Journal of Intelligent & Robotic Systems (2019) 96:591–601
https://doi.org/10.1007/s10846-019-01004-2

Mapless Motion Planning System for an Autonomous Underwater Vehicle Using Policy Gradient-based Deep Reinforcement Learning

Yushan Sun[1] · **Junhan Cheng**[1] · **Guocheng Zhang**[1] · **Hao Xu**[1]

Received: 29 August 2018 / Accepted: 21 February 2019 / Published online: 5 March 2019

Abstract

This research is concerned with the motion planning problem encountered by underactuated autonomous underwater vehicles (AUVs) in a mapless environment. A motion planning system based on deep reinforcement learning is proposed. This system, which directly optimizes the policy, is an end-to-end motion planning system. It uses sensor information as input and continuous surge force and yaw moment as output. It can reach multiple target points in a sequence while simultaneously avoiding obstacles. In addition, this study proposes a reward curriculum training method to solve the problem in which the number of samples required for random exploration increases exponentially with the number of steps needed to obtain a reward. At the same time, the negative impact of intermediate rewards can be avoided. The proposed system demonstrates good planning ability for a mapless environment and excellent ability to migrate to other unknown environments. The system also has resistance to current disturbances. The simulation results show that the proposed mapless motion planning system can guide an underactuated AUV in navigating to its desired targets without colliding with any obstacles.

Keywords Autonomous underwater vehicle (AUV) · Motion planning · Deep reinforcement learning · Curriculum learning

1 Introduction

Autonomous underwater vehicles (AUV) have high application value for the measurement of hydrological information. The ability to plan their movements is an important feature of AUVs when conducting measurement of hydrological information task. For underwater robots, motion planning can be based on the realization of an obstacle map or it can be directly performed in a mapless environment. The goal is to enable a robot to reach a target while avoiding obstacles (e.g., rocks and boats) at a certain depth. Motion planning in a mapless environment is universally important. Considerable literature has investigated the motion planning problem in a mapless environment. The most effective methods for addressing this problem are those based on reinforcement learning and deep learning.

✉ Guocheng Zhang
zhang_china2018@163.com

Junhan Cheng
cheng_junhan@outlook.com

[1] Science and Technology on Underwater Vehicle Laboratory, Harbin Engineering University, Harbin 150001, China

The reinforcement learning method is an artificial intelligence (AI) method. A robot can evaluate the value of status action through continuous trial and error by using the environment state as an input and output policy. Providing label-independent training or building a map in advance to obtain an optimal sports planning policy is unnecessary. In the underwater field, considerable literature has studied motion planning based on reinforcement learning as follows. Kawano et al. [15] proposed a motion planning method that enables an underactuated AUV to avoid obstacles and reach a target in an environment with strong water flow. Carreras et al. [2, 3] presented a behavior control architecture that realizes the target tracking of AUVs. Andres et al. [8, 9] developed a control method based on policy gradient reinforcement learning and achieved AUV trajectory tracking. Kormushev et al. [16] presented a method to improve underwater robot control based on Q learning of periodic signal. Reinforcement learning has also been investigated in other fields of motion planning. The following works provide methods based on reinforcement learning. Lei et al. [17] proposed a motion planning method that realizes obstacle avoidance among mobile robots. Chris et al. [39] presented a motion planning method that achieves manipulator motion with seven degrees of freedom (DOFs). Andrew et al. [24] developed a motion planning

method to realize helicopter hovering. However, motion planning methods based on reinforcement learning exhibit certain adaptive problems when migrating to other unknown environments.

The deep learning method is an AI method that can realize motion planning without limiting the scene. The following are methods based on deep learning. Muller et al. [23] proposed a motion planning method that directly maps monocular vision images onto angular deviation and realizes obstacle avoidance among off-road robots. Chen et al. [4] presented a motion planning method that extracts image information features and realizes autonomous driving through these features. Pfeiffer et al. [26] developed a motion planning method based on a convolutional neural network (CNN) that directly maps image information onto an action to realize obstacle avoidance among mobile robots. However, motion planning methods based on deep learning require a huge amount of actual planning data for training, which are difficult to use extensively in an underwater environment.

In addition to the above artificial intelligence methods, the sample-based path planning method can achieve mapless motion planning. In recent years, in the underwater field, Cui et al. [18, 38] have studied the motion planning algorithm based on sampling. However, the sample-based path planning method is difficult to establish complete constraints.

Overall, mapless motion planning remains a challenging problem. Deep learning and reinforcement learning exhibit potential in solving complex problems. Deep reinforcement learning methods, which combine the two techniques, have elicited considerable attention because of the creation of AlphaGo [31]. In recent years, deep reinforcement learning has been widely applied and studied in the field of AI [21, 22, 28, 35, 36]. Deep reinforcement learning also exhibits potential in the field of robotics. Furthermore, it does not require raw data and achieves good adaptability when migrating to other unknown environments. Zhang et al. [41] proposed a motion planning method based on deep reinforcement learning to achieve a robotic manipulator that can reach a target using extracted image features. Gu et al. [12] presented a motion planning method based on deep reinforcement learning that can realize the opening operation of a manipulator. Lei et al. [32, 33] developed a motion planning method based on deep reinforcement learning that enables a mobile robot to avoid obstacles and reach its target. The application of deep reinforcement learning in the field of underwater robots requires further verification. Cui et al. [6] proposed a motion planning method based on deep reinforcement learning that maps position deviation and velocity deviation onto thrust and achieves the trajectory tracking of fully driven underwater robots; however, this method does not consider obstacle avoidance. Cheng et al. [5] presented a motion planning method based on deep reinforcement learning that uses a CNN to extract features of sensor information and makes decisions to avoid obstacles and reach a target.

However, there is a problem in the reward shaping of deep reinforcement learning. It is difficult for designers to design reward functions to ensure that the movement of artificial intelligence is exactly the same as the designer imagined. Even if the reward function is designed for AUV motion planning algorithms, AUV may not be able to reach the target, or more generally, the trajectory of reaching the target is unexpected to the designer. The abuse of intermediate rewards is not an ideal way to train deep reinforcement learning models.

In the current study, a differential AUV model is constructed, and a motion planning system based on deep reinforcement learning is realized. The work can be summarized as follows.

(1) We developed an end-to-end motion planning system for underwater robots based on deep reinforcement learning, which adopts the proximal policy optimization (PPO) algorithms [14, 29, 30]. Sensor data are mapped directly onto the continuous thrust required by the underwater robot. The motion planning system does not need an obstacle map of the unknown environment, to complete the motion planning tasks of avoiding obstacles and reaching multiple target points in a certain order. Furthermore, the system does not require a pre-database for training.
(2) A reward curriculum training method is proposed to solve the problem that intermediate rewards may have a negative impact on training. At the same time, the method can also solve the problem in which the environment provides only sparse rewards. According to the progress of the training, gradually improve the difficulty of the agent to get the intermediate rewards. Different reward curriculums were designed to explore the impact of curriculum design on model training.
(3) The effectiveness of the proposed motion planning system based on deep reinforcement learning is corroborated. The trained model is also migrated to other unknown environments and current environments for simulations. The simulations achieve satisfying results and verify the adaptability of the system to unknown environments and the resistance to current disturbances.

2 Preliminaries

2.1 AUV and AUV Maneuvering Models

A differential AUV model is constructed. The length of the AUV model is $1.46m$, its mass is $45kg$, and the center of

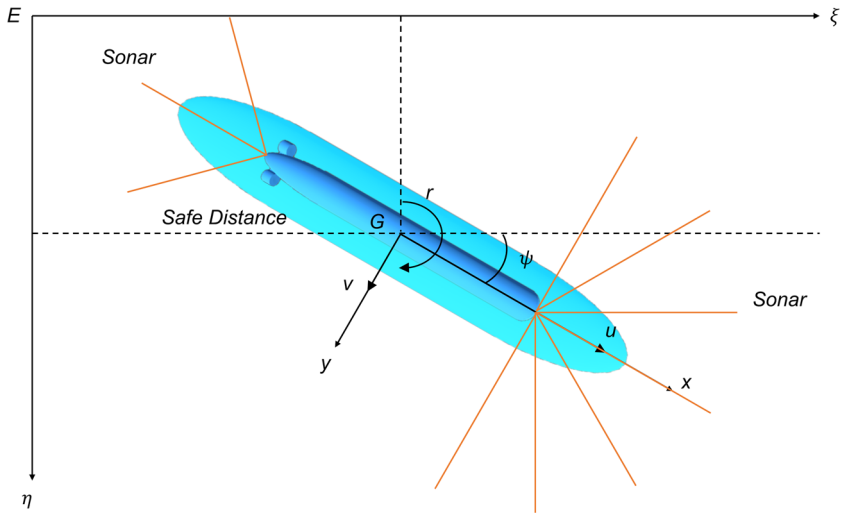


Fig. 1 Major components of the AUV

gravity coordinates are (0, 0) in the body-fixed frame. The AUV has ten range-finding sonars. These sonars can obtain obstacle information around the AUV. The arrangement is shown in Fig. 1. The sampling frequency of the range-finding sonars is $2Hz$, and their detection distance is $20m$. The stern of the robot has two propellers, with $0.2m$ from the y-axis in the body-fixed frame. Each propeller can generate $10kg$ of force. The AUV has an inertial navigation system for measuring velocity, position, and attitude.

We consider the horizontal motion of the AUV, which is described by the motion components as surge, sway, and yaw. On the basis of this consideration, $v = [u, v, r]^T \in \mathbb{R}^3$ denotes the velocity vector, whereas $\eta = [x, y, \psi]^T \in \mathbb{R}^3$ denotes the position vector.(ψ) is the heading of the AUV, and (*x*,*y*) is the position in the earth-fixed inertial frame. The linear velocities $v = [u, v, r]^T \in \mathbb{R}^3$ correspond to surge, sway, and yaw, respectively, in the body-fixed frame of the AUV. Figure 1 illustrates the major concepts of the movement process in this case. The horizontal maneuvering models [10] of the AUV can be expressed as

$$\dot{\eta} = R(\psi)v, \tag{1}$$

$$M\dot{v} + D(v)v + C(v)v + g(\eta) = \tau, \tag{2}$$

where $R(\psi)$ is the rotation matrix for the horizontal motion of the AUV with three DOFs, which can be expressed as

$$R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

In (2), $M = M_A + M_{RB}$ is the combination of the added mass matrix M_A and the rigid body matrix M_{RB}:

$$M_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -Y_{\dot{r}} & -N_{\dot{r}} \end{bmatrix}, \tag{4}$$

$$M_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix}. \tag{5}$$

Moreover, $C(v) = C_A(v) + C_{RB}(v)$ denotes the Coriolis and centripetal terms, where

$$C_A(v) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u & 0 \end{bmatrix}, \tag{6}$$

$$C_{RB}(v) = \begin{bmatrix} 0 & 0 & -mv \\ 0 & 0 & mu \\ mv & -mu & 0 \end{bmatrix}. \tag{7}$$

Similarly, $D(v) = D + D_n(v)$ denotes the hydrodynamic damping matrix, which can be expressed as

$$D = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix}, \tag{8}$$

$$D_n(v) = -\begin{bmatrix} X_{|u|u}|u| & 0 & 0 \\ 0 & Y_{|v|v}|v| + Y_{|r|v}|r| & Y_{|v|r}|v| + Y_{|r|r}|r| \\ 0 & N_{|v|v}|v| + N_{|r|v}|r| & N_{|v|r}|v| + N_{|r|r}|r| \end{bmatrix}. \tag{9}$$

The coefficients $\{X_{(\cdot)}, Y_{(\cdot)}, Z_{(\cdot)}\}$ are the so-called hydrodynamic derivatives that represent the hydrodynamic forces and moments acting on the AUV. $g(\eta) \in \mathbb{R}^3$ indicates the unmodeled dynamics. The propeller can produce only surge force and yaw moment due to the use of a differential AUV model. Therefore, $\tau = \{\tau_u, 0, \tau_r\}^T \in \mathbb{R}^3$.

2.2 Problem Definition

The primary task of AUV motion planning is to avoid obstacles while reaching the target. In a mapless environment,

a trajectory cannot be planned in advance. Only the sensor information can be used to understand the environment and its own state, and then the planning policy is outputted, which has high real-time requirements. The proposed end-to-end system can realize motion planning in an unknown environment. Sensor information is directly mapped onto the output of the force, such that

$$\tau_t = f(s_t), \tag{10}$$

$$s_t = (x_t, p_t, v_{t-1}), \tag{11}$$

where s_t denotes the sensor information of the AUV at step t, and τ_t denotes the surge force and yaw moment required by the AUV at step t. Moreover, s_t consists of three parts: x_t denotes the obstacle distance information obtained by a range-finding sonar, p_t denotes the coordinates of the target in the body-fixed frame at step t, and $v_{t-1} = [u, r] \in \mathbb{R}^2$ denotes the velocity in the body-fixed frame at step $t-1$. In addition, the motion planning system is derived under the following assumptions.

Assumption The motion planning task is horizontal and has three DOFs.

Assumption To meet the real-time requirements, the planning system can output regularly at an interval of 0.5 s. That is, each step interval is $0.5s$.

2.3 Training and Test Environments

We construct an underwater training environment, as shown in Fig. 2. The training environment has an area of $360 \times 180\ m^2$. In the figure, red bars denote obstacles, green dots denote targets, and the blue line denotes the AUV. The initial position of the AUV is at the center of the map toward the right of the area. The target point will randomly appear outside the obstacle during each training round. The target is a circular region that radius is $3\ m$. A sonar can detect red obstacles but not green targets. The red surrounding area is the boundary of the training environment, which can also be detected by the sonars.

The simulation environment uses the real-time collision detection system, which is tested every $0.02\ s$. We build a capsule shell for the robot model, as shown in Fig. 1. A shell overlapping with an obstacle indicates a collision. The real-time collision detection system is also used when the robot arrives at the target, but the safety shell of the AUV is not used. The target is reached only when the AUV body overlaps with the target area.

To demonstrate the adaptability of the model trained according to the method in the unknown environment, we construct two virtual unknown environments to test the planning system, namely, unknown environments with a single target and multiple targets, as shown in Fig. 3.

Single-target environment is to test the effect of the model in the longer trajectory planning, while multi-target environment is to test the effect of the model in different circumstances. The test environment has an area of $360 \times 180\ m^2$ that contains multiple obstacles. In the single target environment, the AUV starts at the upper right and the target is located at the lower left. The multi-target environment contains ten targets, and the planning system should guide the robot to the target locations by following a sequence of numbers. The robot is initially located at Position 0. In the test environment, the same collision detection system and

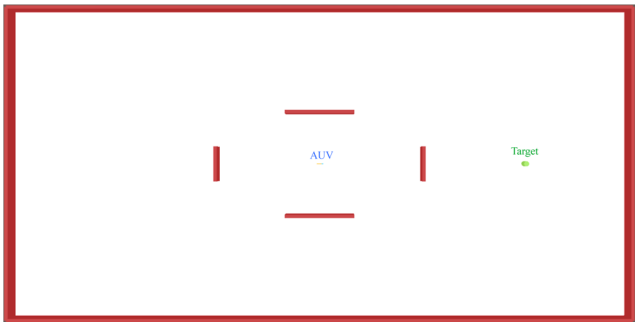
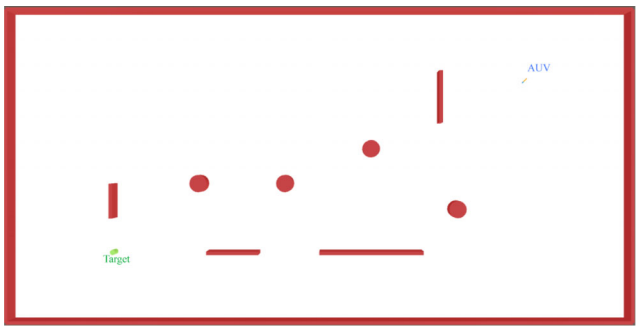
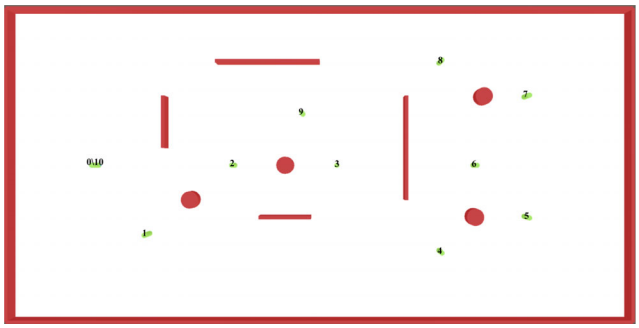


Fig. 2 Training environment



(a) Single-target unknown environment



(b) Multi-target unknown environment

Fig. 3 Unknown environment

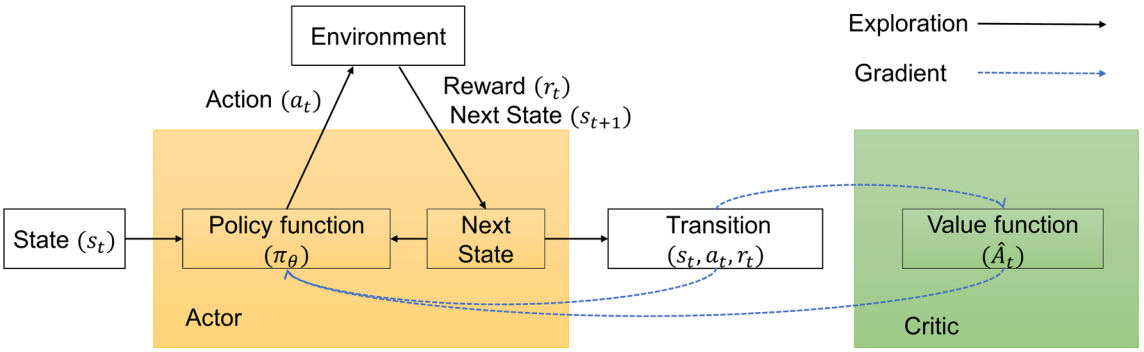


Fig. 4 Actor–critic framework

detection conditions used in the training environment are adopted. For the deep reinforcement learning system, all the test environment maps are used only for the visualization of the planning trajectory; the AUV does not know the environment map.

3 Motion Planning System

3.1 System Framework and Network Structure

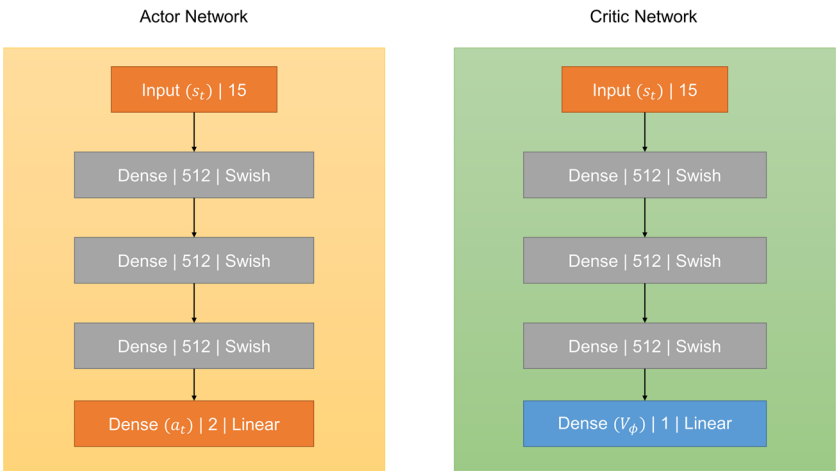
This study uses a policy gradient-based reinforcement learning method. In contrast with the traditional value-based reinforcement learning, this method can directly search for policy. Therefore, it can be applied to continuous and high-dimensional action spaces. The actor–critic framework can effectively solve the problem of reinforcement learning based on policy gradient [19, 20, 37]. Hence, we use this framework to implement reinforcement learning.

The actor–critic framework aims to jointly estimate policy function $\pi_\theta(a_t|s_t)$ and value function $V_\phi(s)$. From the current state s_t, the agent explores the environment by sampling the policy function $\pi_\theta(a_t|s_t)$ and receives positive/negative reward r_t until the terminal state or a maximum number of steps are reached. The exploration provides a trajectory $\{(s_t, a_t, r_t), (s_{t+1}, a_{t+1}, r_{t+1}), \cdots\}$, from which the policy function and value function are updated. The framework is shown in Fig. 4.

Two neural networks are used to describe the policy function $\pi_\theta(a_t|s_t)$ and value function $V_\phi(s)$. As shown in Fig. 5, the actor network has three fully connected hidden layers, with each layer containing 512 nodes. We use the $Swish$ function [27] as the activation function between the hidden layers. The input is a 15-dimensional vector that denotes s_t, including 10-dimensional sonar information, 2-dimensional target information, and 3-dimensional velocity information. The output is a 2-dimensional vector that consists of surge force and yaw moment. We clip the ranges of surge force in $(-0.2, 1)$ and yaw moment in $(-1, 1)$. $Linear$ activation functions are used between the hidden and output layers.

The critic network also has three fully connected hidden layers, and each hidden layer contains 512 nodes. The $Swish$ function is used as the activation function between

Fig. 5 Network structure



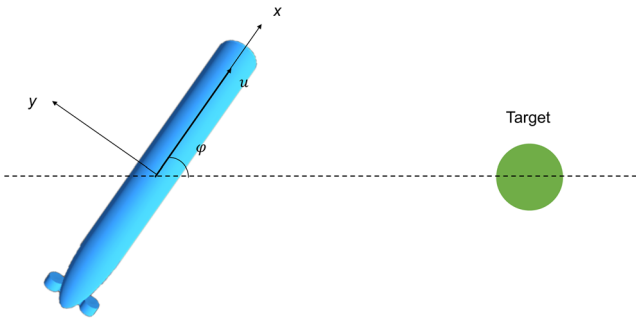


Fig. 6 Intermediate reward

the hidden layers. *Linear* activation functions are used between the hidden and output layers. The input is an 15-dimensional vector that denotes s_t. The output is $V_\phi(s_t)$. The forward propagation process of a neural network can be expressed as

$$h_i^k = activation^k\left(\sum_{i=1}^{n^{k-1}} x_i^k w_i^k + b^k\right), \tag{12}$$

where x^k denotes the input of any layer k of the neural network; h^k denotes the output of layer k; w^k and b^k denote the weight and bias of the neural network, respectively; $activation^k$ denotes the activation function of layer k; and n^k denotes the nodes of layer k. The input layer is layer 0. The time complexity of the planning algorithm can be expressed as

$$Time \sim O\left(n^{k-1} \times n^k \times n^{k+1}\right). \tag{13}$$

3.2 Reward Shaping

Reward shaping is an effective technique that can be applied to reinforcement learning in a complicated environment with sparse reward [7, 25]. In this study, an intermediate reward is also used in addition to a positive reward for reaching the target and a negative reward for colliding with obstacles. The reward function can be expressed as

$$r(s_t, a_t) = \begin{cases} r_a & \text{if arrive} \\ r_b & \text{if collide} \\ r_c u_t \cos\varphi_t & \text{every step} \\ r_d & \text{every step} \end{cases} \tag{14}$$

If the collision detection system perceives that the AUV reaches the target, then the model will receive a positive reward r_a; if the AUV collides with an obstacle, then it will receive a negative reward r_b. Both situations will end the current training iteration. Given the weak steering ability of the AUV, the planning system is required to turn numerous steps in advance to successfully reach the target point or avoid obstacles, which is extremely difficult in reinforcement learning. To solve this problem, we set the intermediate reward $r_c u_t \cos\varphi_t$, as shown in Fig. 6, where r_c is the intermediate reward parameter. The AUV receives an intermediate reward for each step, thereby encouraging it to maintain its speed toward the target point and helping it to reach the target point. The AUV receives a negative reward r_d for each step, thereby encouraging AUV to reach the target point faster.

The model of intermediate reward training (Model 2) is compared with that of non-intermediate reward training (Model 1). Both use PPO algorithms and the same system framework. The change in the rewards per episode of the two models with the training step is shown in Fig. 7. The rewards per episode of the Model 1 is nearly unchanged. Thus, the planning system does not learn any effective policy. The rewards per episode of the Model 2 increases rapidly. Therefore, intermediate rewards help the planning system learn effective policies.

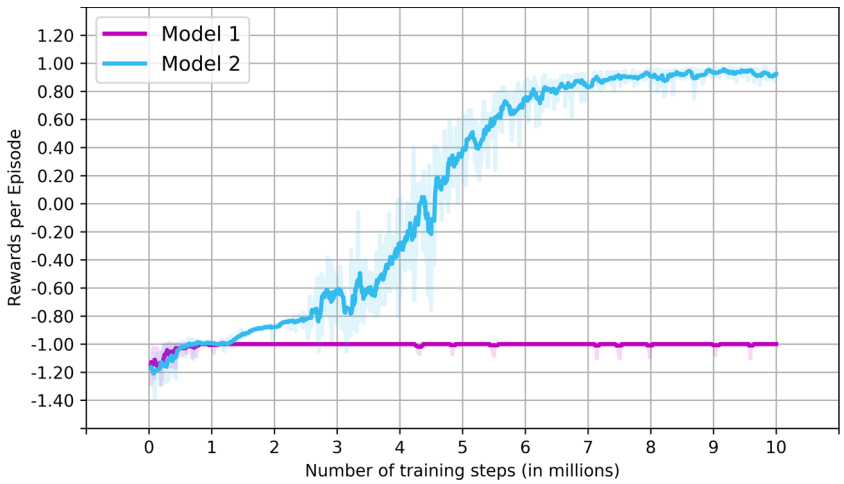


Fig. 7 Rewards per episode of using (Model 2) and not using (Model 1) the intermediate reward model change with the number of training steps

Table 1 Curriculum design

	Curriculum	Class 1	Class 2	Class 3
Model 2	Condition	All times		
	r_c	1		
Model 3	Condition	$R_{pe} \leq 0$	$0 < R_{pe} \leq 0.6$	$R_{pe} > 0.6$
	r_c	1	0.5	0
Model 4	Condition	$R_{pe} \leq 0$	$R_{pe} > 0$	$r_c \leq 0$
	r_c	1	$1 - steps/2 \times 10^6$	0

3.3 Reward Curriculum Learning

The concept of curriculum learning is to change the training rules during the training process, such as training an AI to play the game model, which can gradually improve the difficulty of the game. Curriculum learning was proposed by Bengio [1]; it was originally a technique applied to deep learning training. Zaremba and Graves conducted further studies on this method [11, 40]. Tambet et al. [34] studied the application of curriculum learning to reinforcement learning and proposed a training method to change training difficulty according to training iteration.

The current study proposes a reward curriculum learning training method that combines reward shaping and curriculum learning to improve the existing curriculum learning method. Different reward functions are designed according to the curriculum to gradually improve the difficulty of obtaining rewards for the AUV.

Reward curriculum learning can be described as follows. In the beginning of the training, additional intermediate rewards can be given to help the AUV find its target, to solve problem that the rewards that the model can obtain are sparse. When the robot becomes capable of finding its target, the intermediate reward is gradually reduced, to reduce the negative impact of intermediate rewards on model training.

This study designs two reward curriculum learning schemes to study the influence of curriculum design on the training model. The changes in reward parameter with the iteration are shown in Table 1. R_{pe} in the table denotes rewards per episode. The reward parameter of Model 3 changes step by step with an increase in the rewards per episode, whereas Model 4 exhibits a linear relationship between the reward parameter and the number of training iterations. Model 2 does not use curriculum learning, its reward parameters do not change.

4 Simulations

4.1 Model Training

In this study, $t_{max} = 500(steps)$, the learning rate is 10^{-5}, and the total training step is $10^7 steps$. All the models are trained and tested using TensorFlow. We trained the model from a single Nvidia GeForce GTX 1080 GPU which took over 40 hours.

The rewards per episode with the training step of the four models are shown in Fig. 8. All the four models can converge to a high reward. However, the reward growth rate of Model 4 is relatively slow, which may be attributed to the negative impact of the constantly changing rewards

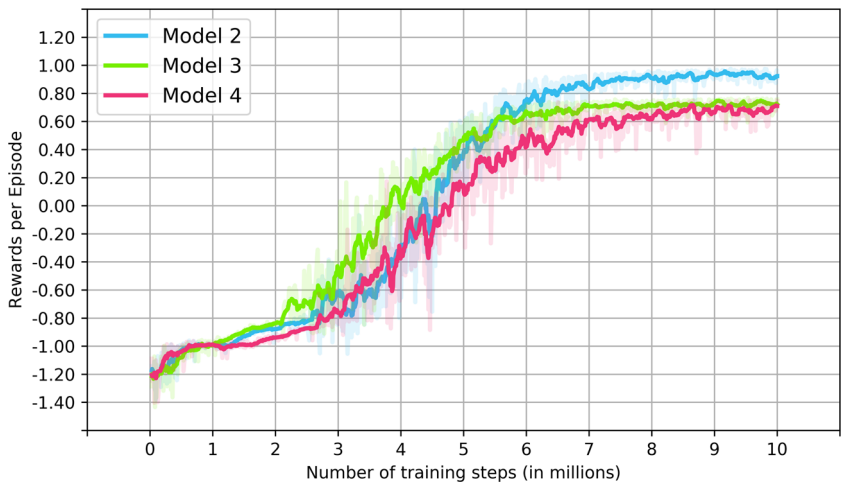


Fig. 8 Rewards per episode of Models 2, 3, and 4 with training steps

Table 2 Performance evaluation in the training environment over 1000 episodes

Metric	Trajectory distance	Time	Collision frequency
Model 2	99.61m	66.67s	< 0.01
Model 3	97.43m	64.01s	< 0.01
Model 4	100.91m	70.23s	< 0.01

Table 3 Performance evaluation in the single-target environment over 1000 episodes

Metric	Trajectory distance	Time	Collision frequency
Model 2	272.73m	160.25s	< 0.001
Model 3	275.48m	163.48s	< 0.001
Model 4	273.70m	173.33s	< 0.001

on training. Models 2 and 3 have a similar convergence speed, thereby indicating that the step by step curriculum learning has no negative influence on the training process. The reward curriculum helps the planning system train at a similar rate.

To quantitatively describe the planning effects of the three models in the training environment, three indicators are used to evaluate different models: (1) planning time (the time required for the robot to reach target), (2) trajectory distance (the distance that the AUV travels when it reaches all the targets), and (3) collision frequency (the ratio of the number of collisions in the test to the total number of tests). We did thousands of simulations on each model and averaged, and the results as shown in Table 2.

As can be seen from the table, all three models can successfully complete the task of motion planning, and the planning data of Model 2 and 3 are not far from each other. The data of Model 4 is inferior to the other two models, which is consistent with the rewards per episode data shown in the Fig. 8. The training results show that the step-by-step curriculum learning training model can replace the model that only uses the intermediate reward training, which can also solve the problem of sparse reward.

4.2 Unknown Environment Simulations

This study constructs two kinds of unknown environments, namely, single-target and multi-target unknown environments, as shown in Fig. 3. The initial position of the AUV in a single-target environment is far from the target point. This environment is mainly used to verify the planning ability of the model to reach a long-distance target point. There are ten target points in a multi-target environment. This environment is used to verify the planning ability of the model in various situations.

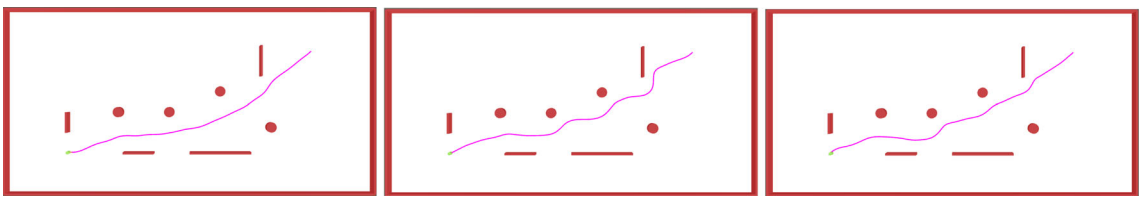
First, the three models are tested in the single-target unknown environment. To qualitatively describe their planning performance, their paths are shown in Fig. 9. All three models can complete the motion planning task of simultaneously avoiding obstacles and reaching a remote single target.

The same indicator is used to evaluate the single-target unknown environment planning effect of different models, as shown in Table 3.

Due to the loose arrangement of obstacles in the single-target environment, the three models did not collide with obstacles in simulations. Compared with Model 2 and 3, Model 4 requires more planning time. In other respects, the three models differ little.

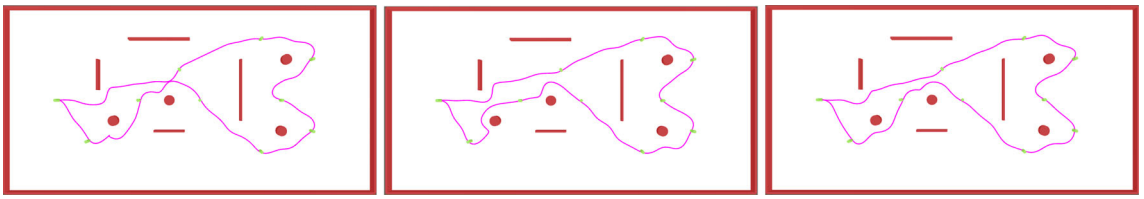
Then, the three models are tested in the multi-target unknown environment. To describe their planning performance, Fig. 10 shows their planned trajectory in the multi-target environment. All three models can complete motion planning tasks of simultaneously avoiding obstacles and reaching multiple targets in sequence.

The same indicator is used to evaluate the multi-target unknown environment planning effect of different models, as shown in Table 4.



(a) Model 2 (b) Model 3 (c) Model 4

Fig. 9 Trajectory in the single-target unknown environment



(a) Model 2 (b) Model 3 (c) Model 4

Fig. 10 Trajectory in the multi-target unknown environment

Similarly, none of the three models collided with obstacles in simulations. Model 3 performs best in multi-target environment, and all evaluation indicator is the best. The simulation results show that our reward curriculum learning training method can improve the adaptability of the algorithm model in various situations. However, the reward parameter should be reduced step by step with the number of training steps, rather than decreasing linearly with the number of training steps.

4.3 Simulations in Current Field

Autonomous underwater vehicles (AUVs) must frequently operate in current fields. Therefore, we have carried out the motion planning simulation under the current disturbance. At any step t the robot has an actual velocity v_{it} given by:

$$v_{it} = v_c + v_t, \tag{15}$$

where v_c denotes the current velocity.

We have added currents in three environments and tested three models. The Current field cover the entire environment, facing the bottom of the map, the current velocity is constant at $0.25m$. The same indicator is used to evaluate the planning effect under the influence of current of different models, as shown in Table 5.

Under the influence of the current, the planning effects of the three models have all declined. The most obvious decline comes from the collision frequency, and all three models have multiple collisions due to the influence of currents. There is also a significant increase in trajectory distance and planning time. The best performer among the three models is Model 3. Model 3 has the shortest trajectory distance and the shortest planning time, which means it has better ability to resist current disturbances. Model 2, which does not use the reward curriculum, does not perform well in the presence of current disturbances. The simulation results show that the proposed reward curriculum learning can alleviate the negative impact of the intermediate reward on training.

However, Model 4 uses the reward curriculum to train, but the effect is not good under the current interference. This is most likely due to the negative impact of changing reward parameters. The influence of different reward curriculums on the training model requires further research.

Table 4 Performance evaluation in the multi-target environment over 1000 episodes

Metric	Trajectory distance	Time	Collision frequency
Model 2	830.17m	539.76s	< 0.001
Model 3	768.24m	488.96s	< 0.001
Model 4	916.37m	615.65s	< 0.001

4.4 Analysis of Simulations

The simulation results show that the motion planning system based on deep reinforcement learning is efficient and can be directly migrated to other mapless environments. In the current environment, the simulation results show that the model 3 trained using the step by step reward curriculum has a better planning effect. The proposed reward curriculum learning can alleviate the negative impact of intermediate rewards on training. The difference between Models 3 and 4 in the planning trajectory for the unknown environment shows that the design of the reward curriculum has an impact on the training of the motion planning system. The design of the reward curriculum is still valuable for further study. For example, the curriculum can be adjusted according to the rewards of different states. If the value of a certain state is low, then the training of this part is strengthened; if the value of a certain state is high, then the training of this part is reduced.

However, the trajectory output of the motion planning system based on deep reinforcement learning is not a global optimal path because planning uses no map as input. A possible explanation is that the network has neither the memory of the previous observation nor long-term prediction ability. RNN and long short-term memory [13, 36] are possible solutions to this problem and will be studied in the future. We set this revision as a future work.

Table 5 Performance evaluation under the interference of currents over 1000 episodes

	Training environment			Single-target environment			Multi-Target environment		
	Trajectory distance	Time	Collision frequency	Trajectory distance	Time	Collision frequency	Trajectory distance	Time	Collision frequency
Model 2	159.07m	108.02s	< 0.1	466.84m	285.92s	< 0.1	1447.50m	979.41s	< 0.1
Model 3	123.79m	83.39s	< 0.1	376.01m	236.16s	< 0.1	1202.42m	806.47s	< 0.1
Model 4	161.72m	118.02s	< 0.1	458.79m	343.82s	< 0.1	1350.05m	998.22s	< 0.1

5 Conclusion

An AUV motion planning system based on deep reinforcement learning is proposed for mapless environments. The system realizes motion planning through the policy by using sensor information as input and surge force and yaw moment as output. It is an end-to-end planning system that uses the PPO algorithm to optimize policies. In addition, this study proposes a training method of reward curriculum learning. This method solves the problem of sparse rewards in complex environments while avoiding the negative impact of intermediate rewards on training. The effectiveness of the system is demonstrated via simulations.

References

1. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th annual international conference on machine learning, pp. 41–48. ACM (2009)
2. Carreras, M., Batlle, J., Ridao, P.: Hybrid coordination of reinforcement learning-based behaviors for auv control. In: 2001 IEEE/RSJ international conference on intelligent robots and systems, 2001. Proceedings, vol. 3, pp. 1410–1415. IEEE (2001)
3. Carreras Pérez, M., Yuh, J., Batlle i Grabulosa, J., Ridao Rodríguez, P.: A behavior-based scheme using reinforcement learning for autonomous underwater vehicles. © Oceanic Engineering **30**, 416–427 (2005)
4. Chen, C., Seff, A., Kornhauser, A., Xiao, J.: Deepdriving: Learning affordance for direct perception in autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2722–2730 (2015)
5. Cheng, Y., Zhang, W.: Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. Neurocomputing **272**, 63–73 (2018)
6. Cui, R., Yang, C., Li, Y., Sharma, S.: Adaptive neural network control of auvs with control input nonlinearities using reinforcement learning. IEEE Trans. Syst. Man Cybern. Syst. Hum. **47**(6), 1019–1029 (2017)
7. Devlin, S., Kudenko, D., Grześ, M.: An empirical study of potential-based reward shaping and advice in complex, multi-agent systems. Adv. Complex Syst. **14**(02), 251–278 (2011)
8. El-Fakdi, A., Carreras, M.: Policy gradient based reinforcement learning for real autonomous underwater cable tracking. In: IEEE/RSJ international conference on intelligent robots and systems, 2008, IROS 2008. pp. 3635–3640. IEEE (2008)
9. El-Fakdi, A., Carreras, M.: Two-step gradient-based reinforcement learning for underwater robotics behavior learning. Robot. Auton. Syst. **61**(3), 271–282 (2013)
10. Fossen, T.I.: Handbook of marine craft hydrodynamics and motion control. John Wiley & Sons (2011)
11. Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S.G., Grefenstette, E., Ramalho, T., Agapiou, J., et al.: Hybrid computing using a neural network with dynamic external memory. Nature **538**(7626), 471 (2016)
12. Gu, S., Holly, E., Lillicrap, T., Levine, S.: Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In: 2017 IEEE international conference on robotics and automation (ICRA), pp. 3389–3396. IEEE (2017)
13. Heess, N., Hunt, J.J., Lillicrap, T.P., Silver, D.: Memory-based control with recurrent neural networks. arXiv:1512.04455 (2015)
14. Heess, N., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, A., Riedmiller, M., et al.: Emergence of locomotion behaviours in rich environments. arXiv:1707.02286 (2017)
15. Kawano, H., Ura, T.: Motion planning algorithm for nonholonomic autonomous underwater vehicle in disturbance using reinforcement learning and teaching method. In: IEEE international conference on robotics and automation, 2002. Proceedings. ICRA'02, vol. 4, pp. 4032–4038. IEEE (2002)
16. Kormushev, P., Caldwell, D.G.: Towards improved auv control through learning of periodic signals. In: Oceans-San Diego, 2013, pp. 1–4. IEEE (2013)
17. Lei, T., Ming, L.: A robot exploration strategy based on q-learning network. In: IEEE international conference on real-time computing and robotics (RCAR), pp. 57–62. IEEE (2016)
18. Li, Y., Cui, R., Li, Z., Xu, D.: Neural network approximation-based near-optimal motion planning with kinodynamic constraints using rrt. IEEE Transactions on Industrial Electronics (2018)
19. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D. arXiv:1509.02971 (2015)
20. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International conference on machine learning, pp. 1928–1937 (2016)
21. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv:1312.5602 (2013)
22. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)
23. Muller, U., Ben, J., Cosatto, E., Flepp, B., Cun, Y.L.: Off-road obstacle avoidance through end-to-end learning. In: Advances in neural information processing systems, pp. 739–746 (2006)
24. Ng, A.Y., Coates, A., Diel, M., Ganapathi, V., Schulte, J., Tse, B., Berger, E., Liang, E.: Autonomous inverted helicopter flight

via reinforcement learning. In: Experimental Robotics IX, pp. 363–372. Springer (2006)
25. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: Theory and application to reward shaping. In: ICML, vol. 99, pp. 278–287 (1999)
26. Pfeiffer, M., Schaeuble, M., Nieto, J., Siegwart, R., Cadena, C.: From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In: 2017 IEEE international conference on robotics and automation (icra), pp. 1527–1533. IEEE (2017)
27. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. arXiv:1710.05941 (2018)
28. Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. arXiv:1511.05952 (2015)
29. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization. In: International conference on machine learning, pp. 1889–1897 (2015)
30. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv:1707.06347 (2017)
31. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. Nature **529**(7587), 484 (2016)
32. Tai, L., Liu, M. arXiv:1610.01733 (2016)
33. Tai, L., Paolo, G., Liu, M.: Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp. 31–36. IEEE (2017)
34. Tambet, M., Avital, O., Taco, C., John, S.: Teacher-student curriculum learning. arXiv:1707.00183 (2017)
35. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: AAAI, vol. 2, pp. 5. Phoenix, AZ (2016)
36. Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N.: Dueling network architectures for deep reinforcement learning. arXiv:1511.06581 (2015)
37. Wu, Y., Mansimov, E., Grosse, R.B., Liao, S., Ba, J.: Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In: Advances in neural information processing systems, pp. 5279–5288 (2017)
38. Xiao, H., Cui, R., Xu, D.: A sampling-based bayesian approach for cooperative multiagent online search with resource constraints. IEEE Trans Cybern **48**(6), 1773–1785 (2018)
39. Xie, C., Patil, S., Moldovan, T., Levine, S., Abbeel, P.: Model-based reinforcement learning with parametrized physical models and optimism-driven exploration. In: 2016 IEEE international conference on robotics and automation (ICRA), pp. 504–511. IEEE (2016)
40. Zaremba, W., Sutskever, I.: Learning to execute. arXiv:1410.4615 (2014)
41. Zhang, F., Leitner, J., Milford, M., Upcroft, B., Corke, P.: Towards vision-based deep reinforcement learning for robotic motion control (2015). arXiv:1511.03791

Yushan Sun received the M.S. degree and the Ph.D. degree in Building of Ship and Ocean Structure from Harbin Engineering University, Harbin, China, in 2005 and 2008 respectively .He is currently working as a Professor in the Science and Technology on Underwater Vehicle Laboratory, Harbin Engineering University, Harbin 150001, China. His current research interests include intelligent control and autonomous decision-making for underwater vehicle.

Junhan Cheng received the B.S. degree in Building of Ship and Ocean Structure from Harbin Engineering University, Harbin, China, in 2012. He is currently working toward the M.Eng. degree in Harbin Engineering University. His current research interests include motion planning and reinforcement learning.

Guocheng Zhang received the Ph.D. degree in Building of Ship and Ocean Structure from Harbin Engineering University, Harbin, China, in 2013. He is currently working as a Lecturer in the Science and Technology on Underwater Vehicle Laboratory, Harbin Engineering University, Harbin 150001, China. His current research interests include the technology of autonomous underwater vehicle.

Hao Xu received B.S. degree in Naval Architecture and Marine Engineering from Jiangsu University of Science and Technology, Zhenjiang, China, in 2015 , the M.Eng. degree in Naval Architecture and Marine Engineering from Harbin Engineering University, Harbin, China, in 2017. He is currently working toward the Ph.D. degree in Harbin Engineering University. His current research interests include intelligent control technology of maritime robotics and autonomous underwater vehicle design.