Proceedings of 2021 IEEE
International Conference on Mechatronics and Automation
August 8 - 11, Takamatsu, Japan

The Vector Control Scheme for Amphibious Spherical Robots Based on Reinforcement Learning

He Yin[1, 2], Shuxiang Guo[1, 2*], Liwei Shi[1*], Mugen Zhou[1], Xihuan Hou[1], Zan Li[1], Debin Xia[1]

[1] *Key Laboratory of Convergence Medical Engineering System and Healthcare Technology, the Ministry of Industry and Information Technology, School of Life Science, Beijing Institute of Technology, No.5, Zhongguancun South Street, Haidian District, Beijing 100081, China*

[2] *Faculty of Engineering, Kagawa University, 2217-20 Hayashi-cho, Takamatsu, Kagawa, Japan*

E-mails: yinhe@bit.edu.cn; guoshuxiang@bit.edu.cn; shiliwei@bit.edu.cn,

* Corresponding author

Abstract **–Due to variable underwater working conditions and unfavorable environments, it is difficult to design a controller suitable for underwater robots. This paper uses the adaptive ability of reinforcement learning to propose a two-layer network framework based on reinforcement learning to realize the control of amphibious spherical robots. The upper planning layer mainly plans the total torque of the robot at each moment according to the desired position and speed. The lower control layer mainly configures the parameters of the four machine legs according to the planning instructions of the upper planning layer. Through the cooperation of the planning layer and the control layer, the adaptive motion control of the amphibious spherical robot can finally be realized. Finally, the proposed scheme was verified on a simulated amphibious spherical robot.**

Index Terms - Reinforcement learning; Amphibious spherical robot; Motion control.

I. INTRODUCTION

Compared with other types of underwater robot platforms, Autonomous Underwater Vehicles (AUV) have the advantages of high autonomy and large detection range. AUV has been widely used in the fields of marine environmental monitoring, resource investigation, security and defense, etc. Due to the complexity and time-varying nature of the underwater environment and the uncertainty of the underwater robot system, these factors will have an impact on the stability and reliability of the underwater robot system. Therefore, the design of an accurate and reliable controller suitable for underwater robots has attracted many scholars' attention [1], [2]. With the rapid development of artificial intelligence technology, its application in unmanned systems has gradually become the focus of attention of various countries [3]. Through the use of advanced artificial intelligence methods, the AUV's environmental perception, behavioral decision-making and underlying control capabilities can be improved, so that it can efficiently complete underwater tasks. In recent years, researchers have tried to apply various intelligent methods to the AUV system platform to improve its autonomy and intelligence.

The design of the control method is one of the most important parts of the control system, which enables the underwater robot to follow instructions to move accurately in the underwater environment. The design of the control method affects the response speed, stabilization time and overshoot amplitude of the robot in the process of motion. Due to the particularity of the underwater environment and the motion control of underwater robots with multiple degrees of freedom, the design of underwater robot control is more difficult than that of land robots such as unmanned vehicles. At present, the conventional controller methods applied to underwater robots mainly include PID control, fuzzy control, sliding mode control and adaptive control. Tanakitkom K et al. proposed a low computational cost PID-based control system for driving underwater robots [4]. Behrooz R et al. proposed an optimized fuzzy control algorithm that can realize 3D path planning of underwater robots in complex underwater environments through sonar models [5]. Kantapon developed a sliding mode heading control system for over-actuated, hover-capable AUVs operating over a range of forward speeds [6]. Lakhekar G V combines fuzzy control with adaptive control and dynamically adjusts the adaptive law of controller parameters through fuzzy logic [7]. However, these methods have some shortcomings that the model parameters are difficult to determine in practical applications. In order to simplify the problem, many controllers use linear models or decouple the models. This setting fails to fully consider the uncertainties and special disturbances of the underwater environment, and it is difficult to complete the precise autonomous control of the AUV.

The controller of reinforcement learning maximizes the cumulative reward obtained by interacting with the environment, and can achieve autonomous control [8]. Reinforcement learning does not have to rely on the dynamic model of the system platform, which weakens the influence of model accuracy on control. And it has better adaptive ability in dynamic environment, which makes it attract the attention of robot control field. At present, in the field of unmanned aerial vehicle (UAV) control and unmanned driving, the application of reinforcement learning has achieved many results. Fernandez-Gauna et al. used continuous acotr-critic automatic learning machine (CACLA) to achieve AUV Speed control [9]. Walters et al. adopted a model-based dynamic programming method for AUV control experiments in a real environment [10]. The dynamic model is obtained by real-time learning in experiments. Carlucho et al. designed a deep reinforcement learning method with a deterministic behavior-evaluation framework based on the DDPG method to realize the underlying control of AUV [11]. Knudsen et al. designed a dual controller [12]. The controller based on the depth deterministic

strategy gradient method is responsible for the movement in the horizontal plane, while the PD controller is responsible for the movement in the vertical plane.

Therefore, we hope to propose a motion control scheme suitable for amphibious spherical robots, which can adapt to variable underwater working conditions and has versatility. In response to these needs, this paper uses the adaptive ability of reinforcement learning to propose a two-layer network framework based on reinforcement learning to realize the control of the amphibious spherical robot. First, the combined force control of the amphibious robot is achieved through the training of the lower control layer. Then, by calling the lower control layer to train the upper planning layer, the total torque of the robot at each moment can be planned according to the desired position and speed. Through the cooperation of the planning layer and the control layer, the movement of the amphibious spherical robot can finally be realized. Compared with traditional control strategies, the proposed scheme does not depend on precise model parameters and has strong adaptability. In addition, the two-layer architecture design can shorten the training time of reinforcement learning and has strong versatility.

The rest of this paper is organized as follows. Section II presents a brief introduction about our new generation amphibious spherical robot. The background knowledge about reinforcement learning and the general framework of our proposed method for the amphibious spherical robots is presented in Section III. Experiments and results are provided in Section IV to assess the performance of the proposed approach. Section V provides a conclusion of the paper.

II. The Structure of the Amphibious Spherical Robot

In this section, we give a brief introduction about our new generation amphibious spherical robot and sensors mounted on the robot.

On the basis of the robots mentioned in references [13]-[17], a new generation of amphibious spherical robots with more perfect performance has been developed. The structure of the robot is shown in Fig. 1. The appearance of the amphibious spherical robot is designed to be spherical in order to reduce the resistance of underwater movement. The amphibious spherical robot consists of two parts: the control part and the drive part. The control part is located in the upper hemisphere of the robot. It is mainly responsible for the perception of the environment and the acquisition of its own state information, as well as the control of the robot. It is equipped with binocular industrial camera inertial sensor, GPS positioning module and depth sensor. The driving part is located in the lower hemisphere of the robot. It is mainly responsible for the movement of the amphibious spherical robot on the road and in the water. The drive part includes water jet and steering gear. The maximum speed of this amphibious robot in the water can reach 60 cm/s, and the maximum crawling speed of the robot on land is 6.05cm/s.

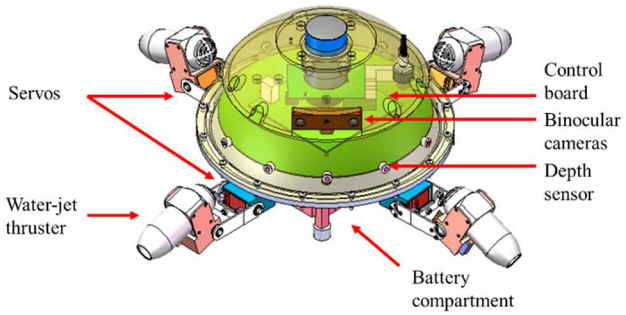

Fig. 1. The structure of proposed amphibious spherical robot.

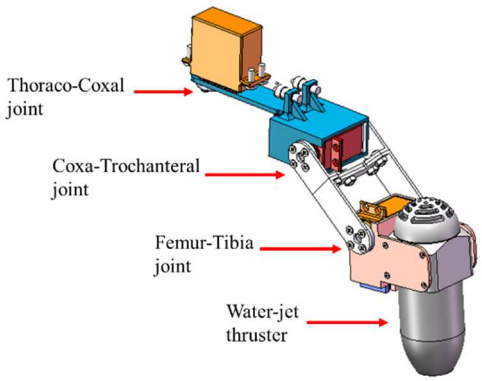

Fig. 2. The structure of the mechanical leg.

In order to enable the robot to work both on land and underwater, a mechanical leg is designed, and the structure is shown in Fig. 2. The mechanical leg consists of three joints, three connecting rods and a water-jet thruster. The three joints are TC joint (Thoraco-Coxal joint), CTr joint (Coxa-Trochanteral joint) and FTi joint (Femur-Tibia joint). TC joint, CTr joint and FTi joint correspond to TC servo, CTr servo and FTi servo respectively. The three connecting rods are hip bone connecting rod, femoral connecting rod and tibia connecting rod. The water jet thruster can generate thrust in the water to propel the robot. The servo on each mechanical leg can control the angle of the mechanical leg to change the direction of thrust generated by the water jet thruster. By configuring the parameters of the servo and the water jet thruster, the desired torque can be generated, so as to realize the motion control of the robot.

III. Method

In this section, a reinforcement learning scheme suitable for amphibious spherical robots is proposed. It is used to control the underwater movement of amphibious spherical robots.

A. Reinforcement learning statement

In the past, the application of reinforcement learning in the field of robot control was limited to low-dimensional discrete state space and action space. However, complex tasks in the real world usually have high-dimensional state spaces and continuous action spaces. In 2015, the DeepMind team proposed a DQN algorithm combining deep neural networks and reinforcement learning [18], which solved the problem of high-dimensional input. But it is still difficult to apply DQN to continuous-action tasks. On the basis of deep reinforcement

learning, Lillicrap et al. designed a deep deterministic policy gradient (DDPG) method for the continuous state/action space to realize the control of continuous actions.

Reinforcement learning generally contains two main parts: Agent and environment. Its basic idea is to make the agent learn the strategy of completing specific tasks in the process of interaction with the environment. Agents continuously adjust their strategies to deal with the environment according to the strategies they have learned and the feedback they have obtained from the environment until they obtain the maximum reward. Reinforcement learning problems can be modeled by Markov Decision Process (MDP). MDP is defined by the four-tuple <*S, A, R, W*>, where *S* is the set of environmental states, A is the set of agent actions, R is the reward function, and W is the state transition function. In each time, the agent can get an observation S_t of the current state. Let $\pi(a|s) = P(A_t = a|S_t = s)$ be the probability that the agent will take action A_t when the state is S_t, which is the action strategy of the agent. In each time, the agent can get an observation S_t of the current state. Then the Agent takes action a_t based on the current state S_t and action strategy. At time t+1 after the end of the action, the state of the environment changes, and the agent gets rewards R_{t+1} and observations of the new state S_{t+1}. Then the agent adjusts its action strategy based on the rules determined in advance. This process is continuously executed until the number of learning times or learning goals are reached. The cumulative reward U in the learning process is the attenuated sum of all rewards obtained, which can be defined as:

$$R_t = R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots \quad (1)$$

where $\gamma \in [0,1]$ is the reward discount factor. Since the purpose of reinforcement learning is to obtain the optimal action strategy by maximizing the cumulative reward, it is necessary to consider the influence of the state of the agent and the actions taken on the cumulative reward. Then the action value function $Q_\pi(s,a)$ can be defined as

$$Q_\pi(s,a) = E(U_t|S_t = s, A_t = a) \quad (2)$$

The DDPG structure includes an actor network with a parameter of θ^π and a Critic network with a parameter of θ^Q. The actor network is used to generate the action strategy π of the agent, and the critic network is used to judge the quality of the action and guide the update of the action. Since the learning process of a single network is not stable, the strategy network and the value network are divided into a current network and a target network respectively. The current network and the target network have the same structure and different update frequencies. The Critic current network is responsible for updating the value network parameter θ^Q and calculating the current Q. The Critic target network is responsible for calculating the Q' part of the target Q. The loss function of the current value network is defined as

$$J(\theta^Q) = \frac{1}{m}\sum_{j=1}^{m} \omega^j (y^j - Q(S_t^j, A_t^j, \theta^Q)) \quad (3)$$

where

$$y^j = \begin{cases} R^j, end^j \\ R^j + \gamma Q(S_{t+1}^j, A_{t+1}^j, \theta^{\acute{Q}}), not\ end^j \end{cases} \quad (4)$$

m is the number of samples, ω^j is the weight corresponding to sample j, $Q(S_t^j, A_t^j, \theta^Q)$ is the action value of sample j when the state is S_t, which is evaluated by the current value network, y^j is the target action value of sample j calculated by the target value network. R^j is the instant reward obtained by the sample j taking action A_t when the state is S_t, and γ is the discount factor. The actor current network is responsible for updating the policy network parameters θ^π and selecting the current action A_t according to the current state S_t. The actor target network is responsible for selecting the optimal next action $\acute{A}_t$ according to the next state $\acute{S}_t$ sampled in the empirical playback pool. The loss function of the current actor network is defined as

$$J(\theta^\pi) = -\frac{1}{m}\sum_{j=1}^{m} Q(S_t^j, A_t^j, \theta^Q) \quad (5)$$

The target value network and target strategy network are updated in the following ways:

$$\theta^{\acute{Q}} \leftarrow \tau\theta^Q + (1-\tau)\theta^{\acute{Q}} \quad (6)$$

$$\theta^{\acute{\pi}} \leftarrow \tau\theta^\pi + (1-\tau)\theta^{\acute{\pi}} \quad (7)$$

where τ is the soft update coefficient, and the update speed of the neural network can be controlled by adjusting τ.

B. General Framework

Considering the motion pattern of the amphibious spherical robot in the underwater environment, a two-layer deep reinforcement learning network based on DDPG is proposed. The structure diagram of the whole scheme is shown in Fig.3. The proposed control scheme can control the amphibious spherical robot to generate the corresponding leg angle and thrust configuration according to the desired position, so as to realize the underwater motion control of the amphibious spherical robot.

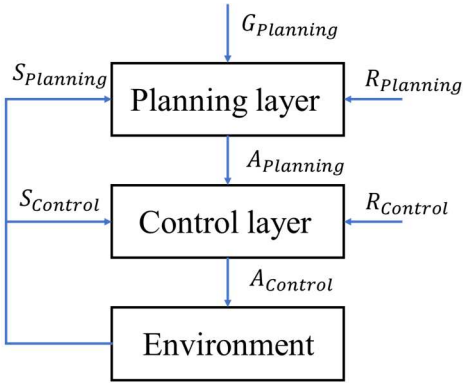

Fig.3. General framework for the proposed two-layer reinforcement learning neural network.

The proposed two-layer reinforcement learning neural network includes two parts: the upper planning layer and the lower control layer. The operating frequency of the upper planning layer is 1 Hz, and the operating frequency of the lower control layer is 10 Hz. Avoid damaging the robot by setting the drive interval. The upper planning layer mainly plans the total torque of the robot at each moment according to the desired position and speed. The input of the upper planning layer includes the desired position and speed and the state of the robot. The output of the upper planning layer is the total torque at time *t*, which is mainly used to guide the lower controller to produce corresponding motion. The lower control layer mainly configures the parameters of the four machine legs according to

the planning instructions of the upper planning layer. The input of the lower control layer is the total output torque of the upper planning layer. The output of the lower control layer is the parameter configuration of the four machine legs, which can configure the angle and thrust of the four machine legs to control the movement of the robot. Through the monitoring of sensors, the observation of the robot state can be obtained, which is the feedback of the environment. By judging whether the robot achieves the expected goal, the upper planning layer can be rewarded $R_{Planning}$. By judging whether the robot generates a total torque $A_{Planning}$, the lower control layer can be rewarded $R_{Control}$.

C. The Lower Control Layer

The responsibility of the lower control layer is to generate a suitable machine leg configuration based on the control signal of the upper planning layer, so as to generate the desired resultant force to push the robot. Each time, the input of the lower control layer includes the observation of the robot state and the expected resultant force from the upper planning layer. The output of the control layer is the state configuration of the robot's four legs.

In the process of robot motion control, we simplified the robot model. We fixed the tibial joints and tibial joints of the amphibious spherical robot so that the amphibious spherical robot can only move on the surface of the water. Therefore, the movement of the robot can be realized by the angle of the hip joint of each leg and the thrust of the water jet. The relationship between the torque and the configuration of the leg parameters is shown in Fig.4. The state $S_{Control}$ of the robot can be represented by a vector, which consists of two parts S_{Angel} and S_{Thrust}. S_{Angel} is the angle of each mechanical leg, and S_{Thrust} is the thrust generated by the water jet on each mechanical leg. The desired motion of the robot can be achieved by properly configuring the state of four legs for the robot. In addition, the error between the desired resultant force and the actual resultant force is also regarded as a state. In order to ensure that the resultant force generated remains stable, a flag state is also regarded as an observation state, which indicates the duration of the resultant force. This yields an 11D action space. The goal of the control layer is $G_{Control}$, which comes from the planning layer. $G_{Control}$ represents the expected sum torque of the robot, which is a two-dimensional vector. It can be calculated by the planning layer according to the desired position and speed. The neural network of the planning layer is represented by a four-layer neural network. The receiving input of the neural network is the state $S_{Control}$ and target $A_{Planning}$ of the robot. The output of the natural network is action $A_{Control}$, which is the configuration parameter of the four mechanical legs. The value function of the planning layer is modeled by a similar network, but there is only one linear unit in the output layer.

According to the requirements of the expected total torque, by reasonably configuring the parameters of each mechanical leg, it can not only realize the expected motion of the amphibious spherical robot, but also protect the mechanical structure, save energy and increase the endurance time. According to this, when the expected total torque of the robot is given by the planning layer, the total reward function of the control layer is defined as:

$$R_{Control} = \omega_f(\boldsymbol{f}_t - \boldsymbol{f}_r)^T(\boldsymbol{f}_t - \boldsymbol{f}_r) \quad (8)$$

where $\boldsymbol{f}_{t-1}$ is the thrust of the water jet motor at the previous moment. The reward function $R_{Control}$ is about the reward function of the thrust of the water jet motor, which is used to reward the error between the expected resultant force and the actual resultant force.

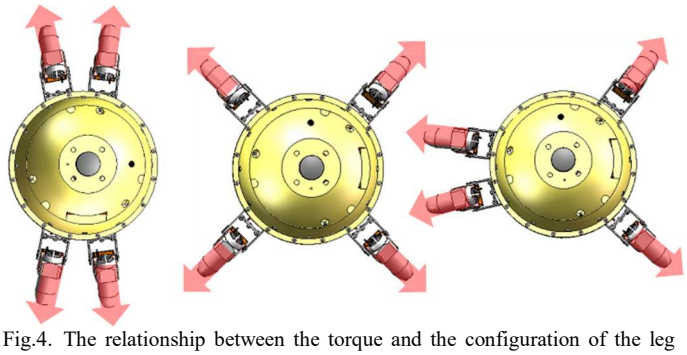

Fig.4. The relationship between the torque and the configuration of the leg parameters.

D. The Upper Planning Layer

The function of the planning layer is to generate corresponding action planning signals according to the required position information. Compared with the control layer, planning does not directly control the robot, but generates the goal of the control layer to control the movement of the robot. Through the cooperation of the planning layer and the control layer, the movement of the amphibious spherical robot can finally be realized. Therefore, the planning layer can be regarded as a transition layer, which converts the input instructions into commands that the robot can execute. Each time, the input of the planning layer includes the desired position and the feedback of the state. The output of the planning layer is the expected resultant force, which is also the input of the control layer.

In order to achieve precise control of the speed and position of the amphibious spherical robot, the feedback state of the planning layer includes position information. Each state $S_{Planning}$ is a vector, which contains position and velocity characteristics. $S_{Position}$ is a two-dimensional coordinate vector, which is the position feedback of the amphibious spherical robot. And S_{Speed} is also a two-dimensional coordinate vector, which represents the speed of the robot. $S_{Acceleration}$ is also a two-dimensional coordinate vector, which represents the acceleration of the robot. In addition, in order to keep the robot stable at the desired position, a flag state is also regarded as an observation state, which represents the maintenance time of the robot at the desired position. Therefore, the state space of the planning layer is a 7D space. The output of the planning layer is a resultant force, which is the goal of the control layer. Due to the simplification of the motion of the amphibious robot, the output of the planning layer is a vector in a two-dimensional plane. The neural network of the planning layer is represented by a three-layer neural network, which regards $S_{Planning}$ and $G_{Planning}$ as inputs, and outputs

$A_{Planning}$. The planning layer value function is modeled by a similar network.

In order to ensure that the amphibious spherical robot can reach the desired position in the water, the reward function is also designed. The reward function of the amphibious spherical robot moving in the water is defined as

$$R_{Planning} = \omega_p R_p \tag{9}$$

where R_p is the position reward. It is used to reward the error between the desired position and the actual position

Through the constraints of the above four reward functions, accurate planning of the motion of the amphibious spherical robot can be realized

IV. Simulation and Results

In this section, the two-layer reinforcement learning motion control scheme for the amphibious spherical robot is verified by simulation. The performance of the lower control layer and the upper planning layer are verified and analyzed respectively.

A. Experimental Setup

In the simulation, OpenAI gym and tensorflow were used to verify the proposed motion control scheme. OpenAI gym is Open source interface to reinforcement learning tasks. In the simulation environment, a simple model based on the shape and parameters of the amphibious spherical robot is built. The basic parameters of the robot in the simulation environment are exactly the same as those of the actual robot. In addition, due to the simplification of the robot movement, the tibial joint and the tibial joint of the amphibious spherical robot are fixed, so the amphibious spherical robot can only move on the surface of the water. Reinforcement learning is trained by a computer with a main frequency of 2.2GHz and a memory of 8Gb.

B. Experimental Result

In the simulation, the control layer is first trained separately. The training process of the control layer is shown in Fig.5. It can be seen from the figure that when the training iteration reaches 300 times, the control layer can already get a higher reward. It implies that the control layer can generate corresponding leg configurations according to the desired resultant force.

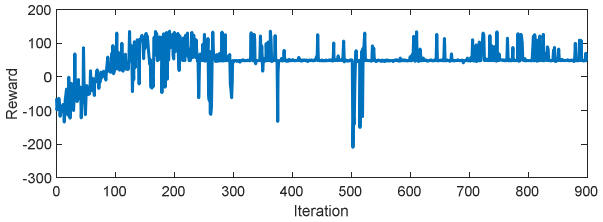

Fig.5. Control layer learning curves.

In order to verify the control effect of the control layer, the control layer is set to track a desired resultant force in the simulation. The resultant force of this desire is defined as

$$\begin{cases} x_t = d\sin(\alpha_t) \\ y_t = d\cos(\alpha_t) \end{cases} \tag{10}$$

where d = 100 is the magnitude of the resultant force, and α is the direction of the resultant force. The error of the resultant force generated by the control layer and the desired resultant force is shown in Fig.6, where Fig (a) is the amplitude error, and Fig (b) is the direction error. It can be seen from the figure that the control layer can generate accurate resultant force by setting the parameters of the legs.

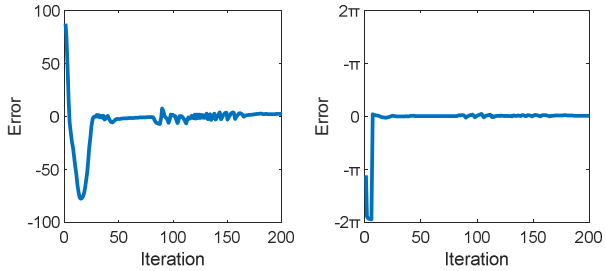

Fig.6. The error between the desired resultant force and the actual resultant force. (a) Amplitude error. (b) Angle error.

In the simulation, the planning layer was also trained 900 times, and the training results are shown in Fig.7. As can be seen from the figure, when the number of training reaches 300 times, the planning layer can already get a higher reward score, which means that the planning layer can generate a reasonable joint effort plan based on the desired location.

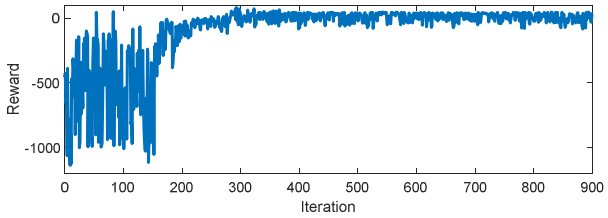

Fig.7. The planning layer learning curves.

The robot is set to track a square trajectory in order to verify the control effect of the planning layer, where the side length of the square trajectory is 100cm. The error of the robot trajectory between the desired trajectory on the x-axis and the y-axis is shown in Fig. 8, where the error of the x-axis in the Fig. 8(a) and the error of the y-axis in the Fig. 8(b). The results show that the robot can still track the desired trajectory steadily despite some fluctuations in the motion trajectory.

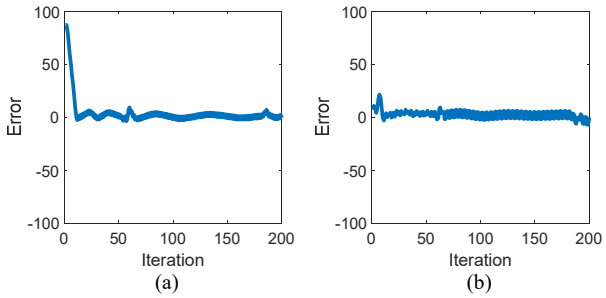

Fig.8. The error between the desired position and the actual position. (a) Error in the x-axis direction. (b) Error in the y-axis direction.

To realize the motion control of the robot, the proposed two-layer reinforcement learning control strategy requires the cooperation of the planning layer and the control layer. In the simulation, the effect of the proposed strategy is verified by letting the robot track a circular trajectory. The circular trajectory can be defined by equation (10), and its radius is 100 cm. The trajectory of the robot is shown in Fig. 9. The error of the robot real trajectory and the desired trajectory is shown in Fig. 10, where Fig. 10(a) is the error of the x-axis, and Fig. 10(b) is the error of the y-axis. It can be seen from the figure that although there are some fluctuations in the motion trajectory, the robot can still basically track the desired trajectory

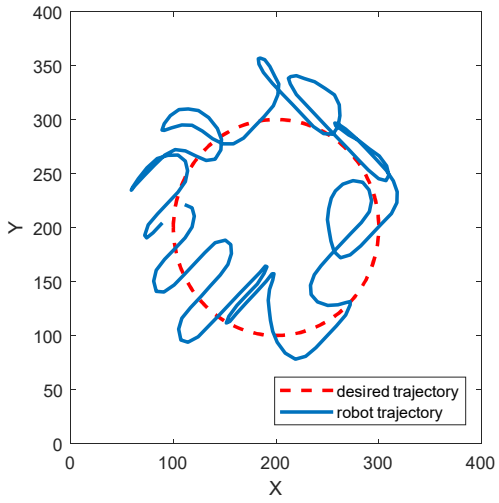

Fig.9. The trajectory of the robot

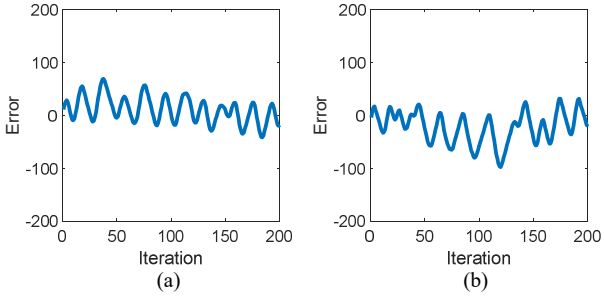

Fig.10. The error between the desired position and the actual position. (a) Error in the x-axis direction. (b) Error in the y-axis direction.

V. Conclusion

A two-layer reinforcement learning scheme for amphibious spherical robots is proposed. Through the planning of the upper layer and the hardware control of the lower layer, the proposed control scheme can realize the desired motion of the amphibious spherical robot on the horizontal plane. The simulation experiment verifies the feasibility and stability of the proposed scheme.

Acknowledgment

This work was partly supported by National High Tech. Research and Development Program of China (No.2015AA043202), the National Natural Science Foundation of China (61773064, 61503028), the National Key Research and Development Program of China (No. 2017YFB1304401).

References

[1] X. Hou, S. Guo, L. Shi, et al. "Improved Model Predictive-Based Underwater Trajectory Tracking Control for the Biomimetic Spherical Robot under Constraints." *Applied Sciences* , vol.10, no.22, pp.8106, doi: 10.3390/app10228106, 2020.

[2] H. Xing, L. Shi, S. Guo, *et al*. "Design, Modeling and Experimental Evaluation of a Legged, Multi-vectored Water-jet Composite Driving Mechanism for an Amphibious Spherical Robot," *Microsystem Technologies*, vol.25, no.8, pp.1-13, doi: 10.1007/s00542-019-04536-7, 2019.

[3] X. Zhao, Q. Zong, R. Zhang, et al. "Application of brain-inspired intelligence technology in unmanned vehicles." *Control Theory & Application* , vol.36, no.1 ,pp.1-12, 2019.

[4] K. Tanakitkorn, P. A. Wilson, S. R. Turnock, et al. "Depth control for an over-actuated, hover-capable autonomous underwater vehicle with experimental verification." *Mechatronics* , vol.41 ,pp.67-81, doi: 10.1016/j.mechatronics.2016.11.006, 2017.

[5] S. Nakhoob, A. Chatraei, K. Shojaei. "Fuzzy Adaptive Control for Trajectory Tracking of Autonomous Underwater Vehicle." *J. Intell. Proced. Electr. Technol* , vol.4, pp.71-77, 2014.

[6] K. Tanakitkorn, P. A. Wilson, S. R. Turnock, et al. "Sliding mode heading control of an overactuated, hover-capable autonomous underwater vehicle with experimental verification." *Journal of Field Robotics* , vol.35, no.3, pp. 396-415, doi: 10.1002/rob.21766, 2018.

[7] G. V. Lakhekar, L. M. Waghmare. "Robust maneuvering of autonomous underwater vehicle: an adaptive fuzzy PI sliding mode control." *Intelligent Service Robotics* , vol.10, no.3, pp.195-212, doi: 10.1007/s11370-017-0220-2, 2017.

[8] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, et al. "Optimal and Autonomous Control Using Reinforcement Learning: A Survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042-2062, doi: 10.1109/TNNLS.2017.2773458, 2018.

[9] B. Fernandez-Gauna, J. L. Osa, M. Graña. "Effect of Initial Conditioning of Reinforcement Learning Agents on Feedback Control Tasks over Continuous State and Action Spaces." *Springer International Publishing* ,doi: 10.1007/978-3-319-07995-0_13, 2014.

[10] P. Walters, R. Kamalapurkar, F. Voight, et al. "Online Approximate Optimal Station Keeping of a Marine Craft in the Presence of an Irrotational Current," in *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 486-496, doi: 10.1109/TRO.2018.2791600, 2018.

[11] I. Carlucho, M. De Paula, S. Wang, et al. "Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning." *Robotics and Autonomous Systems*, vol. 107, pp.71-86, doi: 10.1016/j.robot.2018.05.016, 2018.

[12] K. B. Knudsen, M. C. Nielsen and I. Schjølberg, "Deep Learning for Station Keeping of AUVs," *OCEANS 2019 MTS/IEEE SEATTLE*, pp. 1-6, doi: 10.23919/OCEANS40490.2019.8962598, 2019.

[13] H. Xing, L. Shi, X. Hou, *et al*, "Design, Modeling and Control of a Miniature Bio-inspired Amphibious Spherical Robot", Mechatronics, in press, 2021

[14] X. Hou, S. Guo, L. Shi, et al. "Hydrodynamic Analysis-Based Modeling and Experimental Verification of a New Water-Jet Thruster for an Amphibious Spherical Robot," *Sensors*, vol.19, no.2, pp.259, doi:10.3390/s19020259, 2019.

[15] J. Guo, C. Li, S. Guo, "Path Optimization Method for the Spherical Underwater Robot in Unknown Environment", *Journal of Bionic Engineering*, vol.17, no.5, pp.944-958, doi: 10.1007/s42235-020-0079-3, 2020.

[16] H. Xing, L. Shi, S. Guo, et al. "Robust RGB-D Camera and IMU Fusion-based Cooperative and Relative Close-Range Localization for Multiple Turtle-Inspired Amphibious Spherical Robot," *Journal of Bionic Engineering*, vol.16, no.3, pp.442-454, doi: 10.1007/s42235-019-0036-1, 2019.

[17] L. Shi, Y. Hu, S. Su, S. Guo, *et al*., "A Fuzzy PID Algorithm for a Novel Miniature Spherical Robots with Three-dimensional Underwater Motion Control", Journal of Bionic Engineering, vol.17, no.5, pp.959-969, doi: 10.1007/s42235-020-0087-3, 2020.

[18] V. Mnih, K. Kavukcuoglu, D. Silver,, et al. "Human-level control through deep reinforcement learning." *nature* , vol.518, no.7540, pp.529-533,2015.