Deep reinforcement learning for adaptive path planning and control of Autonomous Underwater Vehicle

Behnaz Hadi[1], Alireza Khosravi[*1], Pouria Sarhadi[2]

[1] Department of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran

[2] School of Physics, Engineering and Computer Science, University of Hertfordshire, Hatfield, England, United Kingdom

Abstract

Research into intelligent motion planning methods has been driven by the growing autonomy of autonomous underwater vehicles (AUV) in complex unknown environments. Deep reinforcement learning (DRL) algorithms with actor-critic structures are optimal adaptive solutions that render online solutions for completely unknown systems. The present study proposes an adaptive motion planning and obstacle avoidance technique based on deep reinforcement learning for an AUV. The research employs a twin-delayed deep deterministic policy algorithm, which is suitable for Markov processes with continuous actions. Environmental observations are the vehicle's sensor navigation information. Motion planning is carried out without having any knowledge of the environment. A comprehensive reward function has been developed for control purposes. The proposed system is robust to the disturbances caused by ocean currents. The simulation results show that the motion planning system can precisely guide an AUV with six-degrees-of-freedom dynamics towards the target. In addition, the intelligent agent has appropriate generalization power.

Keywords: Autonomous underwater vehicle (AUV), Deep reinforcement learning, Motion planning, Obstacle avoidance, Adaptive actor-critic network

Introduction

Underwater vehicles have been increasingly used for a variety of purposes, such as deep-ocean exploration, pipeline and cable monitoring, ocean floor mapping, environmental security, etc. Therefore, enhancing the autonomy of these vehicles has been of great significance for researchers. Studies in underwater robotics have largely focused on developing intelligent decision-making algorithms that enable autonomous operation in the ocean. The path planning system generates guidance information such as feasible trajectories after considering the starting and goal points and the following constraints: the shortest possible distance to the target, path smoothness, a safe distance from the obstacles, and environmental disturbances. The control system provides the required forces and moments to satisfy these purposes [1]. AUVs have highly nonlinear uncertain dynamics. Therefore, adaptive control methods are extensively used in the AUV control problems [2-5]. These vehicles traverse in unstructured and unexplored environments that have been affected by environmental factors such as ocean waves and currents, etc. To provide safe guidance in such an environment, AUVs must calculate or modify their feasible and collision-free paths based on the new information collected from their surroundings [6, 7]. Several advanced approaches,

[*] Email: akhosravi@nit.ac.ir

including geometric model search [8], artificial potential field algorithms [9], probabilistic sampling methods [10], and evolutionary algorithms [11], can be mentioned as AUV's path planning methods [12, 13].
Reinforcement learning (RL) is another important branch of artificial intelligence, which has been increasingly used in the control design and path planning of AUVs. The RL approach involves one or more agents establishing real-time interactions with an environment that is possibly unknown to the agents and is often expressed based on Markov processes. Based on their experience, the agents learn to select optimal strategies to reach a specific target and maximize their cumulative rewards. Another subfield of machine learning that has received myriad attention in recent years is deep learning, which is the ability to understand and learn the world hierarchically. Deep learning algorithms exploit the proper representation at several levels of unknown structures in the input distribution [14, 15]. Reinforcement learning is a class of methods that incorporate adaptive and optimal control features [16-20]. Deep RL, which combines reinforcement learning and deep learning, is a powerful tool for solving complex problems in autonomous systems and adapting to unpredictable or uncertain conditions. The DRL technique will increase efficiency and reduce the complexity of control system designs [16, 21].

A Q-learning algorithm was used in [22] for an underactuated marine vehicle's path planning with 3-DOF kinematic equations. A comparative study was conducted employing the A* and D* approaches; however, there was no consideration for any obstacle. Standard RL approaches operate on domains with discrete state-action spaces. When encountering problems with large or continuous space, these approaches experience a curse of dimensionality. A combination of function approximators (e.g., neural networks) and RL approaches solved these problems. Therefore, there was no need to store the feedback provided by the state-action value function. Moreover, it was unnecessary to have a large amount of memory and accurate environmental information. A marine vehicle's path planning based on a two-layered RL was presented in [23]. The first layer used the least-squares method for policy iteration to provide the desired velocity direction signal to the second layer (lower-level). The effects of environmental disturbances such as waves, wind, and ocean currents were considered. [24] described the use of modified Q-learning to avoid collisions with static obstacles in AUVs. In this approach, an unsafe zone was defined around the obstacles to reduce the possibility of any collision. As soon as the AUV entered this zone, the exploration process stopped, and the exploitation demand was fulfilled until the AUV left the danger zone. In 2020, Sun et al. deployed the hierarchical deep Q-Network with prioritized experience replay for AUVs' three-dimensional path planning. For this purpose, the path planning task was divided into three layers, which reduced the size of the state space, and an artificial potential field was employed to solve the sparse reward task [25]. Another approach included using deep interactive reinforcement learning for AUVs' path-following. In this approach, DQN was combined with interactive reinforcement learning, and the reward function had environmental and human rewards [26]. Zhao et al. [27] proposed a collision-free model for multiple unmanned ships using the PPO algorithm. The simulation results demonstrated that the ships would avoid collisions with each other while following the predetermined paths. Cheng et al. [28] proposed a concise deep reinforcement learning obstacle avoidance algorithm for 3-DOF unmanned marine vessels based on the deep Q-networks architecture. A convolutional neural network (CNN) was used to capture the vehicle's state and perception information. While deep Q-Network solved problems with high-dimensional observation space [29], it could only work with low-dimensional and discrete action space. Given that most physical control tasks are high-dimensional and have a continuous action space, the DDPG algorithm was consequently introduced. This algorithm learned policies in high-dimensional continuous action spaces [30].

Reference [31] used reinforcement learning strategies in continuous action space (DDPG) and discrete action space (DQN) for AUV docking. These strategies were compared with optimal and classical control techniques. The DDPG method was employed in [32] for an AUV's position tracking. The AUV was guided towards the target by using six thrusters. In [33], a 3-DOF AUV's motion planning was performed according to the PPO algorithm without prior knowledge of the environment. After defining reward curriculum learning, the convergence speed increased, and the algorithm was more adaptable to different situations. In [34], a 3D path-following problem and collision avoidance were presented for an AUV. A path to reach a goal was generated based on conventional waypoint interpolation methods. The tracking error vector included cross-track and vertical track errors. The deep intelligent agent was trained to reduce the tracking error and avoid collisions with obstacles. In [35], an unmanned ship's path planning was performed by using a hybrid DDPG and a virtual potential field. A combination of attractive and repulsive potential functions was used as a cumulative return function for the DDPG. This study did not consider the vehicle's motion dynamic model, and path planning and collision avoidance did not use range finder sensors within the RL framework. In [36], path planning and large-scale continuous obstacle avoidance were prepared for the simple 3-DOF kinematic model based on the deep reinforcement learning method. The AUV had three thrusters: one tail thruster and two side thrusters. The action had four elements: angular velocity, forward speed, and the AUV's position in x and y directions. Therefore, it did not take into account nonholonomic conditions. The effect of the AUV's model to generate a collision-free path was ignored, and the effect of ocean currents was not investigated. Despite the acceptable performance of the DDPG algorithm, it does not usually perform well in complex environments due to over-fitting in current policies, hyper-parameters, and other adjustments [37]. The TD3 algorithm is an extension of DDPG called twin delayed deep deterministic policy gradient. This technique employs three solutions to solve the problems mentioned: dual critic networks, target policy smoothing, and policy delays. This method contains an actor-critic structure that considers the interactions between function approximation errors in both policy and value updates. Compared to DDPG, TD3 speeds up learning and improves performance in some challenging tasks [37].

As discussed in the surveys [7], DRL methods are potential alternatives in AUVs' path planning and control, and research in this field is ongoing. Most of the current AUV-related literature focuses on using value-based algorithms for AUV path planning. However, the discretization of action suffers the accuracy of real-time control. Some studies make use of actor-critic-based methods. The remaining gaps are as follows:

- Unrealistic assumptions such as neglecting obstacles, environmental disturbances, non-holonomic conditions, or dynamic models
- Ignoring desirable output trajectory properties such as feasibility and safety
- Using automatic identification system (AIS) data which is not available underwater [27, 35]

Given the notes mentioned above, the main contributions of the present study are as follows:

- A combined (end-to-end) path planning and control technique for an AUV is presented based on a twin-delayed deep deterministic policy. The AUV's continuous control inputs are computed using the TD3 algorithm through online measurements without prior knowledge of the environment.
- A comprehensive reward function based on the target approach, obstacle avoidance, energy optimization, and the vehicle's practical limitations is proposed.
- The effects of ocean currents are accounted for in the training procedure. A stochastic Gaussian model for the ocean current is exploited.
- Random obstacle locations and ocean current presence show the algorithm's generalizability.
- In our study, to consider the effect of the dynamic limitations on motion planning and obstacle avoidance, an underactuated 6-DOF model is utilized.

The present article includes the following sections: The AUV’s kinematic and dynamic equations, reinforcement learning, deep neural network, and TD3 algorithm are overviewed in Section 2. the proposed DRL-based adaptive motion planning is outlined in Section 3. Simulations and conclusions are presented in Sections 4 and 5, respectively.

2. Background

This section discusses the proposed path planning and control scheme basics: the AUV motion model, reinforcement learning, a deep neural network approximator, and the TD3 algorithm.

2.1. AUV motion model

Most path planning or control methods require a model of the system to design or simulate the behavior of AUVs. Non-linear equations describe the dynamics of underwater vehicles with different coefficients, which have remarkable impacts on the AUV's performance, maneuverability, and controllability. Earth-fixed frame and body-fixed frame are two coordinate systems. The body-fixed frame is attached to the vehicle and rotates with the earth-fixed frame (Fig.1).

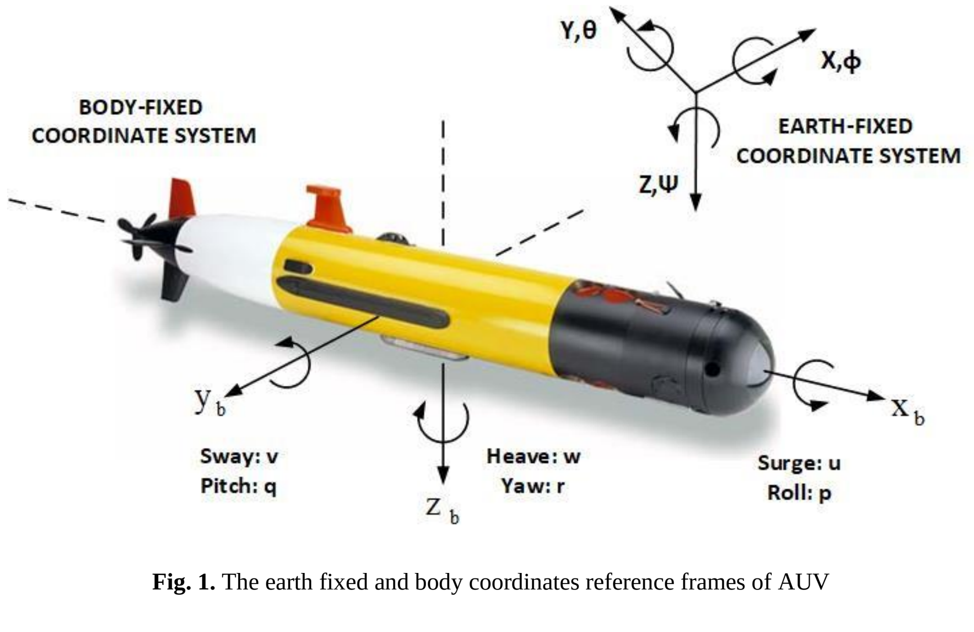

Fig. 1. The earth fixed and body coordinates reference frames of AUV

2.1.1. Kinematic equations

Kinematic equations in a vector form are expressed as below [38, 39]:

$$\dot{\eta} = J(\eta)\nu \tag{1}$$

where $\eta = [x, y, z, \phi, \theta, \psi]^T \in R^6$ and x, y, z refer to North–East–Down (NED). In addition, ϕ, θ, ψ are Euler angles and $\nu = [u, v, \omega, p, q, r]^T \in R^6$ is the vector of linear and angular velocities in the body-fixed frame. The movements along the longitudinal, lateral, and vertical axes are called by surge, sway, and heave, respectively. And the rotations around these axes are known as roll, pitch, and yaw. Although there is usually a collection of these movements, it is possible to classify them into separate categories by simplifying their behavior appropriately. The J transformation matrix relates the velocities defined in the body-fixed frame to the earth-fixed frame.

2.1.2. Dynamic equations

Based on Newton's Second Law, the motion equations in the body-fixed Frame are expressed as below [38, 39]

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

In these equations, M is the system's inertia matrix (including the added mass), C is the central matrix (including the added mass), $D(\nu)$ is the damping matrix, $g(\eta)$ is the force vector and gravity/buoyancy moments, and τ is the vector of control inputs. Further information regarding the AUV model and its unknown parameters can be found in [39].

2.2. Reinforcement Learning

Reinforcement learning is a decision-making framework in which an agent learns a desirable behavior or policy by interacting directly with the environment [40]. According to RL, if an action improves performance, the desire to perform that action is reinforced. By just using a scalar evaluation of efficiency, which is called a reinforcement or reward signal, RL is capable of training agents in complex, uncertain, and stochastic environments without needing a supervisor. At each time step, the agent performs action a in state s. Therefore, the agent is placed in a new state s' and receives a reward *r*. A Markov Decision Process is employed to model the action choice depending on a value function, representing the future reward estimation. Having had a long-term interaction with the environment, the agent learns an optimal policy that maximizes the total expected return. The expected return is defined as the discounted sum of future rewards:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+(k+1)} \tag{3}$$

$\gamma \in [0 \;\; 1]$ is the discount factor that determines the current amount of future reward. The expected discounted state-value function, which starts from state s and follows policy π, is defined as:

$$V^{\pi}(s) = E_{\pi}[R_t \mid s_t = s] = E_{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{t+(k+1)} \mid s_t = s] \tag{4}$$

Similarly, the action-value function, which expresses the action value a in state s under the policy π, is defined as:

$$Q^{\pi}(s,a) = E_{\pi}[R_t \mid s_t = s, a_t = a] = E_{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{t+(k+1)} \mid s_t = s, a_t = a] \tag{5}$$

When an agent starts from state s and follows an optimal policy, the optimal state-value function shows the maximum discounted reward obtained.

$$V^*(s) = \max_a E[r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a] \tag{6}$$

The mentioned equation is the Bellman Optimality Equation, which states that the value of a state under an optimal policy must equal the expected return for the best action in that state. Similarly, the Bellman Optimality Equation for the action-value equation is:

$$Q^*(s,a) = E[r_{t+1} + \gamma \max_a Q^*(s_{t+1}, a') \mid s_t = s, a_t = a] \tag{7}$$

In this equation, $V^*(s) = \max_a Q^*(s,a)$ is for all of s. When Q^* is identified through the interactions, the optimal policy is directly obtained from the following equation:

$$\pi^*(s) = \arg\max_a Q^*(s,a) \tag{8}$$

2.3. Deep Neural Network Approximator

Deep neural networks are networks that are structured in a deep architecture, such as a mammalian brain. In short, deep neural network architectures are structures of successive layers that can convert a high-dimensional input into a reduced output feature. Network parameters are learned in a supervised manner using learning algorithms such as the gradient descend method in the back-propagation algorithms [15]. The actor-critic architecture is used, where deep neural networks are employed to represent policy and state-action value functions. Fully connected neural networks with RELU layers are used to approximate these functions, as shown in Fig. 2.

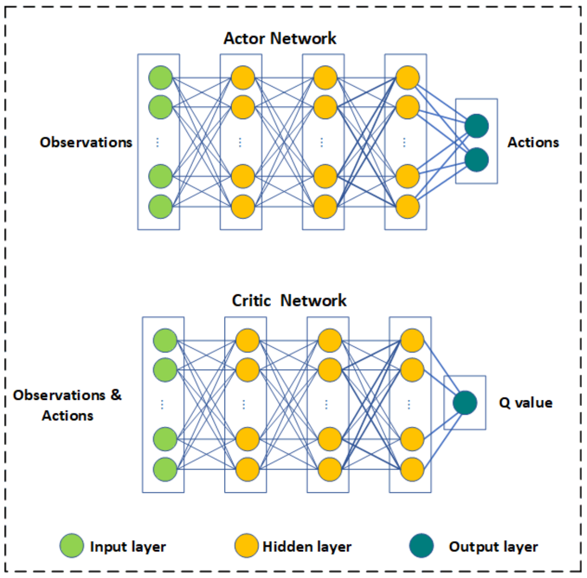

Fig. 2. Structure of the actor and critic networks

2.4. TD3 Algorithm

The DDPG algorithm employs two deep neural networks to approximate the state-action value and policy functions with replay buffer ideas and target networks. The replay buffer breaks up the Markovian nature of the sampled data. Two target networks are utilized to ensure the stability of the training process. TD3 is a model-free, online, and off-policy reinforcement learning algorithm based on the DDPG. This approach implements a scheme that is similar to double-DQN. It reduces over-fitting in function approximation and delays the actor network's update frequency. Moreover, it eliminates the sensitivity and instability of the DDPG network by adding noise to the target actor network. Algorithm 1 shows the different steps to implementing this algorithm [30, 41].

Algorithm 1: TD3 Algorithm.

```
1. Initialize Critic networks parameters $\phi_1, \phi_2$, and actor network parameter $\theta$
2. Initialize target networks parameters equal to main parameters $\phi_{\text{targ}_1} \leftarrow \phi_1,\quad \phi_{\text{targ}_2} \leftarrow \varphi_2,\quad \theta_{\text{targ}} \leftarrow \theta$
3. Initialize replay buffer
4. For $1$ to $N$ do
5. Select action with exploration noise: $a = \mu_\theta(s) + \varepsilon,\quad \varepsilon \sim \mathrm{N}(0,\sigma)$
   execute $a$ in the environment then observe reward $r$ and new state $s'$
6. Store transition tuple $(s, a, r, s')$ in replay buffer
7. Randomly sample a batch of transitions, $M = \{(s, a, r, s')\}$ from replay buffer
8. Compute target actions: $a'(s') = \mu_{\theta_{\text{targ}}}(s') + clip(\varepsilon, -c, c)$, $\varepsilon \sim \mathrm{N}(0,\sigma)$
9. Compute targets: $y \leftarrow r + \gamma \min_{i=1,2} Q_{\phi_{\text{targ}_i}}(s', a'(s'))$
```

```
10. Update Q-functions: $\phi_i \leftarrow r + \arg\min_{\phi_i} \frac{1}{M}\sum\left(y - Q_{\phi_i}(s,a)\right)^2, i = 1,2$
11. If $t$ mod $d$ then
12. Update $\theta$ by the deterministic policy gradient: $\nabla_\theta J(\theta) = \frac{1}{M}\sum_{s\in M}\nabla_\theta Q_{\phi_1}\left(s, \mu_\theta(s)\right)$
13. Update target networks:
        $\phi_{\text{targ}_i} \leftarrow \tau\phi_{\text{targ}_i} + (1-\tau)\phi_i,\ for\ i = 1,2$
        $\theta_{\text{targ}} \leftarrow \tau\theta_{\text{targ}} + (1-\tau)\theta$
14. End if
15. End for
```

Lines 1-2 show the initialization of actor-critic networks with random values. Target networks' initialization is exerted with the same values. Line 3 is the initialization of the replay buffer to store experiences in the form of a tuple. In line 5, which is a step in Markov processes in a stochastic environment, an action is selected with exploration noise. In line 6, after applying the selected action to the environment, state transitions are stored in the buffer to update the networks. One of the most significant steps of the algorithm is implemented in lines 6-10, which includes updating the critic network. When a certain number of transitions are stored in the buffer, a mini-batch of the stored state transitions is sampled. In the next step, an action is selected for each of the states of the mini-batch. In line 8, target policy smoothing is performed as one of the steps in improving the TD3 algorithm. In this step, a clipped noise is added to the target policy. In line 9, the critic's target Q-values are calculated. In TD3, two target critic networks are used to evaluate the Q-value to learn an optimal policy. The smaller network's Q-value updates the critic networks' parameters by minimizing the Mean Squared Bellman Error (MSBE) in a gradient descent method (Line 10).

$$L(\phi_k) = \frac{1}{M}\sum_{i=1}^{M}\left(y_i - Q\left(s_i, a_i \mid \phi_k\right)\right)^2 \quad, k = 1,2 \tag{9}$$

In this equation, y_i is the target state-action value with a smaller value generated by target deep networks $Q_{\phi_{\text{targ1}}}, Q_{\phi_{\text{targ2}}}$, parametrized by $\phi_{\text{targ}_1}, \phi_{\text{targ}_2}$.

$$y_i = r(s_i, a_i') + \gamma \min_{k=1,2} Q_{\phi_{\text{targ}_k}}(s_i, a_i' \mid \phi_k) \tag{10}$$

Here, the target action a_i' is the output of the Actor target neural network μ_{targ}:

$$a_i' = \mu_{\text{targ}}(s' \mid \theta_{\text{targ}}) \tag{11}$$

Another feature of the TD3 algorithm is the delayed update of the actor network compared to the critic network. The actor network is updated after each d time step. After the adjusting-delay procedure, the actor-policy function, represented by the deep network μ_θ, is updated using the critic value function ϕ_1, which is based on a deterministic gradient to optimize the expected return. In addition, the target network parameters are updated by using the rules of line 13, which are for the actor and critic network parameters.

3. Adaptive AUV Motion Planning using twin delayed deep deterministic policy gradient

The underwater vehicles' highly nonlinear dynamics and the unknown disturbances resulting from environmental conditions are challenges for AUV design. Therefore, adaptive approaches that do not require dynamic models can be charming. The reinforcement learning algorithms based on actor-critic structures are optimal adaptive approaches that converge online to an optimal solution for a completely unknown system. The critic identifies and evaluates the performance of the current control policy and uses this information to update the controller. An optimal control policy is learned by using the data obtained from interacting with the system without prior knowledge of system dynamics and solving the Bellman equation. In reinforcement learning formulations, the state of a Markovian underwater vehicle is defined by using a set of observable variables. In this approach, the Markov system state includes information provided by onboard sensors, which are linear and angular velocities in the fixed body coordinate system and the AUV's position. The control signal a is the angle change of the rudder. Policy gradient approaches are widely used for reinforcement learning in continuous time. These model-free approaches can be employed to solve robotic problems without prior knowledge of the problem or the robot's dynamics. The policy function uniquely maps states to actions. A control policy's performance improvement is achieved by updating the function's parameters in the direction of the performance gradient. These approaches usually approximate a stochastic policy using a function approximator to maximize the expected future reward. The deterministic policy gradient algorithm has better computational efficiency than the stochastic policy.

The model-free reinforcement learning approach with actor-critic architecture, based on deterministic gradient theory, was used to solve the motion planning problem. The actor is an action selection policy in this structure that maps continuous states to continuous actions in a deterministic path. The critic is a state-value function that maps the states to the expected cumulative reward. The actor-critics cannot learn directly from the standard table-based Q-learning algorithm in continuous control problems. Thus, function approximators are required. A deterministic policy approximates the actor's behavior $\mu_\theta \approx \pi$ with periodically updated parameters using a recursive rule. Therefore, adaptation is achieved by continuously updating the policy parameters based on the experience gained from the interactions between the robot and its environment. Moreover, the critic is approximated by the deep network $Q^\phi(.,.)$ as $Q^\phi(s,a) \approx Q^\pi(s,a)$. Fig. 3 shows the actor-critic architecture for AUV motion control.

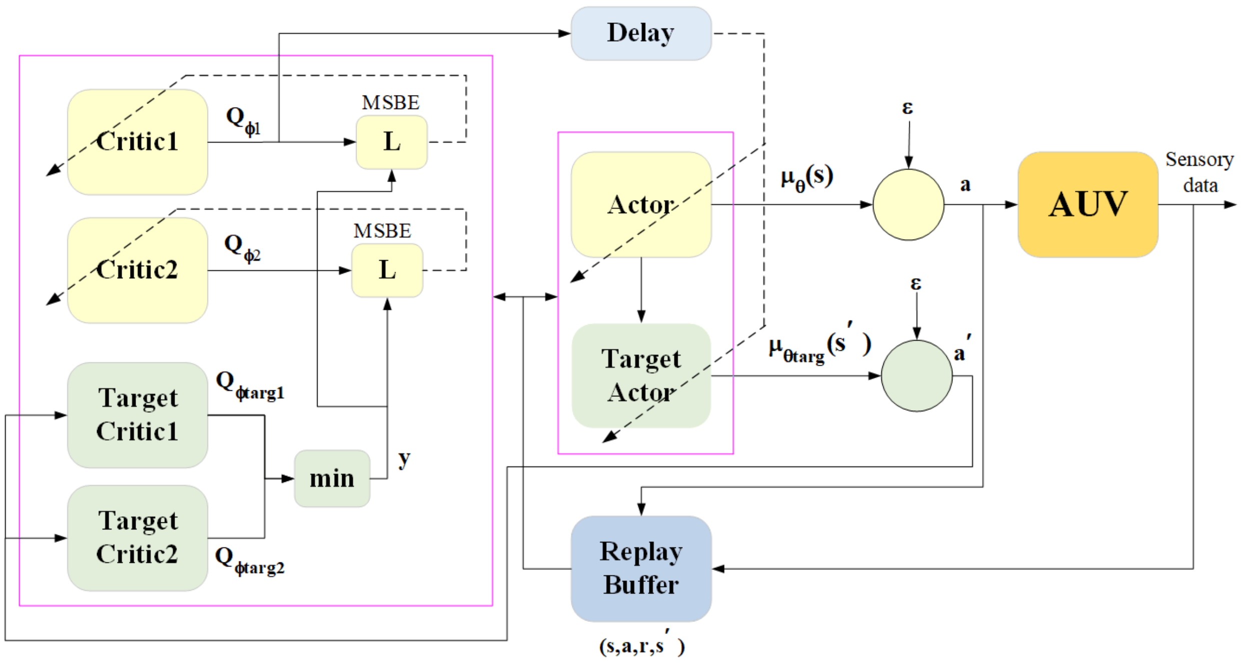

Fig. 3. Structure of the AUV's motion planning system

3.1. DRL-based Motion Planning Implementation

This section aims to design an adaptive motion planning system for an underwater vehicle so that it can cope with different operating conditions. For this purpose, control and path planning are carried out based on the actor-critic architecture to achieve a strong control policy. A path planning system is designed to generate various goals. This system includes creating a starting zone with random points, creating a target zone that changes randomly at each training step, and generating the coordinates of random obstacles. The vehicle's navigation information is obtained from the robot's sensor system, which includes its controllable variables. The difference between the path planning purposes and the robot's current and expected dynamic behavior can be determined using this information. The error signals, which indicate the robot's direction and distance from the target at each time step, provide the deep agent with immediate information on the performance of the path planning system. In addition, the intelligent agent obtains information about the control system's behavior, which is summarized in the last performed action a_{t-1}. The deep agent receives the summarized information in the Markovian system state s_t =[Auv state, distance from obstacles, goal coordinate,controle effort]. This measurable information is obtained from the vehicle's sensors used in underwater robots, and there is no need for the video images employed in most deep learning applications. The state of the Markovian system at any time (t) is described by observable state variables derived from immediate measurements of the robot's sensors, such as $s_t = [AUV_{state}, d_{(AUV,Obs)}, P_{goal}, a_{t-1}] \in R^{14+N}$, N, which represents the number of identified obstacles, and R^m, which represents the m-dimensional Euclidean space. $AUV_{state} = [u, v, w, p, q, r, \phi, \theta, \psi, x, y] \in R^{11}$, $d_{(AUV,\ Obs)} = [\|P_{AUV} - P_{obs_1}\|, \cdots, \|P_{AUV} - P_{obs_N}\|] \in R^N$, $\|\square\|$ represents the Euclidean norm. $P_{goal} = [x_{goal}, y_{goal}] \in R^2$ and $a = \delta_r \in R$. The aim is to minimize the distance covered to reach the target, avoid collisions with obstacles as shown in Fig. 4, reduce energy consumption, and reduce the sudden change in the control signal. Section 3.1.1 explains the reward function for this purpose. Dynamic equations of the 6-DOF REMUS AUV are used to implement the proposed approach. The values of its parameters are based on [42]. Path planning is executed in a horizontal plane.

3.1.1 Reward function framework

For planning a feasible path, collision avoidance, and AUV control, the reward function is designed as follows:

1) Goal reward: The AUV and the target are placed at random coordinates in each episode. An agent receives a reward (r_{goal}) when it reaches the target. In approaching the target, the second term below leads to a higher positive score for the AUV and increases the algorithm's convergence speed ($r_{near} < r_{goal}$). Otherwise, it will receive a penalty proportional to its distance from the target point.

$$r_1 = \begin{cases} r_{goal} & \text{if } \left\| P_{AUV} - P_{goal} \right\| \le d_g \\ r_{near} & \text{if } \mathrm{d}_g < \left\| P_{AUV} - P_{goal} \right\| \le d_n \\ -k_1 \left\| P_{AUV} - P_{goal} \right\| & \text{otherwise} \end{cases} \tag{12}$$

where d_g is defined as the target region radius, P_{AUV} is the AUV's position at each time step, and P_{goal} is the target coordinates that change randomly in each episode. r_{goal} , r_{near} , k_1, and d_n are positive constants and $d_n > d_g$.

2) Obstacle avoidance reward: Furthermore, to effectively avoid obstacles while moving towards the target, the AUV is punished if its distance from the obstacles is less than the safe distance.

$$r_2 = \sum_{i=1}^{\ell} r_i , \quad r_i = \begin{cases} k_{21} & \text{if } \left\| P_{AUV} - P_{obs_i} \right\| > d_O \\ -k_{22} \left| d_O - \left\| P_{AUV} - P_{obs_i} \right\| \right| & \text{otherwise} \end{cases} \tag{13}$$

where ℓ shows the number of obstacles identified by the AUV's range of sensors. P_{obs_i} is an obstacles' coordinates. d_O is the minimum safe distance from obstacles. k_{12} and k_{22} are positive constants.

3) Heading-error reward: This reward confines the change of direction of the AUV and causes effective movement towards the target.

$$r_3 = -k_3 \left| \left(\psi_{LOS0} - \psi_{AUV} \right) \right| \tag{14}$$

where $\psi_{LOS0} = atan\left(\frac{y_{goal}\text{-}y_{start}}{x_{goal}\text{-}x_{start}}\right)$ is the initial line of sight angle in the starting zone with the target.. k_3 is a positive constant.

4) Reducing control effort: This following term reduces the overall control effort.

$$r_4 = -k_4 \left|\delta_r\right| \tag{15}$$

δ_r is a rudder command, and it is limited to the range of -25 to 25 degrees due to practical restrictions and actuator saturation level. k_4 is a positive constant.

5) Reducing control signal fluctuations: This decreases sudden variations in control signal.

$$r_5 = -k_5 \left|a_{t-\tau:t-1} - a_t\right| \tag{16}$$

The difference between the current action (a_t) and the moving average of τ earlier actions ($a_{t-\tau:t-1}$) is considered a fine for sudden change in rudder angle. k_5 is a positive constant.

The total reward in each episode is equal to the sum of all weighted rewards. Each of the above gains $\left(\mathrm{k}_1, \mathrm{k}_{21}, \mathrm{k}_{22}, \mathrm{k}_3, \mathrm{k}_4, \mathrm{k}_5\right)$ indicates the importance of each section in the control policy.

$$R_T = r_1 + r_2 + r_3 + r_4 + r_5 \tag{17}$$

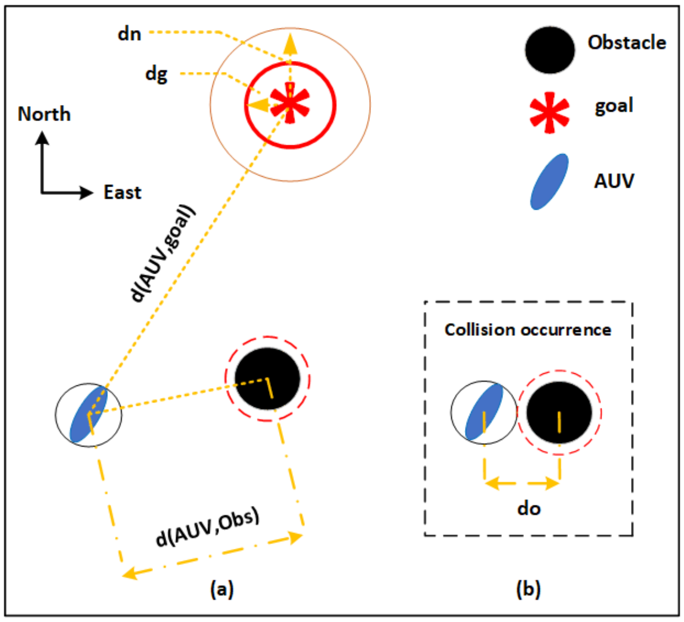

Fig. 2. (a) Representation of AUV, Obstacle and Target. (b) A collision occurrence

The flowchart in Fig. 5 depicts how the environment of the AUV motion planning problem is defined. In each learning episode, the environment randomly samples a goal point and resets the AUV's position, heading, and velocity. The obstacles are placed at random. At each time step of an episode, the action of the agent a_t is determined at state s_t. System states are updated with Equations 1 and 2. Then, the observation states s_{t+1} and r_t are calculated. This process in each episode is repeated until the AUV reaches the target area or time steps have elapsed.

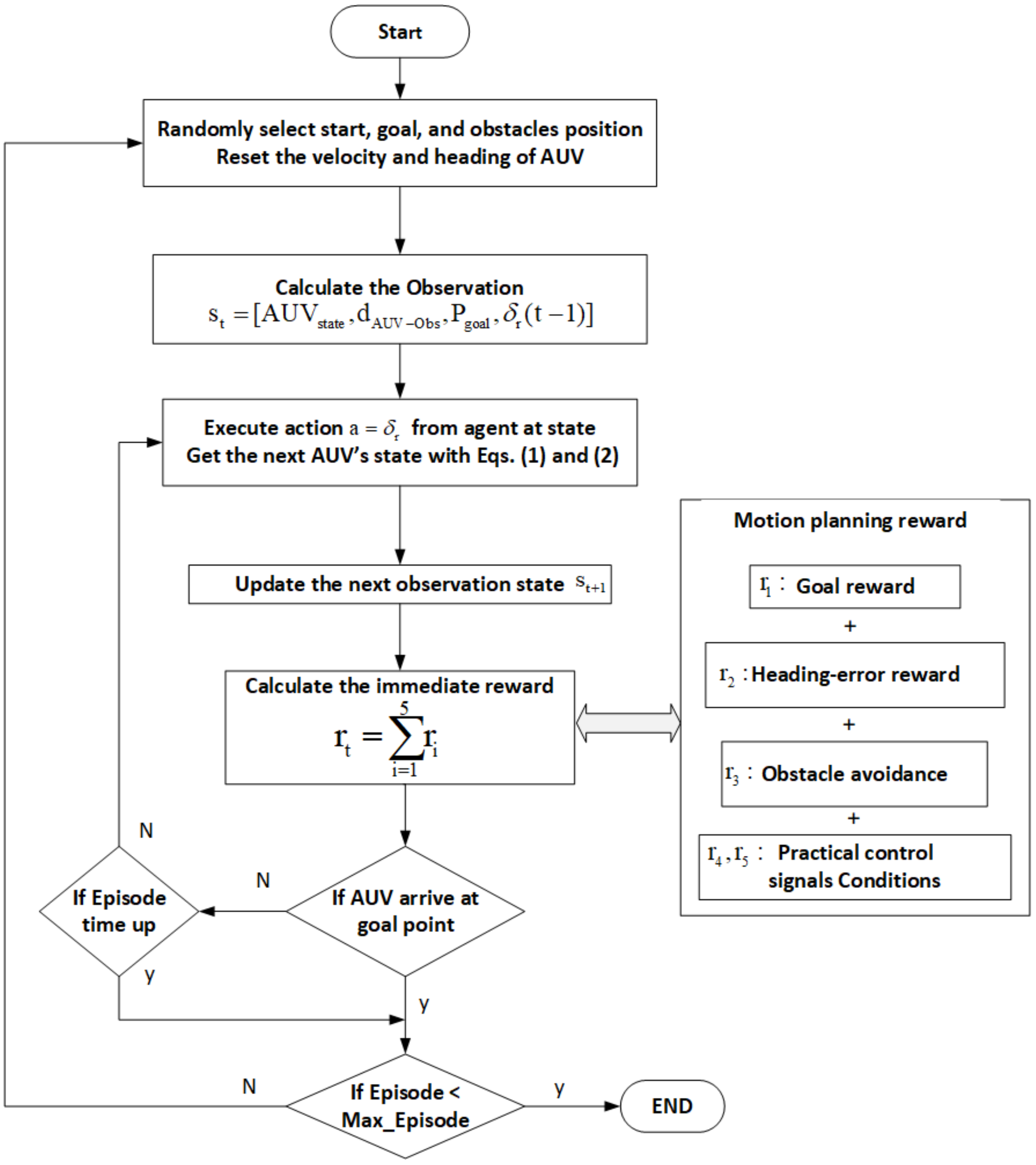
Start
Randomly select start, goal, and obstacles position
Reset the velocity and heading of AUV
Calculate the Observation
$s_t = [AUV_{state}, d_{AUV-Obs}, P_{goal}, \delta_r(t-1)]$
Execute action $a = \delta_r$ from agent at state
Get the next AUV's state with Eqs. (1) and (2)
Update the next observation state s_{t+1}
Calculate the immediate reward
$r_t = \sum_{i=1}^{5} r_i$
Motion planning reward
r_1 : Goal reward
+
r_2 : Heading-error reward
+
r_3 : Obstacle avoidance
+
r_4, r_5 : Practical control signals Conditions
If AUV arrive at goal point
If Episode time up
N
N
y
y
If Episode < Max_Episode
N
y
END

Fig. 5. AUV motion planning method

It should be noted another deep agent is utilized to maintain the depth at a constant value. The agent state observations include $s = [e_z, \int e_z, \dot{e}_z, w, q, \theta]$, $e_z = z_{ref} - z$, and $a = \delta_s$. The reward function is: $r = -\xi_1(e_z{}^2 + \delta_s{}^2) - \xi_2(w^2 + q^2 + \theta^2)$, ξ_1, and ξ_2 are positive constants.

4. Simulation

4.1. Environment Setting

The motion planning simulation of the REMUS-100 AUV in MATLAB is performed with an AMD Ryzen 7 3800XT 8-Core, 3.89 GHz processor. The training environment is a rectangular area that includes AUV, obstacles, and target locations. The 1.4-meter AUV is enclosed in a circle with a radius of 0.7 m. The AUV has a nominal speed of 1.5 m/s and a maximum speed of 2 m/s. Each obstacle has a radius of 1.5 m, and the target area has a radius of 3 m. Figure 4 depicts an avoidance zone of 0.5 m around each obstacle. The target point is randomly considered at the border of the rectangle. At the start of each episode, the AUV's initial position is chosen randomly in a 5*5 square zone in the center of the training area. The AUV's heading angle is set to zero. The coordinates of the obstacles and the target are generated at random. A collision occurs when the AUV's circumferential circle and the avoidance circle (red-dash) converge (Fig. 4).

4.2. Parameter Setting and Training Process

The structure of the actor-critic networks is similar, which means that each network has three fully connected layers with 256 neurons in each layer (shown in Fig. 2). These values have been utilized based on the dimension of state and action spaces, which is satisfactory to approximate the state-action and policy functions. Selection of action is carried out to fully exploit the state and action spaces using the Ornstein-Uhlenbeck process noise. The noise process is defined as follows:

$$\mathrm{N}_{k+1} = \mathrm{N}_k + \alpha_{OU}(\mu_{OU} - \mathrm{N}_k) + \sigma_{OU}\mathrm{N}_G(0,1) \tag{18}$$

where N_k and N_{k+1} are the values of the Ornstein-Uhlenbeck process at time k, k+1, respectively. $\mathrm{N}_G(0,1)$ is Gaussian noise with zero mean and a standard deviation 1. α_{OU}, μ_{OU}, and σ_{OU} are the parameters of the Ornstein-Uhlenbeck process. α_{OU} is a constant that determines how quickly the noise output is attracted to the mean. μ_{OU} and σ_{OU} are the mean and variance of the noise model respectively. The values of the noise parameters and other TD3 parameters based on path planning and AUV control are listed in Table 1.

Table 1: List of parameters used in TD3 algorithm

Parameter	Value
Actor learning rate	0.001
Critic learning rate	0.001
Discount factor	0.99
Batch size	256
Memory size	1000000
Hidden layers 1, 2 and 3	256 units

Smooth update τ	0.005
Policy and target delay update	2
Sample time	0.1 s
α_{OU}	1
μ_{OU}	0
σ_{OU}	0.05

The learning rate is 0.001, and the discount factor is 0.99. The size of the mini-batch is 256. The AUV can move 500 steps in each episode, and the total training episode is 10,000. The AUV's action is the rudder angle, so the heading and position of the AUV are updated by equations (1) and (2). Termination occurs when the AUV reaches the target area or the entire time step of each episode has passed. The total training time with 10,000 episodes was approximately 60 hours. The learning diagram is shown in Fig. 6. The vertical axis shows the average rewards for every 100 episodes. According to the figure, the average reward gradually increases with training episodes, and the reward value becomes stable.

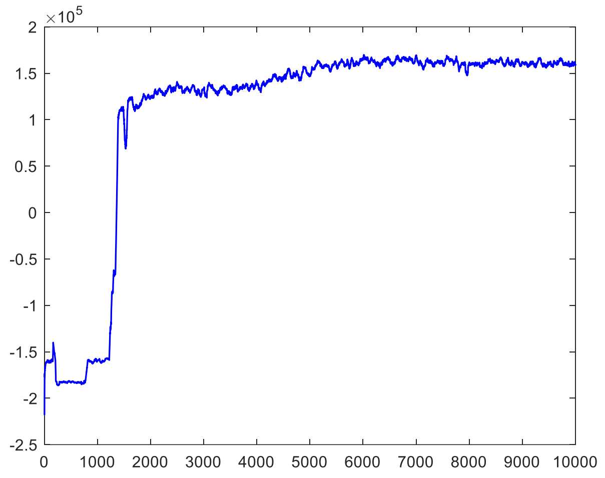

Fig. 6. Average rewards per episode

The AUV's moving trajectories during various steps of learning are depicted in Fig. 7. The red star indicates the location of the target, the black circles indicate obstacles, and the blue lines indicate the AUV's trajectories. As observed, the intelligent agent learns to approach the target while avoiding obstacles after approximately 1500 episodes. With increased episodes, the agent attempts to find more efficient routes to the target while avoiding obstacles. In the concluding episodes, it gradually converges on a constant path.

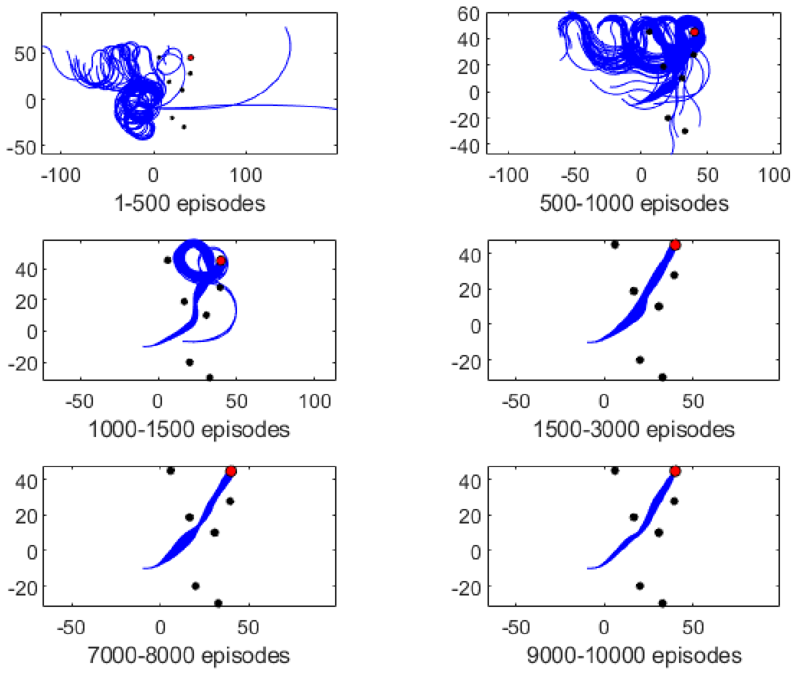

Fig. 7. The motion planning and obstacle avoidance training progress using the TD3 method

4.3. Performance Validation

For evaluating the trained agent, several modes are considered. In the first case, motion planning is completed from different starting points to various goals. The system's performance has been evaluated to achieve multiple goals in a row. To realistically assess the simulations, path planning and control have been examined in the presence of the ocean current. Figures 8–12 show the simulation results.

4.3.1 Results without disturbances

This section discusses motion planning for a single target and a group of consecutive targets without ocean currents.

1) Single target: to evaluate the performance of the AUV motion planning, different conditions are considered according to the AUV start and target coordinates. Obstacles are distributed at random across an unknown environment. The results of six different tests are shown in Fig. 8 to describe the performance of motion planning. This figure represents motion planning, obstacle avoidance, and target achievement at varying distances and directions.

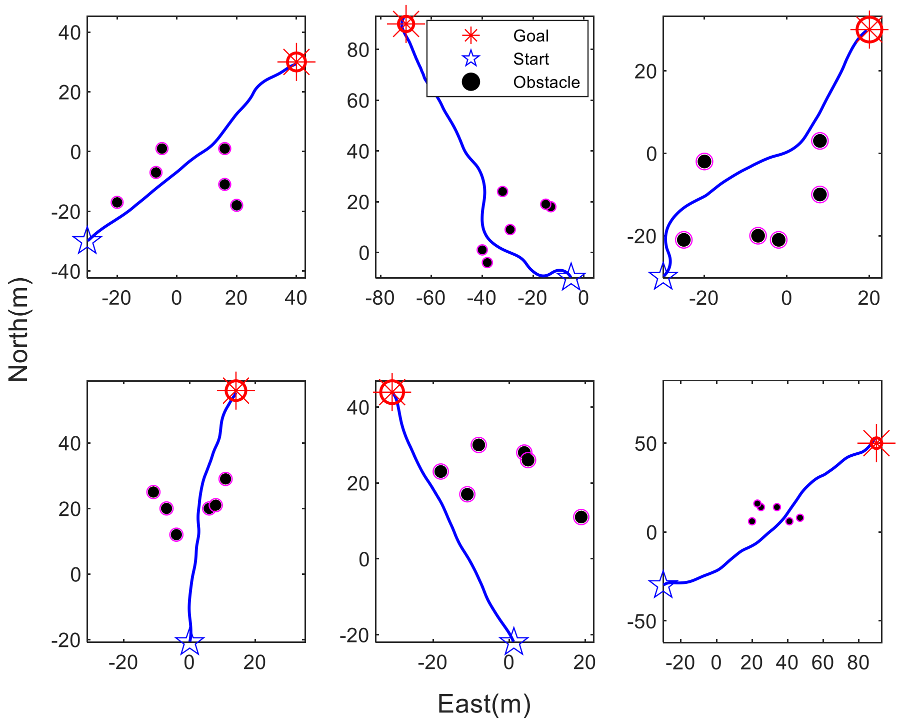

Fig 8. The AUV's trajectory in the single-target unknown environment

2) Multiple consecutive targets (waypoints): this environment is performed to verify the performance of motion planning in different situations. Figure 9 shows trajectory planning for five and three consecutive targets. On the left-hand side of Fig. 9, the movement starts from point (30, -40) and reaches points (-30, 5), (-55, 20), (-40, 30), (-10, 30), respectively, and ends at point (10, 65). On the right-hand side, the AUV moves from points (2, -10) to points (35, 30), (55, -5), and (80, 45), respectively. The results show that the AUV achieves the desired goals and avoids obstacles.

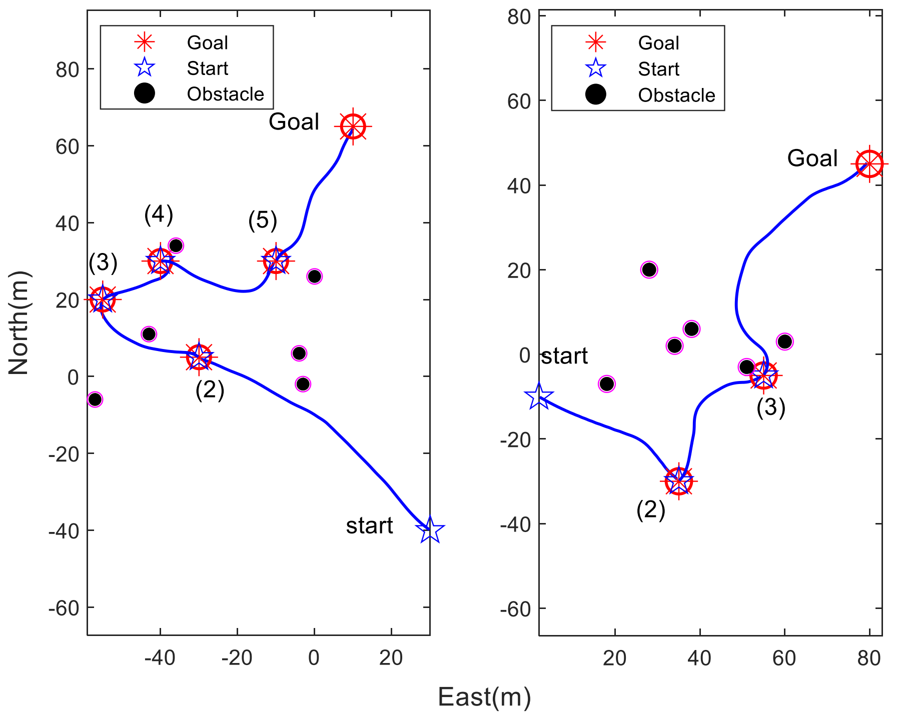

Fig. 9. The AUV's trajectory in the five and three consecutive targets

4.3.2 Results with disturbances

Motion planning in the presence of ocean currents: Ocean currents refer to the horizontal and vertical rotating systems of ocean waters produced by various factors such as wind friction and gravity in various parts of the ocean [38]. The equations of motion can be represented in terms of relative velocity to implement the ocean currents and their effect on AUV motion:

$$v_r = v - v_c \tag{19}$$

where $v_c = [u_c, \upsilon_c, w_c]^T$ is the ocean current in a body-fixed frame. For the 2-D current model, we have:

$$\begin{aligned} u_c &= V_c \cos(\beta_c - \psi) \\ \upsilon_c &= V_c \sin(\beta_c - \psi) \end{aligned} \tag{20}$$

where β_c and V_c are the direction and speed of the ocean current. ψ is the AUV's heading. As a result, the following model can be used [43]:

$$\begin{aligned} M\dot{v} + C(v_r)v_r + D(v_r)v_r + g(\eta) &= \tau \\ \dot{\eta} &= J(\eta)v \end{aligned} \tag{21}$$

The case study is divided into three phases to assess the model's robustness to the effect of ocean currents. Each stage consists of two parts, including motion planning with or without the influence of disturbances.

A multi-target scenario has been used to properly evaluate the effect of ocean currents on the AUV motion planning system.

1) Phase 1: The AUV begins with an initial position of (-100, -100) and a heading angle of $\psi = 0$ (deg). The goal is to arrive safely at the second point (0,50). The first test was carried out in the absence of ocean currents. Figure 10 depicts the simulation outcome. A collision with an obstacle is indicated by fewer than 2.7 meters. The closest obstacle on this path is 3.68 meters away. The second experiment investigates motion planning with constant amplitude $V_c = 0.3$ (m/s) and direction $\beta_c = 120$ (deg) in the NED frame. In this case, the lowest distance from the obstacle is 4.2 meters.

2) Phase 2: The goal of the second phase is to reach points with coordinates (150, -50), (200, 150), (100, 220), and (250, 250) in a row. This stage is accomplished by reaching points (250 and 250). In this case, the minimum distance from the obstacle is 3.61 m. For evaluation of the ocean current influence at this stage, the direction and amplitude of the current at point (2) change instantaneously and are adjusted in $V_c = 0.2$ (m/s) and $\beta_c = 30$ (deg). The closest distance to the obstacle during this phase is 4.34 meters.

3) Phase 3: AUV moves to the ninth and tenth targets. The amplitude and direction of the ocean current change over time. The Gaussian model is used to simulate a stochastic change in the speed and direction of ocean currents. The ranges for amplitude and direction are $0.15 < V_c < 0.3$ (m/s) and $70 < \beta_c < 150$ (deg), respectively. The minimum distance from obstacles in this phase is 7.21 meters in the first part and 7.94 meters in the second part.

According to the results, the proposed system is resilient to ocean currents, and the AUV achieves various consecutive targets and avoids obstacles in the presence of ocean current disturbance. For further assessment of the proposed techniques, Table 2 collates three Key Performance Indicators (KPIs) for 20 different scenarios with and without the ocean currents to evaluate the performance of the proposed technique. Those KPIs include Travelled Distance (TD), Travel Time (TT) and Minimum Distance To Collision (MDTC). It should be noted that stochastic variation of $70 < \beta_c < 150$ (deg) and $0.15 < V_c < 0.3$

(m/s) is considered for ocean currents. Based on this table, the proposed algorithm has been able to manage different encounter scenarios.

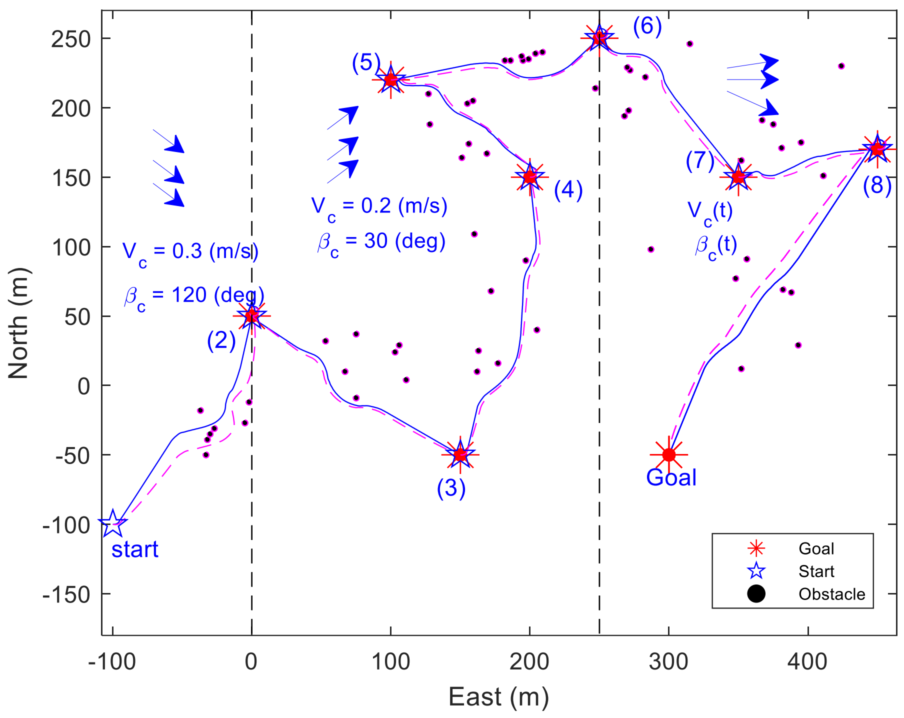

Fig. 10. The assigned task is divided into three phases: (1) and (2) path planning under the influence of a constant ocean current with different directions and speeds, and (3) path planning under the influence of a stochastic ocean current. Phase 2 begins immediately following the vehicle's passage through the second waypoint. Phase 3 begins immediately after the vehicle passes through the sixth waypoint. Solid blue line: AUV's trajectory without ocean current, pink dashed-line: AUV's trajectory with ocean current.

Table 2: The influence of ocean currents on AUV motion planning for different scenarios

Number of scenarios	Performance without disturbance			Performance with disturbance		
	TD (m)	TT (s)	MDTC (m)	TD (m)	TT (s)	MDTC (m)
One target point						
1	142.73	109.5	3.674	159.90	121.5	3.61
2	188.285	138.9	4.148	194.766	182.3	2.962
3	118.589	90.7	3.467	119.223	120.9	3.101
4	128.255	98.5	3.3	133.72	97.2	3.123
5	195.044	144.4	3.805	200.149	185.7	4.9267
6	215.67	160	11.205	278.340	223.7	3.964
7	187.742	140.8	4.954	188.542	140.1	8.442
8	238.422	175.1	5.186	230.76	144.5	3.612
9	245.18	180	7.654	243.177	158.9	6.63
10	291.493	211	4.977	292.8	196.8	3.15
Consecutive target points						
11	355.863	295	5.155	358.773	335.6	3.525

12	385.131	316.5	3.063	390.089	342.8	3.363
13	446.189	349.1	3.05	448.006	359.4	7.092
14	582.965	452.10	7.968	575.487	445.50	5.815
15	518.021	410.9	11.064	514.4	374.2	14.254
16	1829	1312	4.370	1841	1345	7.705
17	535.857	415.4	5.272	535.801	493.8	6.012
18	603.309	465.3	4.312	603.934	482.1	5.181
19	759.905	574.4	4.279	762.691	597.3	4.283
20	835.519	629.2	4.579	858.650	675.6	4.574

4.3.3 Reward function evaluation

This section is to illustrate the effect of fourth and fifth terms of the reward function (rewards related to the control signal), and the system's states for a typical scenario. Fig. 11 and Fig. 12 show the simulation results for AUV transfer from point (5,10) to point (30,5) by two separate intelligent agents trained with and without fourth and fifth terms of the reward, respectively. In both training processes, the saturation function is considered for the fin angle. In Fig. 11, the rudder signal is reduced, and we will not have a high oscillation.

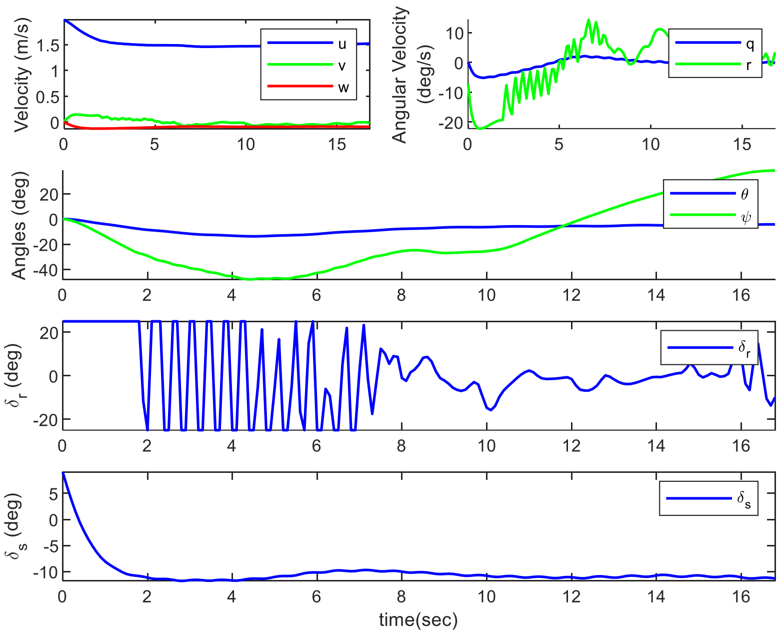

Fig. 11. The state spaces and the control effort for an agent without limitation terms in the rewards function

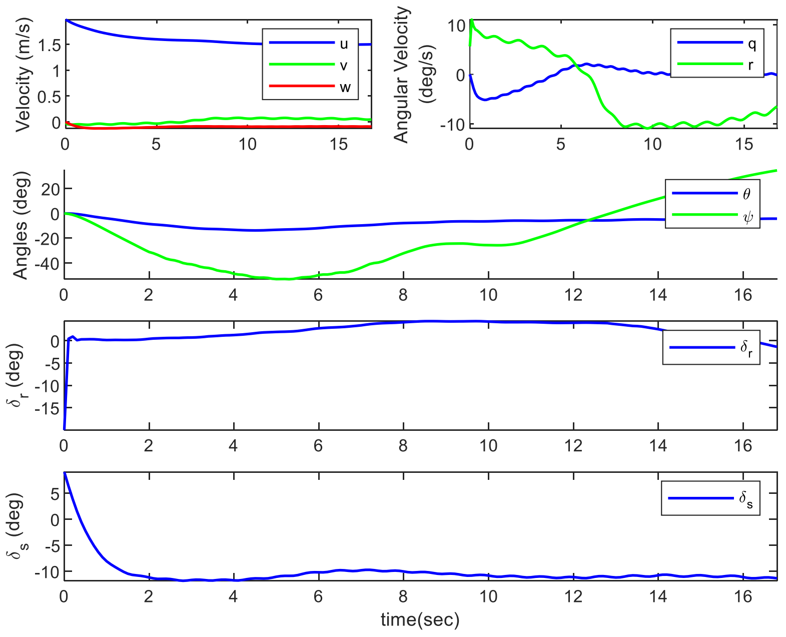

Fig. 12. The state spaces and the control effort for an agent with limitation terms (4 and 5 in section 3.1) in the rewards function

The simulation results show that the motion planning system based on deep reinforcement learning is efficient, adaptable, and resilient and can be applied directly to other unknown environments.

4.4. Discussion

Multiple tests in various environments were conducted to evaluate the performance and capability of the AUV's low-level adaptive motion planning system. Initially, the system's ability to reach the target at various azimuths was assessed. Avoiding the obstacles with random distribution was also shown. Then, motion planning on longer paths and under ocean currents was studied for a practical examination of the proposed method. For this purpose, the ocean current with an amplitude of 0.3 m/s and a direction of 120 degrees of NED frames was considered at the start of the AUV movement in the environment. After reaching the first goal, the ocean current suddenly changes to a speed of 0.2 m/s and a direction of 30 degrees. Finally, the system's performance in the presence of ocean currents was investigated using stochastic amplitude and direction variations. Experiments show that the proposed scheme has desirable robustness and adaptability in the presence of environmental disturbances and that it can carry out designed missions. The results of twenty different movement scenarios are presented in Table 2 in terms of time required to reach the target, distance traveled, and minimum distance from obstacles with and without ocean current. More experiments indicate increased time and distance traveled when ocean currents are present. Several tests demonstrate a decrease in the time required to reach the target due to the relative velocity used in the system's dynamic equation and the alignment of the AUV's movement direction with the direction of the ocean current. Then, the test was run by changing the reward for the control signal to demonstrate the importance of the reward function in the DRL method. As a result, reducing control signal fluctuation resulted in decreased power consumption and equipment costs. In future work, speed control will be used in addition to heading control to provide a more natural behavior for the AUV against obstacles. Furthermore, path planning in a 3D environment and the presence of dynamic obstacles can be considered.

5. Conclusion

The present article proposes a deep RL-based motion planning approach for an AUV, focusing on producing a short, safe, and energy-efficient feasible path. For this purpose, the motion planning issue is formulated as Markov processes, and the TD3 algorithm, which is a deterministic policy gradient algorithm for continuous action, is employed. The proposed approach only uses onboard sensor data to make decisions necessary to solve the continuous control commands effectively. Motion planning is executed without having any prior knowledge of the environment. The obstacles are identified by the AUV's range of sensors. Designing the reward function is an essential part of implementing reinforcement learning. It is carried out so that the AUV can produce a short, safe, and directional path towards the target while considering practical constraints such as energy consumption, actuator saturation, and a reduction of the control signal's sudden fluctuations. Unlike earlier articles in the field of AUVs, which are limited to AUVs with lower degrees of freedom or discrete structures, the present paper considers the non-linear model of an AUV by issuing low-level commands to the AUV's control fins. The system's efficiency is demonstrated through realistic simulation scenarios.

References

1. Fossen, T.I., *Handbook of marine craft hydrodynamics and motion control*. 2011: John Wiley & Sons.
2. Fossen, T.I. and S.I.J.J.o.R.S. Sagatun, *Adaptive control of nonlinear systems: A case study of underwater robotic systems.* Journal of Robotic Systems, 1991. **8**(3): p. 393-412.
3. Khodayari, M.H. and S. Balochian, *Modeling and control of autonomous underwater vehicle (AUV) in heading and depth attitude via self-adaptive fuzzy PID controller.* Journal of Marine Science and Technology, 2015. **20**(3): p. 559-578.
4. Sarhadi, P., A.R. Noei, and A. Khosravi, *Model reference adaptive PID control with anti-windup compensator for an autonomous underwater vehicle.* Robotics and Autonomous Systems, 2016. **83**: p. 87-93.
5. Sarhadi, P., A.R. Noei, and A. Khosravi, *Model reference adaptive autopilot with anti-windup compensator for an autonomous underwater vehicle: Design and hardware in the loop implementation results.* Applied Ocean Research, 2017. **62**: p. 27-36.
6. Cruz, N., *Autonomous underwater vehicles*. 2011: BoD–Books on Demand.
7. Hadi, B., A. Khosravi, and P. Sarhadi, *A Review of the Path Planning and Formation Control for Multiple Autonomous Underwater Vehicles.* J Journal of Intelligent and Robotic Systems, 2021. **101**(4): p. 1-26.
8. Garau, B., A. Alvarez, and G. Oliver. *Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an A* approach*. in *Proceedings of the 2005 IEEE international conference on robotics and automation, Barcelona, Spain*. 2005. Barcelona, Spain: IEEE.
9. Solari, F.J., et al. *Artificial potential fields for the obstacles avoidance system of an AUV using a mechanical scanning sonar*. in *2016 3rd IEEE/OES South American International Symposium on Oceanic Engineering (SAISOE), Buenos Aires, Argentina*. 2016. IEEE.
10. Carreras, M., et al., *Sparus II AUV—A hovering vehicle for seabed inspection.* IEEE Journal of Oceanic Engineering, 2018. **43**(2): p. 344-355.
11. Zhuang, Y., et al., *Efficient collision-free path planning for autonomous underwater vehicles in dynamic environments with a hybrid optimization algorithm.* Ocean Engineering, 2016. **127**: p. 190-199.

12. Li, D., P. Wang, and L. Du, *Path planning technologies for autonomous underwater vehicles-a review.* IEEE Access, 2018. **7**: p. 9745-9768.
13. Panda, M., et al., *A comprehensive review of path planning algorithms for autonomous underwater vehicles.* International Journal of Automation and Computing, 2020. **17**(3): p. 321-352.
14. Wani, M.A., et al., *Advances in deep learning*. 2020: Springer.
15. Goodfellow, I., Y. Bengio, and A. Courville, *Deep learning*. Vol. 1. 2016: MIT press.
16. Buşoniu, L., et al., *Reinforcement learning for control: Performance, stability, and deep approximators.* Annual Reviews in Control, 2018. **46**: p. 8-28.
17. Cui, R., et al., *Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning.* IEEE Transactions on Systems, Man, Cybernetics: Systems, 2017. **47**(6): p. 1019-1029.
18. Lewis, F.L., D. Vrabie, and K.G. Vamvoudakis, *Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers.* IEEE Control Systems Magazine, 2012. **32**(6): p. 76-105.
19. Khan, S.G., et al., *Reinforcement learning and optimal adaptive control: An overview and implementation examples.* Annual reviews in control, 2012. **36**(1): p. 42-59.
20. Gaskett, C., D. Wettergreen, and A. Zelinsky. *Reinforcement learning applied to the control of an autonomous underwater vehicle*. in *Proceedings of the Australian conference on robotics and automation (AuCRA99)*. 1999.
21. You, C., et al., *Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning.* Robotics and Autonomous Systems, 2019. **114**: p. 1-18.
22. Yoo, B. and J. Kim, *Path optimization for marine vehicles in ocean currents using reinforcement learning.* Journal of Marine Science and Technology, 2016. **21**(2): p. 334-343.
23. Blekas, K. and K. Vlachos, *RL-based path planning for an over-actuated floating vehicle under disturbances.* Robotics and Autonomous Systems, 2018. **101**: p. 93-102.
24. Bhopale, P., F. Kazi, and N. Singh, *Reinforcement learning based obstacle avoidance for autonomous underwater vehicle.* Journal of Marine Science Application, 2019. **18**(2): p. 228-238.
25. Sun, Y., et al., *Auv 3d path planning based on the improved hierarchical deep q network.* Journal of Marine Science and Engineering, 2020. **8**(2): p. 145.
26. Zhang, Q., et al., *Deep interactive reinforcement learning for path following of autonomous underwater vehicle.* IEEE Access, 2020. **8**: p. 24258-24268.
27. Zhao, L. and M.-I.J.O.E. Roh, *COLREGs-compliant multiship collision avoidance based on deep reinforcement learning.* Ocean Engineering, 2019. **191**: p. 106436.
28. Cheng, Y. and W.J.N. Zhang, *Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels.* Neurocomputing, 2018. **272**: p. 63-73.
29. Mnih, V., et al., *Human-level control through deep reinforcement learning.* nature, 2015. **518**(7540): p. 529-533.
30. Lillicrap, T.P., et al., *Continuous control with deep reinforcement learning.* arXiv preprint arXiv:.02971, 2015.
31. Anderlini, E., G.G. Parker, and G. Thomas, *Docking control of an autonomous underwater vehicle using reinforcement learning.* Applied Sciences, 2019. **9**(17): p. 3456.
32. Carlucho, I., et al. *AUV Position Tracking Control Using End-to-End Deep Reinforcement Learning*. in *OCEANS 2018 MTS/IEEE Charleston*. 2018. IEEE.
33. Sun, Y., et al., *Mapless motion planning system for an autonomous underwater vehicle using policy gradient-based deep reinforcement learning.* Journal of Intelligent and Robotic Systems, 2019. **96**(3-4): p. 591-601.

34. Havenstrøm, S.T., et al., *Deep Reinforcement Learning Controller for 3D Path Following and Collision Avoidance by Autonomous Underwater Vehicles.* Frontiers in Robotics, 2021. **7**: p. 211.
35. Guo, S., et al., *An autonomous path planning model for unmanned ships based on deep reinforcement learning.* Sensors, 2020. **20**(2): p. 426.
36. Sun, Y., et al., *A 2D Optimal Path Planning Algorithm for Autonomous Underwater Vehicle Driving in Unknown Underwater Canyons.* Journal of Marine Science and Application, 2021. **9**(3): p. 252.
37. Fujimoto, S., H. Hoof, and D. Meger. *Addressing function approximation error in actor-critic methods*. in *International Conference on Machine Learning*. 2018. PMLR.
38. Roberts, G.N. and R. Sutton, *Advances in unmanned marine vehicles*. Vol. 69. 2006: Iet.
39. Prestero, T.T.J., *Verification of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle*. 2001, Doctoral dissertation, Massachusetts institute of technology.
40. Sutton, R.S. and A.G. Barto, *Reinforcement learning: An introduction*. 2018: MIT press.
41. Fujimoto, S., H. Hoof, and D. Meger. *Addressing function approximation error in actor-critic methods*. in *International Conference on Machine Learning, pp. 1587-1596*. 2018. PMLR.
42. Sarhadi, P., A.R. Noei, and A.J.I.t. Khosravi, *Adaptive integral feedback controller for pitch and yaw channels of an AUV with actuator saturations.* ISA transactions, 2016. **65**: p. 284-295.
43. Fossen, T.I., *Guidance and control of ocean vehicles.* University of Trondheim, Norway, Printed by John Wiley & Sons, Chichester, England, ISBN: 0 471 94113 1, Doctors Thesis, 1999.

Appendix

Kinematics equations, indicating the relation between two frames, are as follows:

$$\dot{\eta}_1 = J_1(\eta_2)v_1,\ \dot{\eta}_2 = J_2(\eta_2)v_2, \tag{A.1}$$

$$J_1(\eta_2) = \begin{bmatrix} \cos\theta\cos\psi & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\phi\sin\psi\sin\theta & -\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix},$$

$$J_2(\eta_2) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix}$$

where $\eta_1 = \begin{bmatrix} x & y & z \end{bmatrix}^T$ and $\eta_2 = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ show the relative positions and angles in the earth-fixed frame. $v_1 = \begin{bmatrix} u & v & w \end{bmatrix}^T$ and $v_2 = \begin{bmatrix} p & q & r \end{bmatrix}^T$ show the linear and angular velocities in the body-fixed frame. "^{T}" denotes the transpose operator. The definitions of parameters are given in Table A1.

Table A1. Definition of AUV motion parameters

No.	Definitions		
	Reference frame	Position and Euler angles	Linear and angular rates
1	Motion along with x axis (surge)	x	u
2	Motion along with y axis (sway)	y	v
3	Motion along with z axis (heave)	z	w
4	Rotation around the x axis (roll)	φ	p
5	Rotation around the y axis (pitch)	θ	q
6	Rotation around the z axis (yaw)	ψ	r

Assuming that centers of gravity and buoyancy in three axes are as follows:

$$r_G = \begin{bmatrix} x_G \\ y_G \\ z_G \end{bmatrix}, \quad r_B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \tag{A.2}$$

And assuming that m is the vehicle mass and I_x, I_y and I_z are vehicle moments of inertia, the motion equations of vehicle can be obtained as follows:

$$m[\dot{u} - vr + wq - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] = -(W - B)\sin\theta + X_{u|u|}|u|u \tag{A.3}$$
$$+X_{\dot{u}}\dot{u} + X_{wq}wq + X_{qq}qq + X_{vr}vr + X_{rr}rr + X_{prop}.$$

$$m[\dot{v} - wp + ur - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})] = (W - B)\cos\theta\sin\phi + Y_{v|v|}|v|v$$
$$+Y_{r|r|}|r|r + Y_{\dot{v}}\dot{v} + Y_{\dot{r}}\dot{r} + Y_{ur}ur + Y_{wp}wp + Y_{uv}uv + Y_{uu\delta}u^2\delta_r.$$

$$m[\dot{w} - uq + vp - z_G(q^2 + p^2) + x_G(rp - \dot{q}) + y_G(rq + \dot{p})] = (W - B)\cos\theta\cos\phi + Z_{w|w|}|w|w$$
$$+Z_{q|q|}|q|q + Z_{\dot{w}}\dot{w} + Z_{\dot{q}}\dot{q} + Z_{uq}uq + Z_{vp}vp + Z_{rp}rp + Z_{uw}uw + Z_{uu\delta}u^2\delta_s.$$

$$I_x\dot{p} + (I_z - I_y)qr + m[y_G(\dot{w} - uq + vp) - z_G(\dot{v} - wp + ur)] = (y_GW - y_BB)\cos\theta\cos\phi$$
$$+(z_GW - z_BB)\cos\theta\sin\phi + K_{p|p|}|p|p + K_{\dot{p}}\dot{p} + K_{prop}.$$

$$I_y\dot{q} + (I_x - I_z)rp + m[z_G(\dot{u} - vr + wq) - x_G(\dot{w} - uq + vp)] = -(z_GW - z_BB)\sin\theta$$
$$-(x_GW - x_BB)\cos\theta\cos\phi + M_{w|w|}|w|w + M_{q|q|}|q|q + M_{\dot{w}}\dot{w} + M_{\dot{q}}\dot{q} + M_{uq}uq$$
$$+M_{vp}vp + M_{rp}rp + M_{uw}uw + M_{uu\delta}u^2\delta_s.$$

$$I_z\dot{r} + (I_y - I_x)pq + m[x_G(\dot{v} - wp + ur) - y_G(\dot{u} - vr + qw)] = (x_GW - x_BB)\cos\theta\sin\phi$$
$$+(y_GW - y_BB)\sin\theta + N_{v|v|}|v|v + N_{r|r|}|r|r + N_{\dot{v}}\dot{v} + N_{\dot{r}}\dot{r} + N_{ur}ur + N_{wp}wp$$
$$+N_{uv}uv + N_{pq}pq + N_{uu\delta}u^2\delta_r.$$

W and B show the vehicle weight and buoyancy respectively. These six nonlinear equations are known as six DOF equations and their inputs are δ_s, the elevator fins angle and δ_r, the rudder fins angle. Coefficients in these equations are defined in Tables A2-A3.

Table A2: Definition of hydrodynamic coefficients of AUV

Parameter	Value	Unit	Description
$X_{u\|u\|}$	-3.85	kg/m	Axial drag
$M_{w\|w\|}$	3.18	kg	Cross-flow drag
$N_{v\|v\|}$	-3.18		

$Y_{v\|v\|}$	-1310	kg / m	
$Z_{w\|w\|}$	-131		
$Y_{r\|r\|}$	0.63	$kg.m / rad^2$	
$Z_{q\|q\|}$	-0.63		
$M_{q\|q\|}$	-188	$kg.m^2 / rad^2$	
$N_{r\|r\|}$	-94		
$X_{\dot{u}}$	-0.93	kg	Added mass
$Y_{\dot{v}}, Z_{\dot{w}}$	-35.5		
$M_{\dot{w}}$	-1.93	kg / m	
$N_{\dot{v}}$	1.93		
$Y_{\dot{r}}$	1.93	$kg.m / rad$	
$Z_{\dot{q}}$	-1.93		
$K_{\dot{p}}$	-0.07	$kg.m^2 / rad$	
$M_{\dot{q}}, N_{\dot{r}}$	-4.88		
X_{wq}, Z_{vp}	-35.5	kg / rad	Added mass cross-term
X_{vr}, Y_{wp}	35.5		
Z_{rp}	1.93		
$X_{q\|q\|}, X_{r\|r\|}, M_{vp}, N_{wp}$	-1.93	$kg.m / rad$	
Y_{pq}	1.93		
M_{rp}	4.86	$kg.m^2 / rad^2$	
N_{pq}	-4.86		
Y_{ur}	5.22	kg / rad	Added mass cross term and fin lift
Z_{uq}	-5.22		
N_{ur}, M_{uq}	-2	$kg.m / rad$	
$K_{p\|p\|}$	-1.3	$Kg.m^2 / rad^2$	Rolling resistance
M_{uw}	24	kg	Body and fin lift and Munk moment
N_{uv}	-24		
Y_{uv}, Z_{uw}	-28.6	kg / m	Body lift force and fin lift
$Y_{uu\delta}$	9.64	kg	Fin lift force
$Z_{uu\delta}$	-9.64		
$N_{uu\delta}, M_{uu\delta}$	-6.15	kg / rad	
X_{prop}	9.25	N	Propeller trust and torque
K_{prop}	-0.54	$N.m$	

Table A3: non-hydrodynamic parameters of AUV

Parameter	**Value**	**Unit**	**Description**
g	9.8	m/s^2	Gravity constant
ρ	1030	kg/m^3	Seawater density
m	30.5	kg	Vehicle mass
W	299	N	Vehicle weight
B	308	N	Vehicle buoyancy
I_x	0.17	$kg.m^2$	Vehicle moment of inertia around x axis
I_y, I_z	3.45	$kg.m^2$	Vehicle moment of inertia around y and z axis
x_G, y_G	0	M	Center of gravity in x and y directions with respect to the center of buoyancy
z_G	0.0196	M	Center of gravity in z direction with respect to the center of buoyancy
x_B, ,y_B, z_B	0	M	Center of buoyancy is given as the origin of the body-fixed coordinate