{DRL}}$, otherwise $u_d=u_c+\zeta_u(\alpha_e)$. In this way we only need Wi-Fi or acoustic sensor to exchange common reference velocity u_c between neighbours at the initial time. The output of the FACN u_{DRL} is employed to quickly reach the formation, and the potential function is used to obtain slower and smoother control actions when α_e is very small. It can be



(a) Cross-tracking errors

(b) Along-tracking errors

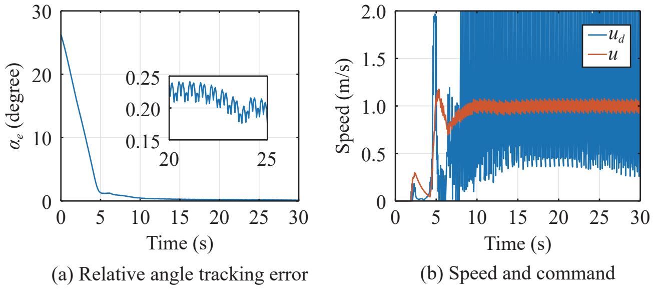
(c) Heading error of DDPG

(d) Norm of neural weights

Fig. 3. Tracking performance of the straight path following task.

Table 1. RMSE and Reward

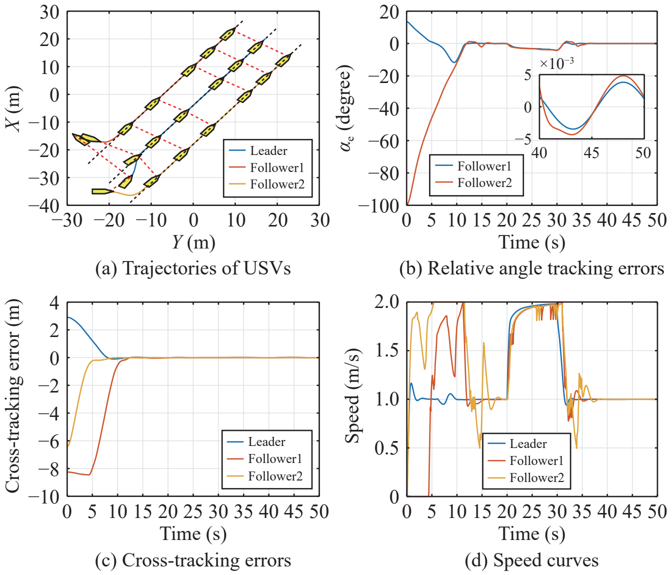
Method	RMSE	Reward
DDPG	0.0074	664.02
PLOS	0.1376	646.27
LOS	0.1576	28.82



(a) Relative angle tracking error

(b) Speed and command

Fig. 4. Evaluation of the FACN performance.



(a) Trajectories of USVs

(b) Relative angle tracking errors

(c) Cross-tracking errors

(d) Speed curves

Fig. 5. Evaluation of PFACN and FACN with the potential function.

seen from Figs. 5(a)−5(d) that in order to quickly achieve formation (α_d is set as 90° for Follower1 and as −90° for Follower2, and u_c is set as 1 m/s), Follower1 remains stationary to wait for leader, and Follower2 speeds up to chase leader at the beginning. Desired formation is obtained at 20 s. Then leader suddenly accelerates for 10 s, and followers also accelerate to maintain formation. Finally, the desired formation shape is recovered at 40 s.

Conclusion: This letter has investigated the problems of USV path following and formation control. Based on a distributed DRL method, a PFACN and a FACN have been formulated to achieve accurate guidance and adaptive formation control. The input saturation problem, the unknown disturbances and model uncertainties are addressed by an NN-based controller. Finally, numerical examples have been carried out to demonstrate the effectiveness of the proposed method.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (U2141238).

References

[1] Z. Peng, J. Wang, D. Wang, and Q.-L. Han, "An overview of recent advances in coordinated control of multiple autonomous surface vehicles," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 732–745, 2021.

[2] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, Hoboken, USA: Wiley, 2011.

[3] Z. Peng, J. Wang, and Q.-L. Han, "Path-following control of autonomous underwater vehicles subject to velocity and input constraints via neurodynamic optimization," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8724–8732, 2019.

[4] Z. Zhang, Y. Shi, Z. Zhang, and W. Yan, "New results on sliding-mode control for takagi-sugeno fuzzy multiagent systems," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1592–1604, 2019.

[5] L. Chen, R. Cui, C. Yang, and W. Yan, "Adaptive neural network control of underactuated surface vessels with guaranteed transient performance: Theory and experimental results," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 4024–4035, 2020.

[6] C. Shen, Y. Shi, and B. Buckham, "Trajectory tracking control of an autonomous underwater vehicle using Lyapunov-based model predictive control," *IEEE Trans. Ind. Electron.* , vol. 65, no. 7, pp. 5796–5805, 2018.

[7] A. B. Martinsen and A. M. Lekkas, "Straight-path following for underactuated marine vessels using deep reinforcement learning," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 329–334, 2018.

[8] J. Woo, C. Yu, and N. Kim, "Deep reinforcement learning-based controller for path following of an unmanned surface vehicle," *Ocean Eng.*, vol. 183, pp. 155–166, 2019.

[9] A. Gonzalez-Garcia, H. Castañeda, and L. Garrido, "USV path-following control based on deep reinforcement learning and adaptive control," in *Proc. Global Oceans, Singapore U.S. Gulf Coast*, 2020, pp. 1–7.

[10] Y. Wang and W. Yan, "Path parameters consensus based formation control of multiple autonomous underwater vehicles in the presence of ocean currents," in *Proc. 17th IEEE Int. Conf. Methods Models Autom. Robot.*, 2012, pp. 427–432.

[11] R. P. Jain, A. P. Aguiar, and J. B. de Sousa, "Cooperative path following of robotic vehicles using an event-based control and communication strategy," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1941–1948, 2018.

[12] X. Wang, B. Zerr, H. Thomas, B. Clement, and Z. Xie, "Pattern formation for a fleet of AUVs based on optical sensor," in *Proc. IEEE OCEANS Aberdeen*, 2017, pp. 1–9.

[13] S. Wang, F. Ma, X. Yan, P. Wu, and Y. Liu, "Adaptive and extendable control of unmanned surface vehicle formations using distributed deep reinforcement learning," *Appl. Ocean Res.*, vol. 110, p. 102590, 2021.

[14] T. P. Lillicrap, *et al.*, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–14.

[15] H. Niu, Y. Lu, A. Savvaris, and A. Tsourdos, "Efficient path following algorithm for unmanned surface vehicle," in *Proc. IEEE OCEANS Shanghai*, 2016, pp. 1–7.

[16] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, 2006.