A LARGE-SCALE PATH PLANNING ALGORITHM FOR UNDERWATER ROBOTS BASED ON DEEP REINFORCEMENT LEARNING

Wenhui Wang,* Leqing Li,* Fumeng Ye,* Yumin Peng,* and Yiming Ma*

Abstract

To ensure the effect and improve the accuracy of large-scale path planning for underwater robots, a large-scale algorithm for planning the path for underwater robots based on deep reinforcement learning is proposed. Deep reinforcement learning is analysed, and the idea, structure, network update method, and training process of deep deterministic policy gradients (DDPG) algorithm are described. A fitness learning model of the robot which under water is confirmed to describe the mathematical relationship between the geographical location and operating speed of the underwater robots. On this basis, DDPG algorithm is applied in large-scale path planning of underwater robots. TensorFlow is used to build Actor and Critic neural network structures, and design environment state models, action state spaces, and reward functions. In deep reinforcement learning, the large-scale navigation planning for the underwater robot, through exploration-online trial and error, finds the optimal search strategy, and considers obtaining the maximum expected reward during the path planning procedure, achieving the large-scale path planning for the underwater robot. According to the experimental results, the proposed algorithm demonstrates good performance in large-scale path planning for underwater robots and effectively improves both the accuracy and efficiency of the planning process.

Key Words

DDPG algorithm, reward function, deep reinforcement learning, underwater robot, large-scale path planning

1. Introduction

With the increasing scarcity of natural resources on land, many countries have accelerated the pace of exploring and developing the ocean, making people increasingly aware of the importance of human research, development, and utilisation of marine resources. Various marine technology giants have sparked a wave of research and development of underwater equipment and made intelligent underwater robots a hot research field. Intelligent underwater robots have a wide range of application possibilities and important value, and are an indispensable subfield in the field of robotics [1]. In recent years, the research and development of intelligent underwater robot technology has made rapid progress and has been applied in marine scientific research, commercial, and military fields, playing an increasingly important role. To navigate in dynamic and complex ocean terrain, key technologies are indispensable for the functionality of intelligent underwater robots [2], [3]. Due to the complex and ever-changing marine environment, known environmental information may not be very accurate, so large-scale path planning technology is needed to ensure the safety of underwater navigation for intelligent underwater robots. Therefore, studying the large-scale path planning of underwater robots has important practical significance for safe underwater navigation.

* CSG PGC Power Storage Research Institute, Guangzhou 510000, China; e-mail: wangwenhui66854@163.com
Corresponding author: Wenhui Wang

Recommended by Ponnambalam S.G.
(DOI: 10.2316/J.2024.206-1035)

2. Related Works

Lim *et al.* [4] proposed a constrained path-planning algorithm for underwater robots based on selective hybrid particle swarm optimisation (PSO) algorithm. The functionality of the path planning relies on the utilisation of two distinct algorithms for PSO, selective differential evolution-hybrid quantum PSO and adaptive PSO. Zhuang *et al.* [5] proposed a collaborative path-planning algorithm for multi-autonomous underwater robots in fluctuating marine surroundings. Using the global Legendre pseudospectral method to calculate the shortest path for cars to avoid collisions in a static environment. Krishnan *et al.* [6] analyzed the control of AUV autonomy. They utilized the powerful processing capabilities provided by machine learning and deep learning to implement robot functions.

Wang *et al.* [7] proposed a multi-behaviour critical reinforcement learning algorithm for path planning of

autonomous underwater vehicles. Ma *et al.* [8] used two different Tabu search methods to update the AUV path in real time and used polynomial coefficient solution to fit part of the path data. Bykova *et al.* [9] described two safety navigation algorithms for autonomous underwater vehicles. Cao and Zuo [10] proposed a new latent field hierarchy structure that utilises fuzzy algorithms to provide a reasonable path for AUVs in underwater environments. However, each algorithm still faced problems, such as suboptimal path planning, reduced accuracy, and slow efficiency.

Therefore, this article proposes a large-scale path-planning algorithm for underwater robots based on deep reinforcement learning.

3. Deep Reinforcement Learning

3.1 DDPG Algorithm Idea

By combining the characteristics of Actor–Critic structure [11], [12], deep Q-network algorithm, and DPG, they are applied to deep deterministic policy gradients (DDPG).

3.2 DDPG Algorithm Structure

The DDPG algorithm takes the Actor write structure as its basic structure [13], [14]. The state transition stochastic strategy is replaced by the deterministic strategy [15], [16].

Both Actor and critical are composed of two networks, with the estimated network of Actor denoted as $\mu(s|\theta^{\mu})$; the target network of Actor is labeled as $\mu'\left(s\left|\theta^{\mu'}\right.\right)$; critical's estimated network is recorded as $Q\left(s,a\left|\theta^{Q}\right.\right)$; the target network of critical is denoted as $Q'\left(s,a\left|\theta^{Q'}\right.\right)$; θ represents the angle between the target point and the robot direction. The DDPG algorithm reduces error values by updating the network. The estimation network of Actor updates parameters according to (1).

$$\nabla_{\theta^{\mu}} J_{\beta}(\mu) = \frac{1}{N}\sum_{t}\left(\nabla_{a} Q(s,a\left|\theta^{Q}\right.)\left|_{S=s_t,A=\mu(s_t)}\right.\right. \cdot \nabla_{\theta^{\mu}}\mu\left(s\left|\theta^{u}\right.\right)\left|_{S=s_t}\right.\right) \tag{1}$$

In (1), s denotes the current state; a refers to expected value; $\nabla_a Q(s,a\left|\theta^{Q}\right.)$ represents the gradient of the critical estimation network; J means the objective function for policy gradient update, and β indicates the azimuth angle, its state s_t will change to the state s_{t+1}, μ refers to the commands given by the policy network. The calculation method of the estimated LOSS is shown in (2).

$$L = \frac{1}{N}\sum_{t}\left(U_t - Q(s_t,a_t\left|\theta^{Q}\right.)\right)^2 \tag{2}$$

In (2), U_t expresses the target value of the temporal difference.

The update of the Actor section is shown in (3).

$$\begin{cases}\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}\end{cases} \tag{3}$$

In (3), τ represents the update frequency of the target network [17], [18]. Normalise the distribution of robot obstacles.

$$\overline{d}_i = \frac{d_i}{\max(\overline{d}_1,\overline{d}_2,\ldots,\overline{d}_{180})} \tag{4}$$

In (4), $\overline{d}_i$ means the vector after dimensionality reduction; d_i represents the vector before dimensionality reduction.

Setting the feedback distance of the laser sensor to be less than the safety threshold range indicates that a collision has occurred and the robot needs to be guided. The reward and punishment mechanism is shown in (5).

$$r(s_t,a_t) = \begin{cases} r_{\text{arrive}} & \rho_t < d_{\text{goal}} \\ r_{\text{collision}} & \min(\overline{d}_1,\overline{d}_2,\ldots,\overline{d}_{180}) < d_{\text{collision}} \\ c_r(\rho_{t-1}-\rho_t) - C & \end{cases} \tag{5}$$

In (5), d_{goal} is the set threshold; $d_{\text{collision}}$ stands for setting a security threshold; c_r and C are constants. The performance function δ of defining policy α is shown in (6):

$$\delta(\alpha) = R_{w\sim\varepsilon}\left[Q^{\alpha}(w,\alpha(e))\right] \tag{6}$$

In (6), ε refers to the distribution function of the state w under strategy α, and $Q^{\alpha}(w,\alpha(e))$ is the evaluation value obtained by strategy α. The gradient of the determination policy α shown in (7) is obtained:

$$\nabla_{\chi^{\alpha}}\delta = R_{w\sim\varepsilon}\left[\nabla_{\chi^{\alpha}} Q^{\alpha}\left(w,e\left|\chi^{Q}\right.\right)\left|_{w=w_t,e=\alpha(w_t|\chi^{\alpha})}\right.\right. \nabla_{\chi^{\alpha}}\alpha\left(w\left|\chi^{\alpha}\right.\right)\left|_{w=w_t}\right.\right] \tag{7}$$

Experience playback strategy randomly selects data update parameters from the experience library.

3.3 DDPG Algorithm Network Update Method

The policy network α of the optimal policy is shown in (8):

$$\alpha_{\text{best}} = \text{argmax}\,\delta(\alpha_{\chi}) \tag{8}$$

The gradient descent method [19], [20] is used to update the policy network's network parameters according to (9):

$$\chi^{\alpha}_{t+1} \leftarrow \chi^{\alpha}_{t} - \phi_e\left(-\nabla\delta\left(\alpha^{\chi^{\alpha}_t}\right)\right) \tag{9}$$

In the Actor–Critic structure, it measures the evaluation network by loss function φ shown in (10):

$$\varphi\left(\chi^{Q}\right) = \left(E_t - Q\left(w_t,e_t\left|\chi^{Q}\right.\right)\right)^2 \tag{10}$$

In (10), Q means the estimated evaluation given by the evaluation network to the action e_t, and E_t expresses the

actual evaluation obtained by the action e_t in the current state.

At this time, the actual evaluation y_t is calculated in (11):

$$E_t = r_t + \beta Q'\left(w_{t+1}, \alpha'\left(w_{t+1}\left|\chi^{\alpha'}\right.\right)\left|\chi^{Q'}\right.\right) \tag{11}$$

In (11), $Q'\left(w_{t+1}, \alpha'\left(w_{t+1}\left|\chi^{\alpha'}\right.\right)\right)$ is the production output of the evaluation target web; $\alpha'\left(w_{t+1}\left|\chi^{\alpha'}\right.\right)$ denotes the output value of the target strategy network, and $\chi^{Q'}$ and $\chi^{\alpha'}$ are the network parameters of the target evaluation network and the target strategy network, respectively, the reference value is χ^{Q}.

The gradient of the evaluation network loss function can be obtained as (12):

$$\begin{aligned}\nabla_{\chi^Q}\varphi\left(\chi^Q\right) = &\left(E_t - Q\left(w_t, e_t\left|\chi^Q\right.\right)\right)\\ &\nabla_{\chi^Q}Q\left(w_t, e_t\left|\chi^Q\right.\right)\end{aligned} \tag{12}$$

This enables DDPG network updates and training.

4. Large-Scale Path Planning Algorithm for Underwater Robot

4.1 Establishment of the Kinematics Model of the Underwater Robot

The kinematic model of the underwater robot is established. Assuming that the speed of the driving wheel is V, the position of the underwater robot at time i is represented by the 3D state vector $U\left(x_i, y_i, \zeta\right)$ Among them, the coordinate (x_i, y_i) denotes the reference point's position of the robot which is underwater in the coordinate system, with the midpoint of the rear axis of the underwater robot as the reference point ζ. The angle of direction is ζ. The steering angle is β, and the steering wheel is l_1. The following (13) characterises the system kinematics:

$$\begin{bmatrix} x_i \\ y_i \\ \zeta \end{bmatrix} = \begin{bmatrix} \cos\zeta 0 \\ \sin\zeta 0 \\ 0 \quad 1 \end{bmatrix} \begin{bmatrix} \zeta \\ Y\tan\eta \\ l_1 \end{bmatrix} \tag{13}$$

4.2 Principle of DDPG Path Planning Algorithm

The DDPG algorithm is employed for the large-scale path planning of the underwater robot.

In the DDPG algorithm, critical network on deterministic χ^{α} policy α and value-action function Q and the parameter is χ^{α}, and the other is χ^{Q}, respectively. The iterative update of the algorithm involves the following steps: use sample accumulation and apply a loss function to update parameters, then use the Adam optimiser for iterative updates.

This article uses the Actor neural network structure built by TensorFlow and the Critic neural network structure constructed by TensorFlow.

4.3 Environmental State Model Design

It supposes that D_{fg} represents the distance variable, where the environmental state parameter is defined as (14):

$$W = (\theta, D_h, D_{1g}, D_{2g}, D_{3g}, D_{4g}) \tag{14}$$

In (14), the variable θ denotes the angle between the robot's heading and the target point, while D_h indicates the distance between the sensor responsible for utilising LiDAR technology and the target point. $D_{1g}, D_{2g}, D_{3g}, D_{4g}$ are the distance between the radar and the nearest obstacle in each area, respectively. Calculate the distance between obstacle points and the underwater robot coordinate system

$$\begin{aligned}\varpi_i &= C_{\varpi} + D_{\mathrm{fg}}\sin\theta\\ \omega_i &= C_{\omega} + D_{\mathrm{fg}}\cos\theta\end{aligned} \tag{15}$$

In (15), C_{ϖ} and C_{ω} indicate the position movement of LiDAR relative to the same coordinate system. The area where the target point is located is specifically divided. The variable D_{fg} is divided into five levels of "collision risk," "warning," "near," "safer," and "safe," and the results are quantified as (16).

$$D_f = \begin{cases} 0, D_{\mathrm{fg}} < 6 \\ 1, 6 \le D_{\mathrm{fg}} < 12 \\ 2, 12 \le D_{\mathrm{fg}} < 18 \\ 3, 18 \le D_{\mathrm{fg}} < 24 \\ 4, 24 \le D_{\mathrm{fg}} \le 30 \end{cases} \tag{16}$$

From this, the input state space of the model contains $5 \times 5 \times 5 \times 5 \times 9 = 5625$.

4.4 Action State Space Design and Selection Strategy

This paper adopts the ε-greedy strategy. The underwater robot can make correct behaviours at time $i+1$ based on the obtained direction angle and other information. More characteristics of underwater robots need to be considered, and the continuity of output actions and the feasibility of behaviours must be guaranteed.

5. Experimental Process

5.1 Experimental Data and Environment

To showcase the value of DRL-based large-scale path planning algorithms for underwater robots, a 20 × 20 km seabed topographic map has been generated. The seabed environment is divided into 20 × 20 grids of the same size. The seabed topographic map obtained from the study is shown in Fig. 1.

The hardware platform configuration of the simulation experiment is represented in Table 1.

The algorithms proposed by Lim *et al.* [4] and Zhuang *et al.* [5] and the proposed algorithm in this

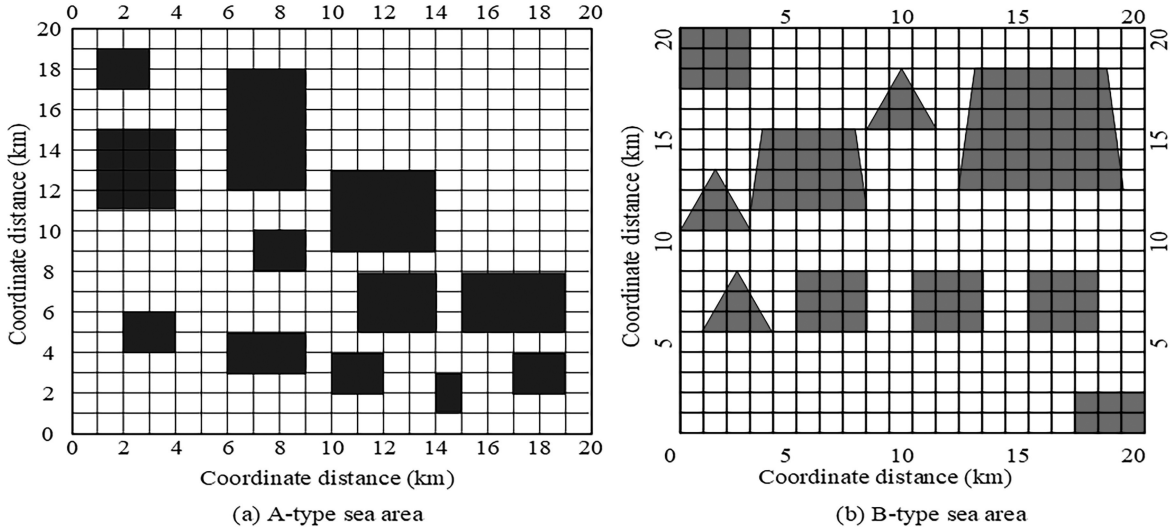


Figure 1. Simulation experiment of submarine topographic map.

Table 1
The Hardware Configuration of the Simulation Experiment Platform

Name	**Model**
Central processing unit CPU	Intel i5 7300HQ
Graphics processing unit GPU	NVIDIA GTX 1050Ti
Memory	16GB DDR4
Operating system	Windows 10 64 bit

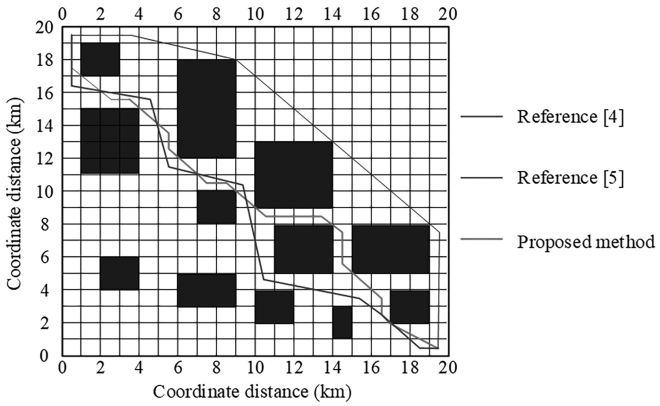


Figure 2. 2D path planning results for type a sea area.

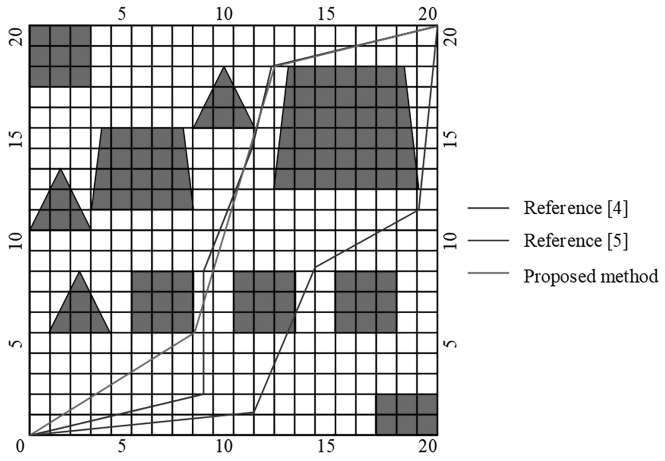


Figure 3. Path planning under different algorithms.

paper were, respectively, used to compare the path planning length, success rate, and efficiency of different algorithms.

5.2 Planning and Simulation of 2D Paths

The paths generated by different algorithms are shown in Fig. 2.

In Fig. 2, the improved algorithm could enable AUV to find the shortest and optimal paths even in front of obstacles. The contrast of the large-scale path planning accuracy of the underwater robot with different algorithms is shown in Table 2.

From Table 2, when the number of iterations reached 500, the success rate of the proposed algorithm was more than 97.3%. This algorithm has a high success rate and accuracy. The path planning formed under different algorithms for B-type sea areas in the experiment is shown in Fig. 3.

From Fig. 3, the algorithm proposed in this study provides the optimal path plan for robots in path planning,

Table 2
Comparison Results of the Success Rate of Large-Scale Path Planning of Underwater Robots with Different Algorithms

Iterations/Time	**The Proposed Algorithm/%**	**Algorithm Proposed by Lim *et al.* [4]/%**	**Algorithm Proposed by Zhuang *et al.* [5]/%**
100	96.4	89.5	86.7
200	96.9	90.2	85.4
300	97.6	88.4	86.9
400	98.2	88.8	86.3
500	97.5	89.1	86.1

Table 3
Comparison Results of Simulation Data

Rating Indicators	**Reference [4]**	**Reference [5]**	**Algorithm in This Article**
Path length (km)	35.24	42.61	30.04
Accumulated corner (rad)	25.28	23.56	12.67
Running time (h)	3.89	4.7	3.3

Table 4
Comparison Results of Large-Scale Path Planning Time and Path for Underwater Robots Using Different Algorithms

Iterations/Time, Distance	**The Proposed Algorithm**	**Algorithm Proposed by Lim *et al.* [4]**	**Algorithm Proposed by Zhuang *et al.* [5]**
100	3.4 s/58.12 km	6.8 s/67.12 km	8.9 s/78.16 km
200	5.9 s/35.18 km	9.5 s/42.23 km	11.2 s/59.41 km
300	7.6 s/30.04 km	12.9 s/35.24 km	14.1 s/42.84 km
400	9.5 s/30.04 km	15.1 s/35.24 km	17.2 s/42.61 km
500	12.5 s/30.04 km	17.9 s/35.24 km	19.6 s/42.61 km

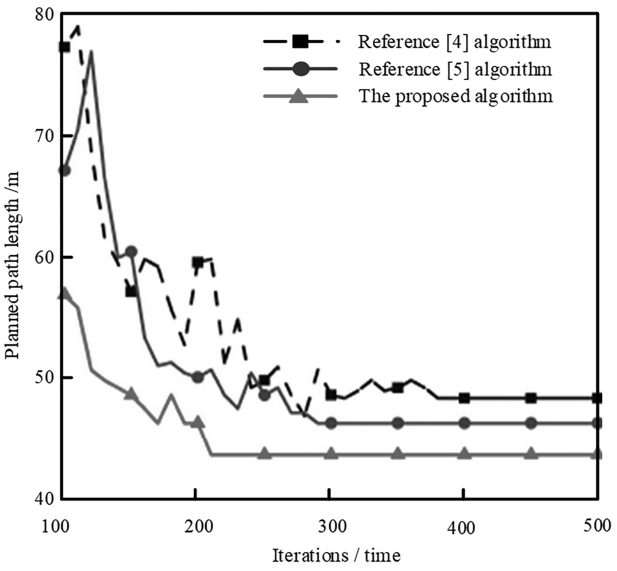


Figure 4. Comparison results of large-scale path planning lengths of underwater robots with different algorithms.

and the stability of the robot is also within the requirements and has the shortest path distance.

The contrast of the large-scale path planning length of the underwater robot with different algorithms is shown in Fig. 4.

From Fig. 4, the large-scale path planning length of this proposed algorithm is only 28.13 km. And the results are shown in Table 3.

From Table 3, the path planning under the algorithm in this article is 30.04 km, with a cumulative angle of 12.67°, and a running time of 3.3 h. The optimal planning path and minimum turning angle meet the stability requirements.

5.3 Comparison Results of Large-Scale Path Planning Efficiency of Underwater Robots

The comparison results of the large-scale path planning time of the underwater robot with different algorithms are shown in Table 4.

From Table 4, the proposed algorithm only takes 12.5 s and the planned path length is only 30.04 km.

6. Conclusion

The DDPG algorithm was used to optimise the control of underwater robots. The results indicated that the method proposed had good effectiveness, which could effectively improve the accuracy and efficiency of large-scale path planning for underwater robots. But, the simulation experiments conducted by this algorithm are static, and in actual dynamic environments, further research is needed on the algorithm.

Acknowledgement

This work was supported by China Southern Power Grid Power Generation Co., Ltd. "Research on Key Technologies and Equipment Development of Long Tunnel Underwater Robot in Deep Water, Moving Water and Muddy Water" (No. 020000k52180012).

References

[1] T. Zhang, H. Zhou, J. Wang, Z. Liu, J. Xin, and Y. Pang, Optimum design of a small intelligent ocean exploration underwater vehicle, *Ocean Engineering, 184*, 2019, 40–58.

[2] L.T. Aloba, Synthesis of intelligent automatic control system of an autonomous underwater vehicle as a group agent, *Shipbuilding and Marine Infrastructure, 1*(11), 2019, 74–84.

[3] G.S. Lima, S. Trimpe, and W.M. Bessa, Sliding mode control with Gaussian process regression for underwater robots, *Journal of Intelligent & Robotic Systems, 99*(3), 2020, 487–498.

[4] H.S. Lim, S. Fan, C.K.H. Chin, S. Chai, and E. Kim, Constrained path planning of autonomous underwater vehicle using selectively-hybridized particle swarm optimization algorithms, *IFAC-PapersOnLine, 52*(21), 2019, 315–322.

[5] Y. Zhuang, H. Huang, S. Sharma, D. Xu, and Q. Zhang, Cooperative path planning of multiple autonomous underwater vehicles operating in dynamic ocean environment, *ISA Transactions, 94*, 2019, 174–186.

[6] A. Krishnan, K. Parvathy, and V. Donekal, ML and robotics integrated with AUVs for sub-aquatic applications, *International Journal of Robotics and Automation*, *6*(2), 2020, 1–16.

[7] Z. Wang, S. Zhang, X. Feng, and Y. Sui, Autonomous underwater vehicle path planning based on actor-multi-critic reinforcement learning, *Proceedings of the Institution of Mechanical Engineers*, *Part I: Journal of Systems and Control Engineering*, *235*(10), 2021, 1787–1796.

[8] X.-W. Ma, Y.-L. Chen, G.-Q. Bai, Y.-B. Sha, and J. Liu, Multi-autonomous underwater vehicles collaboratively search for intelligent targets in an unknown environment in the presence of interception, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *235*(9), 2021, 1539–1554.

[9] V.S. Bykova, A.I. Mashoshin, and I.V. Pashkevich, Safe navigation algorithm for autonomous underwater vehicles, *Giroskopiya i Navigatsiya*, *29*(1), 2021, 97–110.

[10] X. Cao and F. Zuo, A fuzzy-based potential field hierarchical reinforcement learning approach for target hunting by multi-AUV in 3-D underwater environments, *International Journal of Control*, *94*(5), 2021, 1334–1343.

[11] D.C. Cicek, E. Duran, B. Saglam, F.B. Mutlu, and S.S. Kozat, Off-policy correction for deep deterministic policy gradient algorithms *via* batch prioritized experience replay, *Proc. 2021 IEEE 33rd International Conf. on Tools with Artificial Intelligence (ICTAI)*, Washington, DC, 2021, 1255–1262.

[12] Z. Shi, Z. Jin, and H. Wang, Satellite attitude tracking decision method based on deep deterministic policy gradient for moving target observation, *Proc. 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conf. (IAEAC)*, Chongqing, China, 2021, 868–872.

[13] L. Chen, Y. Zhao, H. Zhao, and B. Zheng, Non-communication decentralized multi-robot collision avoidance in grid map workspace with double deep Q-network, *Sensors, 21*(3), 2021, 841–849.

[14] S. Sanaye and A. Sarrafi, A novel energy management method based on deep Q network algorithm for low operating cost of an integrated hybrid system, *Energy Reports, 7*(3), 2021, 2647–2663.

[15] H. Ahmadi, M. Rafiei, M.A. Igder, M. Gheisarnejad, M.-H. Khooban, An energy efficient solution for fuel cell heat recovery in zero-emission ferry boats: Deep deterministic policy gradient, *IEEE Transactions on Vehicular Technology, 70*(8), 2021, 7571–7581.

[16] Z. Ma, Q. Huo, T. Zhang, J. Hao, and W. Wang, Deep deterministic policy gradient based energy management strategy for hybrid electric tracked vehicle with online updating mechanism, *IEEE Access, 9*, 2021, 7280–7292.

[17] Q. Shen, Seeking for passenger under dynamic prices: A Markov decision process approach, *Journal of Computer and Communications, 9*(12), 2021, 80–97.

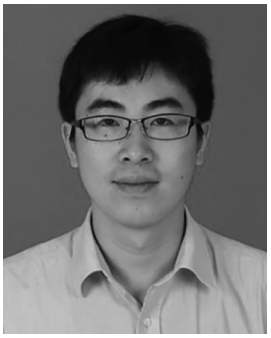
[18] D. Qiao, Y. Xie, Q. Jia, and T. Yao, Research on fleet control based on Markov random channel allocation, *Computer Simulation, 38*(9), 2021, 138–144.

[19] L. Zeng, Z. Yang, S. Liao, C. Yang, and D. Li, Metamorphosis relationship generation based on fixed memory step gradient descent method with noise, *Proc. 2021 IEEE 12th International Conf. on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2021, 282–286.

[20] Y. Wang, Y. He, and Z. Zhu, Study on fast speed fractional order gradient descent method and its application in neural networks, *Neurocomputing, 489*, 2022, 366–376.

Biographies

Wenhui Wang was born in October 1983, is a Senior Engineer. He received the graduation degree in detection technology and automation devices from North China Electric Power University in 2010. His research areas include automation, intelligence, and robot automatic control in hydropower plants. He has published over 10 academic articles. He is currently with the Energy Storage Research Institute of China Southern Power Grid Peak shaving Frequency Modulation Power Generation Co., Ltd.



Leqing Li was born in December 1986, is an Engineer. He received the graduation degree in electrical engineering and automation from the Changsha University of Technology in 2010. His research field involves electrical automation. He has published over 10 academic articles. He is currently with the Energy Storage Research Institute of China Southern Power Grid Peak shaving Frequency Modulation Power Generation Co., Ltd.

Fumeng Ye was born in July 1965, is a Senior Engineer. He received the graduation degree in hydraulics and river dynamics from Tsinghua University, Beijing, in 1993. His research field involves automation and intelligence of hydraulic detection in hydropower plants. He has published over 10 academic articles. He is currently with the Energy Storage Research Institute of China Southern Power Grid Peak shaving Frequency Modulation Power Generation Co., Ltd.

Yumin Peng was born in December 1979, is a Senior Engineer. He received the graduation degree in industrial automation from Wuhan University in 2003. His research field involves automatic control technology in power plants. He has published over 10 academic articles. He is currently with the Energy Storage Research Institute of China Southern Power Grid Peak shaving Frequency Modulation Power Generation Co., Ltd.

Yiming Ma was born in July 1995, is an Engineer. He received the graduation degree in electrical engineering from the Huazhong University of Science and Technology in 2022. His research field involves motor design and analysis of operating characteristics. He has published over 10 academic articles. He is currently with the Energy Storage Research Institute of China Southern Power Grid Peak shaving Frequency Modulation Power Generation Co., Ltd.