Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Development and application of the coverage path planning based on a biomimetic robotic fish

Jincun Liu[1,2,3,4] | **Jian Zhao**[1,2,3,4] | **Zhenna Liu**[5] | **Yang Liu**[1,2,3,4] | **Yinjie Ren**[1,2,3,4] | **Dong An**[1,2,3,4] | **Yaoguang Wei**[1,2,3,4]

[1]National Innovation Center for Digital Fishery, China Agricultural University, Beijing, China

[2]Key Laboratory of Smart Farming Technologies for Aquatic Animals and Livestoc, Ministry of Agriculture and Rural Affairs, China Agricultural University, Beijing, China

[3]Beijing Engineering and Technology Research Centre for Internet of Things in Agriculture, China Agricultural University, Beijing, China

[4]College of Information and Electrical Engineering, China Agricultural University, Beijing, China

[5]Shandong Labor Vocational and Technical College, Shandong Labor Vocational and Technical College, Jinan, China

Correspondence

Corresponding author Jincun Liu and Yaoguang Wei, National Innovation Center for Digital Fishery, China Agricultural University, Beijing, China.
Email: liujincun@cau.edu.cn, wyg@cau.edu.cn

Present address

17 Qinghua East Road, Haidian District, Beijing.

Abstract

This paper studies the coverage path planning and path following problems for an underwater biomimetic robotic fish to finish the deep-sea net cage water quality monitoring. Firstly, with a focus on minimizing total path length, repetition rate, and turning occurrences to enhance path coverage efficiency and reduce energy consumption, we propose a novel coverage path planning strategy. This strategy incorporates DQN, a reward function, and a rewrite strategy inspired by RRT*. Secondly, a high-performance path-following method, which takes into account robot performance, is designed to cope with adverse conditions in net cages. Finally, the simulation and field experiments demonstrate significant improvements over existing methods and the effectiveness in practical applications, showcasing its applicability in aquaculture management. The proposed algorithm offers valuable insights into optimizing coverage path planning for underwater robots in practical scenarios.

KEYWORDS

coverage path planning, DQN, robotic fish, deep-sea net cage

1 | INTRODUCTION

Deep-sea net cage culture is becoming a new type eco culturing trend in the future for its pollution-free and saving feeding costs. Nevertheless, ensuring water quality safety within these net cages requires regular manual monitoring. The geographical challenge arises as these cages are typically located tens of kilometers offshore, complicating personnel transportation. Moreover, the sampling areas, positioned ten meters above the water surface, are susceptible to frequent wind and waves, adding risks to manual water quality assessments. The conventional fixed water quality monitoring methods encounter obstacles, including deployment difficulties and the need for periodic sensor replacement. In light of these challenges, there is a critical need to implement comprehensive coverage path planning (CCPP) and detection strategies for the surface of deep-sea net cages, utilizing Unmanned Underwater Vehicles(UUVs).

Coverage path planning (CPP) is instrumental in crafting an energy-efficient and comprehensive path that systematically covers a designated area, strategically circumventing potential obstacles. The core objectives of CPP are centered around determining the optimal coverage path. This entails the minimization of critical factors such as travel time, processing speed, energy costs, path length, and the number of turns, all while concurrently reducing repetition. The outcome is the generation of

Abbreviations: ANA, anti-nuclear antibodies; APC, antigen-presenting cells; IRF, interferon regulatory factor.

collision-free trajectories, enhancing the efficiency and precision of path planning in technological applications(Tan, Mohd-Mokhtar, & Arshad, 2021). With the rapid development in the field of robotics, the application of coverage path planning methods has gradually expanded, encompassing areas such as cleaning robots (Le, Veerajagadheswar, Thiha Kyaw, Elara, & Nhan, 2021; Miao, Lee, & Kang, 2020; Van Pham, Asadi, Abut, & Kandilli, 2019; Wijegunawardana et al., n.d.), painting robots (Zhou et al., 2022), unmanned underwater vehicles (D. Zhu, Tian, Sun, & Luo, 2019), lawn mowing robots (Maini, Gonultas, & Isler, 2022; Bai, 2022; Mitschke, Uchiyama, & Sawodny, 2018), harvesters (G. Zhang et al., 2022; Nørremark, Nilsson, & Sørensen, 2022; Wagner, Kirk, Hanheide, & Cielniak, 2021), picking robots (Wang et al., 2023), and more. Owing to the extensive applications of coverage path planning and its considerable potential for enhancing the efficiency of robotics, it has drawn a great deal of interest from the robotics.

Due to the impact of unpredictable factors of the deep-sea net cage , such as wind and waves, the water surface environment is often subject to fluctuations, affecting the performance of UUVs. The frequent execution of turns can result in a reduction in UUVs speed, potentially causing significant deviations from the pre-established route. Concurrently, UUVs encounter constraints in endurance, and the frequent cycles of accelerating and decelerating towards waypoints can significantly deplete their energy resources. This renders such a strategy unsuitable for scenarios characterized by a substantial number of waypoints along the designated path. Therefore, in the coverage path planning algorithm for UUVs, it is not only necessary to ensure coverage of all reachable areas but also to optimize the planned path to reduce the frequency of turns and the number of waypoints, thereby decreasing the energy consumption of the UUV (Huang, Xu, Shi, & Liu, 2022).

Researchers have put forth a variety of specialized CPP methods to effectively mitigate the challenge of minimizing the number of turns and waypoints for robotic systems.

Lu, Zeng, Tang, Lam, and Wen (2023) form Guangdong University of Technology introduced a Turn-minimizing Multirobot Spanning Tree Coverage Star (TMSTC*) algorithm, the global map was divided into minimal brick-like branches, and a greedy strategy was employed to interconnect these bricks, creating a tree structure with the objective of minimizing the number of turns along the resulting circumnavigating coverage path. Experimental results demonstrated that the proposed method effectively decreased the number of robot turns, leading to a more efficient completion of coverage tasks. Ramesh, Imeson, Fidan, and Smith (2022) utilzed a linear program to partition the environment into thin axis-parallel ranks, aiming to minimize turns in robot environment coverage. The algorithm demonstrated a noteworthy 6% reduction in turns and an average of 3% shorter coverage tours compared to an alternative method. However, it is important to note that this approach may generate “staircase” paths in narrow areas with non-axis-parallel or curved boundaries. Kim and Kim (2010) implemented a minimum-time grid coverage trajectory planning (GCTP) algorithm to find the minimum time grid coverage trajectory. They also introduced a minimum time turning trajectory planning algorithm with efficient iterative binary search algorithm. Simulation results indicated a 16% reduction in the robot’s grid coverage time. From Southeast University, Cao, Cheng, and Mu (2022) elaborated an enhanced Probabilistic Roadmap (PRM) algorithm, incorporating a back-and-forth mode and imposing constraints on the number of turns. Extensive multi-group simulations substantiated the viability of the proposed algorithm in achieving concentrated coverage for aerial photography. Parameters such as path length, number of turns, computation time, coverage rate, and repetition rate were considered. It was crucial to underscore that the aforementioned methods were grounded in conventional coverage path planning approaches, involving relatively higher time costs and being suitable for application within smaller environmental maps.

Meanwhile, reinforcement learning(RL) have gained increasing scientific attention in the field of coverage path planning. Serving as a model-free control policy, RL possesses the capability to learn adaptive planning and decision-making to maximize a numerical reward signal through trial-and-error search and delayed rewards within an unknown environment. Following an extensive developmental period, RL has found successful applications in robotic control (Luo, Tian, Li, Chen, and Tan (2024); Yu et al. (2023); Masmitja et al. (2023);Gan, Huo, and Li (2023)), emerging as a novel avenue for enhancing coverage path planning. As early as 2019, Shakeri et al. (2019) had demonstrated the potential of reinforcement learning in path planning. Piardi, Lima, Pereira, and Costa (2019) interpreted an optimization approach for coverage path planning employing the Q-Learning algorithm. This algorithm effectively reduced the number of waypoints in comparison to genetic algorithms. A new-type reward functions originating from Predator-Prey model into traditional Q-learning based CPP solutions was designed. This model could reduce repetition rates and the number of turns in coverage path planning (M. Zhang, Cai, and Pang (2023); Hassan and Liu (2019)).Luis, Reina, and Marín (2020)evaluated two deep reinforcement learning algorithms for the effective execution of cruising tasks. These studies indicated that an increasing number of researchers believe that adopting reinforcement learning can better address the issues of coverage path planning and the reduction of turns and waypoints for robots.

However, the table-based Q-learning methods discussed earlier are applicable in scenarios where the state space and action space of the problem are relatively small. In the context of coverage path planning for deep-sea net cage inspection, especially

in intricate maritime environments, the challenge involves achieving thorough inspection coverage within a designated area without explicitly defined target points. It is essential to note that the complexity of underwater robot states and actions further complicates the state space and action space in this context.

Given the intricate network architecture and robust generalization capabilities, deep reinforcement learning can adeptly manage high-dimensional state and action spaces. This proficiency presents a promising solution to address the issue at hand. In this paper, a coverage path planning algorithm based on deep Q-network (DQN) is proposed and applied for the deep-sea net cage complete coverage inspection based on a biomimetic robotic fish. Various reward mechanisms are proposed in this algorithm, which including dynamic step size reward function, predation avoidance reward function, smoothness reward function and boundary reward function. These reward functions will endow the robot to choose larger steps in spatial order to reduce the number of turns and complete coverage path planning. Simultaneously, inspired by the Rrapidly-exploring Random Tree (RRT) rewrite strategy, a rewrite strategy is employed to replan path points, reducing the number of points on the path and consequently decreasing the energy consumption of the robot.

The main contributions of this paper can be summarized in the following three aspects:

- A DQN-based coverage path planning algorithm, specifically suitable for the robotic fish, is introduced in this paper. The algorithm takes into account both kinematic constraints and concurrent coverage achievement, meeting other specific practical requirements.
- A rewriting strategy inspired by RRT* is presented in this paper for the replanning of waypoints along the generated coverage path. This strategy effectively reduces the number of waypoints, resulting in a lowered energy consumption for the robot.
- Field experiments with untethered robotic fish in deep-sea net cages validate the efficacy of the coverage path planning and tracking methods. The collection of water quality data during these experiments serves to substantiate the practical applicability of the proposed approach. The resulting experimental outcomes robustly endorse the practical application and feasibility of the algorithm.

The remaining sections of this paper are organized as follows: Section 2 provides a concise overview of the mechatronic design of a free-swimming robotic fish and its swimming performance. The proposed DQN-based coverage path planning algorithm is presented in Section 3, along with ongoing research on waypoint tracking within the Section 4. Section 5 exhibits simulated comparative results, evaluating five reinforcement learning coverage path planning methods in a simulation environment. Field experiments on a deep-sea net cage are implemented in Section 6. Discussion and Concluding remarks are provided in Sections 6 and 7 respectively.

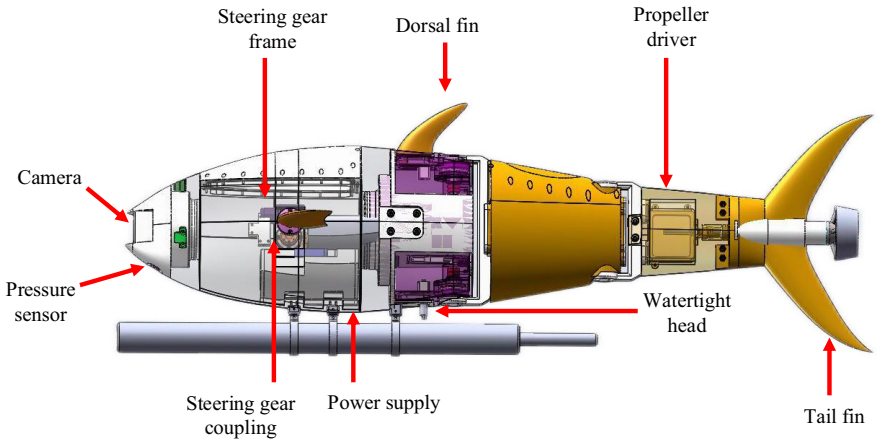
2 | OVERVIEW OF THE BIOMIMETIC ROBOTIC FISH

2.1 | Robotic fish prototype

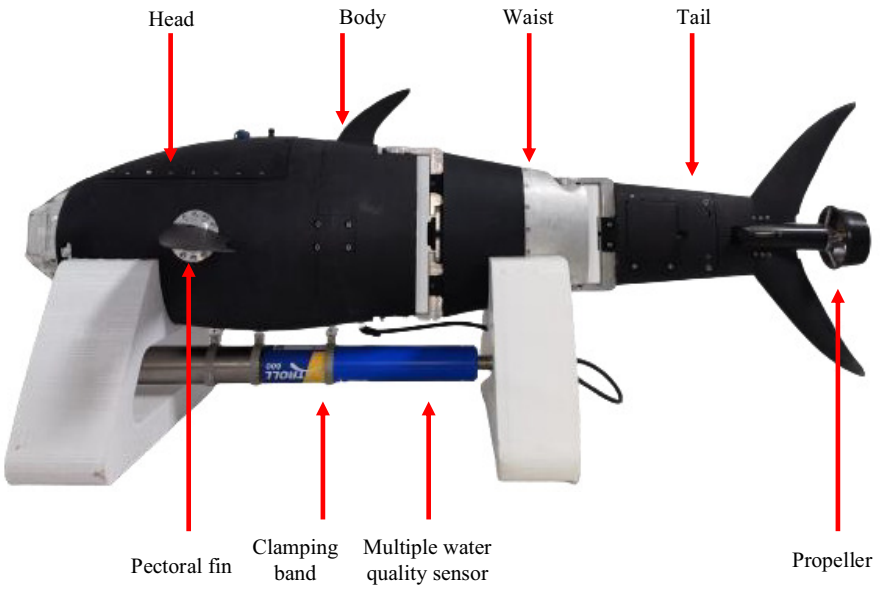
Conventional underwater vehicles are propelled by rotary propellers. Notwithstanding their operational simplicity, thrusters face several challenges, such as poor maneuverability, low energy efficiency, significant environmental disturbance, poor concealment, and the potential to harm underwater life. Fish propel themselves through undulatory motion, demonstrating impressive maneuverability and high energy efficiency in their swimming. Drawing inspiration from the undulatory fins of live fish, researchers and engineers are devoted to the innovative design of propulsors for underwater swimming robots(Xie et al. (2023); J. Zhu et al. (2019); P. Zhang, Wu, Chen, Tan, and Yu (2023); Yan et al. (2023)). The undulation or oscillation robots enable them to have high swimming efficiency, smaller turning radius, and less disturbance to the surrounding lives. Nevertheless, conventional bionic body designs often encounter challenges such as slow swimming speed and limited resistance to waves.

The hybrid-driven biomimetic robotic fish incorporates the traditional propeller-driven method into the bionic design of the robotic fish, allowing for compatibility and synergy between the two approaches. The detailed proof-of-concept prototype can be referenced from our earlier research(Ji, Wei, Liu, & An, 2023). Figure 1 illustrates the designed untethered robotic fish, consists of a well streamlined tuna-like body, waist and tail swinging joints with a tail fin, a pair of 2 degree of freedom (DOF) mechanical fins, a fixed doral fin, and a replaceable propeller. The robot is equipped with sensors including a GPS sensor for measuring geographical location, an Inertial Navigation System (INS) for providing posture information, a depth sensor for submersion depth, and a water quality sensor for collecting water quality parameters. These sensors provide the foundation

for completing comprehensive water quality monitoring tasks, covering complex water surfaces. The more relevant technical parameters are presented in Table 1.



(a) The SolidWorks drawing of the robot



(b) Robotic prototype

FIGURE 1 Mechanical structure and prototype of the hybrid-driven biomimetic robotic fish.

2.2 | Performance of the free-swimming robot

The hybrid-driven biomimetic robotic fish is equipped with a pair of bilaterally symmetrical pectoral fins, each having one DOF capable of performing biased or oscillatory flapping. The waist and tail joints also have one DOF, allowing for the adjustment of the oscillation frequency during reciprocal swinging and biasing to specific angles. The tail fin of the biomimetic robotic fish is equipped with a propeller, providing propulsion when the robotic fish is swimming. By virtue of these characteristics, the robot can expediently perform multimodal swimming gaits involving fast forward swiming, fish-like body and/or caudal fin (BCF) and median and/or paired fin (MPF) swimming, backward swimming, submersion, surfacing, and turning. The swimming performance indicators of the hybrid-driven biomimetic robotic fish are presented in Table 2.

TABLE 1 Technical parameters of the hybrid-driven biomimetic robotic fish

Items	Characteristics
Size ($L \times W \times H$)	1.15 m × 0.562 m × 0.393 m
Total mass	19.5 kg
Controller	STM32F407
Power supply	DC 24 V
Drive mode	4
Waist joint motor	maxon EC 60 200 W
Tail joint motor	maxon EC 22 100 W
Propeller	whale 98 W and 120 W
Pectoral fin motor	Dynamixel XM540-W270
Communication module	E62-433T20D (433 MHz)
On-board sensors	Pressure sensor, inertial sensor, GPS, Multiple water quality sensor

TABLE 2 Swimming performance metrics of the hybrid-driven biomimetic robotic fish

Motion Mode		Value
Forward swimming speed (m/s)	MPF	0.058
	BCF	0.144
	Propellers	1.16
Turning radius (m)	Turning driven by the pectoral fins	1.06
	Turning coordinated with the waist and tail joints by the propellers	0.53
	Turning coordinated with the pectoral fins, waist, and tail joints by the propellers	0.49
Turning Speed (°/s)	Turning driven by the pectoral fins	2.07
	Turning coordinated with the waist and tail joints by the propellers	78.6
	Turning coordinated with the pectoral fins, waist, and tail joints by the propellers	70.69

There are three forward swimming gaits for the robotic fish: the MPF mode through swinging the pectoral fins, the BCF mode involving oscillations of the waist and tail joints, and the thruster mode achieved through the propulsion force of the tail propeller. Through experiments, the maximum swimming speed is about 1.16 m/s in the propeller full-speed mode, equivalent to 1.1 body lengths per second (BL/s).

The hybrid-driven biomimetic robotic fish features three turning gaits: turning driven by the pectoral fins, turning coordinated with the waist and tail joints by the propeller, and turning coordinated with the pectoral fins, waist, and tail joints by the propeller. Through experiments, the turning speed of the pectoral fin-driven turning is 2.07°/s, which is relatively slow and only suitable for fine-tuning the swimming angle of the robotic fish but not ideal for maritime environments with wind and waves. The latter two turning methods have higher turning speeds, but they result in lateral drift during turning, deviating from the predetermined route and affecting the stability of data collection.

The method used by the hybrid-driven biomimetic robotic fish for ascending and descending relies on the bias of the pectoral fins, combined with the thrust provided by the propeller to accomplish. Its diving speed can acheve about 0.19 m/s. The emergency stop of the hybrid-driven biomimetic robotic fish involves halting the movement of the propeller while providing resistance through the biased pectoral fins. Frequent emergency stops during the traversal of waypoints in water quality monitoring tasks can significantly deplete the energy resources of the hybrid-driven biomimetic robotic fish.

In summary, the hybrid-driven biomimetic robotic fish is not suitable for tracking water quality monitoring paths with frequent turns and numerous waypoints when conducting comprehensive water quality monitoring tasks covering complex water surfaces.

3 | DQN-BASED COVERAGE PATH PLANNING ALGORITHM

Frequent turning often of the robot always leads to the existence of small dead zones, which can impact water quality data collection. Excessive waypoints often lead to frequent acceleration and deceleration, resulting in energy consumption. Therefore, in this section, with the objective of minimizing the number of turns and waypoints in the coverage path, a dynamic step size and predator-prey reward based deep Q-network (DSS-PPDQN) coverage path planning algorithm is proposed. The algorithm introduces several reward mechanisms, including a dynamic step size reward function, predation avoidance reward function,

smoothness reward function, and boundary reward function. These mechanisms guide the robot in choosing larger steps in spatial order to minimize the number of turns, avoid obstacles, and complete the coverage path planning. Additionally, a rewrite strategy inspired by RRT* is employed to replan waypoints, reducing the number of waypoints and thereby lowering the energy consumption of the robot.

3.1 | Problem formulation

The coverage path planning problem for the biomimetic robotic fish can be treated as a Markov Decision Process (MDP). A typical MDP consists of a state space, action space, reward function, and state transition function. In the realm of coverage path planning, the biomimetic robotic fish acts as an agent interacting with the environment. Now, each component of the Markov Decision Process applied to the coverage path planning problem for underwater bionic robots will be explained one by one.

3.2 | State space

At each moment, the water surface environment assumes a state, representing a summary of the current water surface conditions. The state space encompasses the set of all possible states. The algorithm models the environment using a grid map, dividing the robot's water surface working area into uniformly sized grids and converting it into a two-dimensional matrix. In this grid matrix, "0" signifies grids not covered by the robot, "-1" denotes grids with obstacles, "1" represents covered grids, and "2" indicates the robot's current position. In this algorithm, the state space comprises the collection of all possible two-dimensional matrices representing environments. An example of a state space grid map transformation is illustrated in Figure 2.

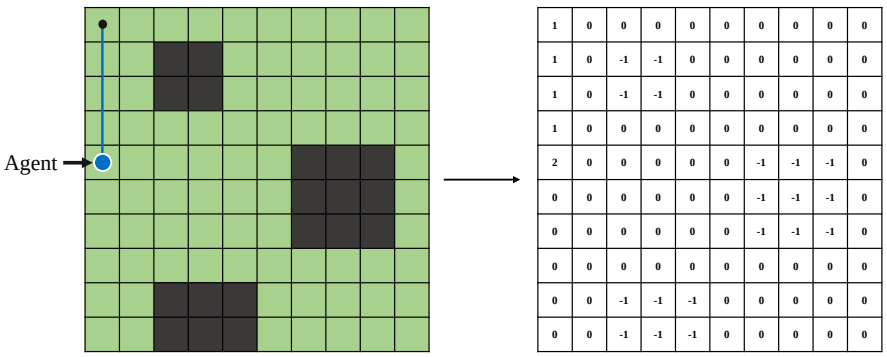


FIGURE 2 An example grid map transformation for the state space.

3.3 | Action space

Based on the current state, the robot selects specific actions. Reference to Section 2,the action space encompasses not only forward and backward movements and turning but also the step size that can be taken within the current action. Reference to Section 2.2, the robot's actions are subject to various constraints, such as turning radius and swimming speed. It is worth noting that, when the robot is in a certain state, the action selected through the policy function includes not only the direction of movement but also the step size in that direction.

In this problem, the reachable step size is defined as the maximum dimension of the column space of the action's two-dimensional matrix, denoted as M . Therefore, the action space of this algorithm can be represented as:

$$A = \left\{ \begin{array}{l} forward\ 1, backward\ 1, turnleft\ 1, turnright\ 1, \\ forward\ 2, backward\ 2, turnleft\ 2, turnright\ 2, \\ ..., \\ forward\ M, backward\ M, turnleft\ M, turnright\ M \end{array} \right\}$$

3.4 | Reward

In the context of reinforcement learning, a reward is a numerical value provided by the environment to the robot upon executing a specific action. It serves as a quantitative feedback mechanism, indicating the performance of the action with respect to the overall task objective. In this algorithm, reward functions encompass dynamic step size, predation avoidance, smoothness, and boundary.

The algorithm defines the region of attractors as the farthest positions in all feasible directions that the agent can move in a single decision. During this process, the dynamic step size reward is defined using the distance from the robot to the attractor to encourage the generation of larger step sizes. To make the generated coverage path more orderly, reducing the total path length of coverage path planning, additional rewards are introduced on the basis of dynamic step size reward, including predation avoidance reward, smoothness reward and boundary reward, representing the robot's incentives for avoiding static predators, planning smooth paths, and planning paths along the boundary, respectively.

At this point, O is defined as the set of all regions in the environment, k represents the index of the region where the agent is currently located, O_k is the current region of the agent, and j represents the index of uncovered and obstacle-free neighboring regions of the current region where the agent is located.

3.4.1 | Dynamic step size reward function

To minimize the number of waypoints traversed by the robot, saving decision time, minimizing turns, and reducing overall coverage time, the robot is encouraged to select the maximum step size during each decision. The region of attractors is defined as the farthest position the robot can reach along all feasible directions in a single decision. The attractor region and dynamic step size are illustrated in Figure 3.

In this process, the reward function is defined based on the distance from the robot to the attractor, referred to as the dynamic step size reward function, to encourage the generation of paths with larger step sizes. The reward formula is as follows:

$$R^{\mathrm{L}}\left(\mathbf{o}_j\right) = \frac{\mathrm{L}\left(\mathbf{o}_j\right) - \mathrm{L}_{\min}\left(\mathbf{o}_k\right)}{\mathrm{L}_{\max}\left(\mathbf{o}_k\right) - \mathrm{L}_{\min}\left(\mathbf{o}_k\right)}$$

where, $\mathrm{L}\left(\mathbf{o}_j\right)$ represents the distance from $\mathbf{o}_j$ to the attractor, $\mathrm{L}_{\max}\left(\mathbf{o}_k\right)$ is the maximum distance from an agent's neighbor to the attractor, and similarly, $\mathrm{L}_{\min}\left(\mathbf{o}_k\right)$ defines the minimum distance from an agent's neighbor to the attractor. Based on the above formula, the range of rewards that the agent can receive is [0, 1].

3.4.2 | Predation avoidance reward function

Inspired by the Pedator-Prey model, the prey maximizes its reward at each step by moving towards the uncovered neighbor farthest from the predator. In this model (Zhang et al., 2023; Hassan, 2019), the predator is defined as a stationary virtual point located outside the coverage area, while the prey is defined as the robot, as shown in Figure 3. The predator provides spatial order to the robot's (prey's) movement, assisting the robot in planning shorter coverage paths. The foraging behavior of the robot (prey) initially guides it to the region farthest from the predator, but eventually, the robot must gradually approach the predator to explore new uncovered areas. Therefore, the robot needs to search throughout the entire environment to achieve complete coverage. Positions farther from the predator receive higher rewards, and the reward formula is as follows:

$$R^{d}\left(\mathbf{o}_j\right) = \frac{D\left(\mathbf{o}_j\right) - D_{\min}\left(\mathbf{o}_k\right)}{D_{\max}\left(\mathbf{o}_k\right) - D_{\min}\left(\mathbf{o}_k\right)}$$

where, $D\left(\mathbf{o}_j\right)$ represents the distance from $\mathbf{o}_j$ to the predator ψ, $D_{\max}\left(\mathbf{o}_k\right)$ represents the maximum distance from a neighbor of the current prey target to the predator, and similarly, $D_{\min}\left(\mathbf{o}_k\right)$ represents the minimum distance from a neighbor of the current prey target to the predator. Based on the above formula, the prey receives rewards within the range of [0, 1].

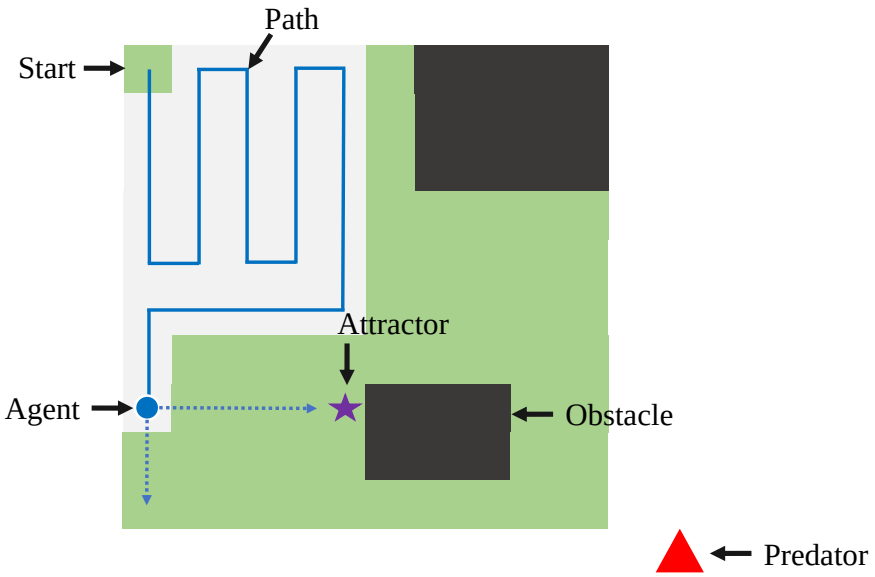


FIGURE 3 The illustration of the attractor region and dynamic step size.

3.4.3 | Smoothness reward function

In the context of coverage path planning scenarios, it is advantageous for the robot to exhibit consistent straight-line movement to minimize energy and time consumption associated with frequent turns. Consequently, a reward mechanism for continuous straight-line motion is expressed as following:

$$R^s\left(\mathbf{o}_j\right) = \frac{\angle \mathbf{o}_{k-1}\mathbf{o}_k\mathbf{o}_j}{180^\circ}$$

where, based on the above formula, the prey receives rewards within the range of [0, 1]. $\angle \mathbf{o}_{k-1}\mathbf{o}_k\mathbf{o}_j \in (0^\circ, 180^\circ]$ represents the angle between two vectors $(\mathbf{o}_{k-1} - \mathbf{o}_k)$ and $(\mathbf{o}_k - \mathbf{o}_j)$.

3.4.4 | Boundary reward function

The boundary is defined here as the boundary of the environmental space and the boundary of the uncovered area. To encourage the agent to prioritize planning paths along the boundary, we have introduced a boundary reward function, and its formula is as follows:

The boundary is defined as both the boundary of the environmental space and the boundary of the uncovered area. In order to incentivize the agent to prioritize planning paths along the boundary, a boundary reward function is represented by the following:

$$R^b\left(\mathbf{o}_j\right) = \frac{n^{N_{\max}} - n^N\left(\mathbf{o}_j\right)}{n^{N_{\max}}}$$

where, $n^N\left(\mathbf{o}_j\right)$ calculates the number of uncovered neighbors of $\mathbf{o}_j$. $n^{N_{\max}}$ is the maximum possible number of neighbors for the agent. Based on the above formula, the agent receives rewards within the range of [0, 1].

3.4.5 | Total reward function

The total reward for moving to the uncovered neighbor $\mathbf{o}_j$ is the weighted sum of all previously stated rewards.

$$R\left(\mathbf{o}_j\right) = \omega^{\mathrm{L}}\left(R^{\mathrm{L}}\left(\mathbf{o}_j\right)\right) + \omega^d\left(R^d\left(\mathbf{o}_j\right)\right) + \omega^s\left(R^s\left(\mathbf{o}_j\right)\right) + \omega^b\left(R^b\left(\mathbf{o}_j\right)\right)$$

where, ω^d, ω^s, ω^L, and ω^b are the weighting factors for the four reward functions. These factors will influence the weights of each reward in the total reward, thereby affecting the planning of the coverage path.

3.5 | State transition

The state transition function describes the probability of the robot transitioning to a new state after taking a certain action. In this algorithm, since the robot has GPS sensor localization capability and does not consider the underwater environmental factors, such as ocean currents, water temperature, water quality, etc., the state transition function may be influenced, affecting the robot's position and state. Therefore, the state transition in this algorithm is deterministic.

3.6 | Main process

Algorithm 1 DQN based CPP Problem

while not over **do** ▷ DQN based CPP
  s_i= {map$_i$} and preprocessed sequence s_i=$\phi(s_i)$
  if encounter dead zone **then**
    s_i= BFS(s_i)
  else
    With probability ε select a random action a_i
    otherwise select a_i= argmax$_a Q(s_i, a; \theta)$
    Get map$_{i+1}, a_i, r_i, over$
  end if
end while
Initialize replay memory D to capacity N ▷ DQN Training
Initialize action-value function Q with random weights θ
Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$
for episode,i **do**
  Execute DQN based CPP N times Update memory D
  for t,T **do**
    Sampling random data in D and obtain N pieces of data for $s_{\text{now}}, a_{now}, r_{now}, s_{\text{next}}, over$
    Set $y_{now} = \begin{cases} r_{now} \text{ifoverisTrue} \\ r_{now} + \gamma \max_{a'} \hat{Q}(s_{next}, a', \theta^{-}) \text{otherwise} \end{cases}$
    Perform a gradient descent step on $(y_{now} - Q(s_{now}, a_{now}, \theta))^2$ with respect to the network parameters θ
    Every C steps reset $\theta^{-} \leftarrow \theta$
  end for
end for
Obtain pathway points (x_1,x_2,...x_n) based on the map ▷ rewrite strategy inspired by RRT*
for i,N **do**
  for j=i+1,N **do**
    if not $\overrightarrow{x_i x_j} = \lambda \overrightarrow{x_i x_{i+1}}$ **then**
      $i = j$, add x_j to the final path
    end if
  end for
end for

Furthermore, DQN is employed to address the coverage path planning problem for the robot, allowing the handling of larger and continuous state spaces. The input is a two-dimensional matrix representing the current state, and the neural network of the DQN fits the Q-value table. The output not only determines the current action direction but also influences the selection of the current step length, aiming to minimize the robot's number of turns. Additionally, the algorithm introduces an experience replay

pool to address the issue of high correlation between samples. Furthermore, a target network is incorporated to handle the TD bias in the temporal difference algorithm. The main process is detailed in Algorithm 1 1 using pseudo-code.

The specific formula for updating the neural network parameters is given by the following (Algorithm 1, line 19):

$$Loss = (r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-) - Q(s_t, a_t; \theta))^2$$

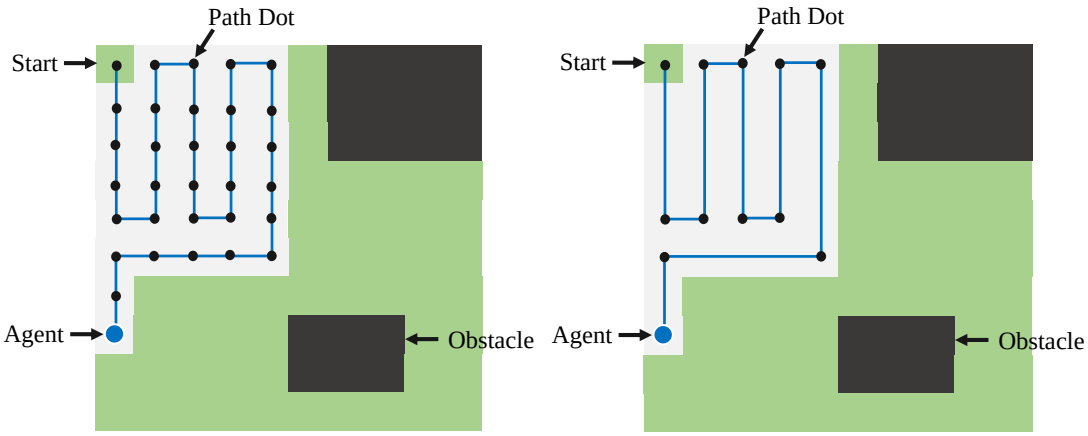
where, s_t represents the state at time t, r is the reward at time t, γ is the discount factor, $Q(s, a; \theta)$ is the value output by the current network (Q-eval) for determining the optimal action in the current state. The $Q(s\prime, a\prime; \theta^-)$ elaborates the output results of the target network (Q-target). Therefore, after the agent takes an action, the parameters of Q-eval can be updated based on the above equation. After a certain number of iterations, the parameters of Q-eval are copied to Q-target, completing one learning iteration (Algorithm 1, line 20).

$$\theta^-_{new} \leftarrow \tau\theta_{new} + (1 - \tau)\theta^-_{now}$$

where, τ is a hyperparameter with a range of (0, 1). θ^-_{now} represents the current parameters of the target network (Q-target), θ_{new} interprets the updated parameters of the current network (Q-eval), and θ^-_{new} depicts the updated parameters of the target network (Q-target) after the update.

3.7 | Rewrite strategy inspired by RRT*

Due to the adoption of a greedy strategy and being influenced by the global environment, the proposed algorithm may not select the maximum step length available for coverage under the current policy. Inspired by the RRT* algorithm, a rewrite strategy inspired by RRT* for coverage path planning is introduced (Liu, Wu, Yu, and Xue (2019); C. Zhang, Liu, Wei, An, and Liu (2022)). The schematic diagram of this method is depicted in Figure 4. This strategy aids in reducing the number of waypoints in the robot's coverage path, thereby lowering the energy consumption of the robot (Algorithm 1, lines 23-30).



(a) Prior to optimization with RRT*

(b) Following optimization with RRT*

FIGURE 4 Rewrite strategy inspired by RRT*.

4 | PATH FOLLOWING CONTROL ALRORITHM

Through the aforementioned coverage path planning algorithm, the target path for the biomimetic robotic fish is determined. To ensure precise tracking of this path, a sophisticated control strategy, incorporating both line-of-sight (LOS) and PID controllers, is implemented. This method adeptly guides the biomimetic robotic fish through the designated path, ensuring efficient and accurate path following.

The desired path is denoted as $\Omega = (x_d(\xi), y_d(\xi))$, where ξ is a parameter related to the path. The controller outputs different motion modes to ensure the biomimetic robotic fish converges to the desired path. To accomplish the aforementioned path tracking objective, a path tracking strategy is designed, as illustrated in Figure 5.

Initially, the LOS navigation method is applied to transform the path tracking problem into a heading angle tracking problem. Subsequently, a PID controller is designed based on the target yaw angle to generate different motion modes for the biomimetic robotic fish.

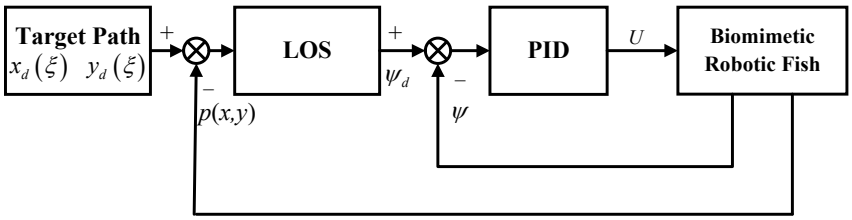


FIGURE 5 Path tracking strategy.

4.1 | LOS-based navigation strategy

The LOS navigation method, based on the principle of line-of-sight navigation used by experienced sailors, is a widely employed navigation system (Liu, Liu, and Yu (2021); Liu, Wu, Yu, and Tan (2018)). The line-of-sight navigation principle for any curve in a two-dimensional plane is illustrated in Figure 6. Any geometric path Ω on a two-dimensional plane can be represented by the variable ξ , and for any point on the path represented as $p_d(\xi) = \left[x_d(\xi), y_d(\xi)\right]^T$, the angle formed by the tangent line drawn at that point can be expressed:

$$\chi_p = \text{atan2}\left(\frac{\partial y_d}{\partial \xi}, \frac{\partial x_d}{\partial \xi}\right)$$

Taking the center of mass of the biomimetic robotic fish as the origin, a circle is chosen with an appropriate radius R. This circle intersects with the tangent line at the forward-looking point $p_{los}(x_{los}, y_{los})$ and the rear-looking point p_0. In the illustration, the forward-looking point serves as the guidance point for the biomimetic robotic fish, representing the next path point. At time t , the position of the robotic dolphin is $p(t) = \left[x(t), y(t)\right]^T$. When the lateral tracking error $e(t)$ of the biomimetic robotic fish to the tangent line approaches zero, the biomimetic robotic fish has approached that tangent line.

The target heading angle ψ_d is determined using the forward-looking method, as illustrated in Figure 6.

$$\chi_p = \chi_r + \chi_d$$

$$\chi_r = \text{atan2}(e, \Delta)$$

Where Δ represents the forward-looking distance, which can be calculated using the following formula:

$$e^2 + \Delta^2 = R^2$$

$$\chi_d = \psi_d$$

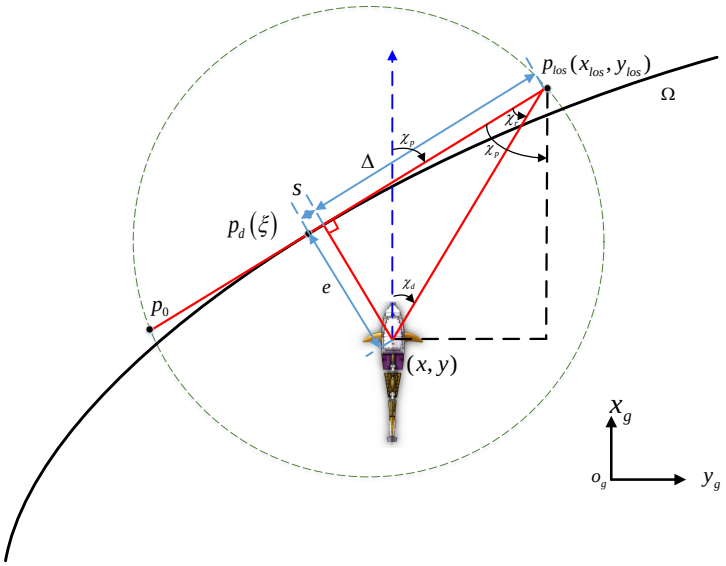


FIGURE 6 LOS-based navigation strategy for arbitrary two-dimensional geometric path.

Substituting the expression into the equation yields the target heading angle:

$$\psi_d = \chi_p - atan2\left(e, \sqrt{R^2 - e^2}\right)$$

4.2 | PID-based path following controller

During the execution of the path following task, the biomimetic robotic fish, equipped with an IMU sensor onboard, can obtain its pose heading angle ψ. The heading angle tracking error ψ_{e} is calculated by following.

$$\psi_{\mathrm{e}} = \psi - \psi_{\mathrm{d}}$$

The PID controller controls the motion mode U of the biomimetic robotic fish by summing the proportional, integral, and derivative coefficients of the tracking error ψ_{e} into the input. It continuously reduces the error, and when the error approaches and maintains zero, the control process concludes, signifying the completion of the path tracking task. The expression for the PID controller is as follows:

$$u(t) = K_P\left[e(t) + \frac{1}{T_I}\int_0^t e(t)dt + T_D\frac{de(t)}{dt}\right] = K_P e(t) + K_I\int_0^t e(t)dt + K_D\frac{de(t)}{dt}$$

In the equation: K_P, K_I, and K_D are the proportional, integral, and derivative coefficients, respectively; T_I and T_D are the integral and derivative time constants, respectively; $e\,(t)$ represents the tracking error of the heading angle ψ_{e}.

5 | SIMULATION AND ANALYSIS

The hardware configuration for the simulation experiments of this algorithm includes an Intel i9-12900H. The code implementation is carried out using Python, and the gym reinforcement learning algorithm toolkit is utilized, with modifications made to its functions such as step to meet the requirements of this study.

5.1 | Comparisons and ablation studies

5.1.1 | Evaluation metrics

To evaluate the efficacy of the coverage path planning algorithm, four key metrics are employed: total path length, repetition rate, number of waypoints, and number of turns. Smaller values in these metrics signify a higher quality of the generated coverage path. The definitions are as follows:

- Total path length (L_p): The number of grid cells actually traversed by the agent in the coverage path.
- Repetition rate (R_r): The repetition rate is defined as the ratio of the number of grid cells repeatedly covered by the agent in the coverage path to the total path length (L_p).
- Number of waypoints (P_n): The number of waypoints refers to the total count of waypoints generated by the coverage path planning algorithm.
- Number of turns (T_n): The number of turns in the path refers to the count of turns or changes in direction present in the generated path.

5.1.2 | Comparative algorithms

In this section, the efficacy of five distinct coverage path planning algorithms will be systematically evaluated across two environmental maps. The algorithms under consideration are:

1. PP Q-Learning: Predator-prey reward-based Q-Learning coverage path planning(M. Zhang et al., 2023)
2. PP DQN: Predator-prey reward-based Deep Q Network coverage path planning, where the Q-Learning in PP Q-Learning is replaced with DQN. The DQN network takes the global environment as input and outputs Q values for actions in the action space $A = \{forward, backward, turnleft, turnright\}$.
3. PP DQN RRT* rewrite: Predator-prey reward-based Deep Q Network and rewrite coverage path planning, an extension of PP DQN with the rewrite strategy inspired by RRT*.
4. DSS-PP DQN: the dynamic step size and predator-prey reward based Deep Q-network (DSS-PP DQN) coverage path planning algorithm. It builds upon PP DQN by incorporating a dynamic step size reward function. The DQN network takes the global environment as input and outputs Q values for actions in the action space(3.3).
5. Our proposed method: the dynamic step size and predator-prey reward based Deep Q-network (DSS-PPDQN) coverage path planning algorithm with the rewrite strategy inspired by RRT*.

The comparisons are presented in Table 3. These algorithms were tested on two environmental maps of different sizes, with one map measuring 20×20 and the other 40×40 grid cells. Static obstacles were placed on the maps.

TABLE 3 Comparison of Methodsfor Five Algorithms

Algorithm Name	Reinforcement Learning Methods	The PP Reward	The Dynamic Step Size Reward	The Rewrite Strategy Inspired by RRT*
PP Q-Learning	Q-Learning	Yes	No	No
PP DQN	DQN	Yes	No	No
PP DQN RRT* Rewrite	DQN	Yes	No	Yes
DSS-PP DQN	DQN	Yes	Yes	No
Our Proposed Method	DQN	Yes	Yes	Yes

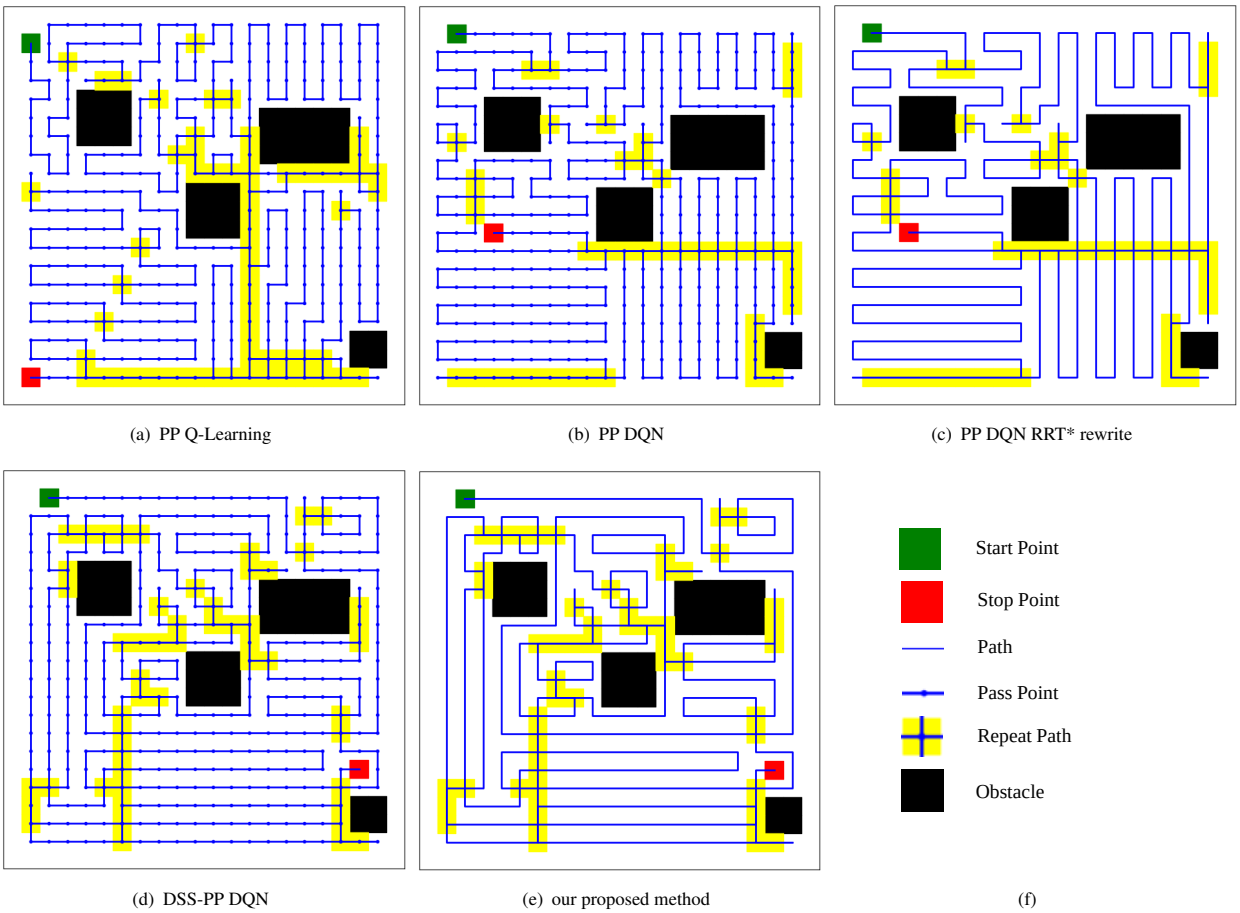
The parameters for these algorithms are shown in Table 4, with the greedy strategy parameter was set as 0.9. The predator's position was set to [100,100]. It was worth noting that the reward function for the PP DQN algorithm was chosen from the reward function provided in the PPCPP paper(M. Zhang et al., 2023), which included the weighted factor of the shortest path.

TABLE 4 Configuration of Parameters for Five Algorithms

Algorithm Name	w^d	w^s	w^L	w^b	Dimension of Input	Dimension of Output
PP Q-Learning(M. Zhang et al., 2023)	1	0.53	0	0.48	None	None
PP DQN	1	0.53	0	0.48	20*20/40*40	4
PP DQN RRT* Rewrite	1	0.53	0	0.48	20*20/40*40	4
DSS-PP DQN	1	1	1	1	20*20/40*40	4*20/4*40
Our Proposed Method	1	1	1	1	20*20/40*40	4*20/4*40

5.2 | Return

We conducted a total of 1000 training episodes and averaged the results over 100 experiments for the evaluation of performance metrics. These experiments involved coverage path planning in two different environment map sizes (20×20 and 40×40 grid maps). Examples of planned paths are illustrated in Figures 7 and 8, while the results of the evaluation metrics are presented in Tables 5 and 6.

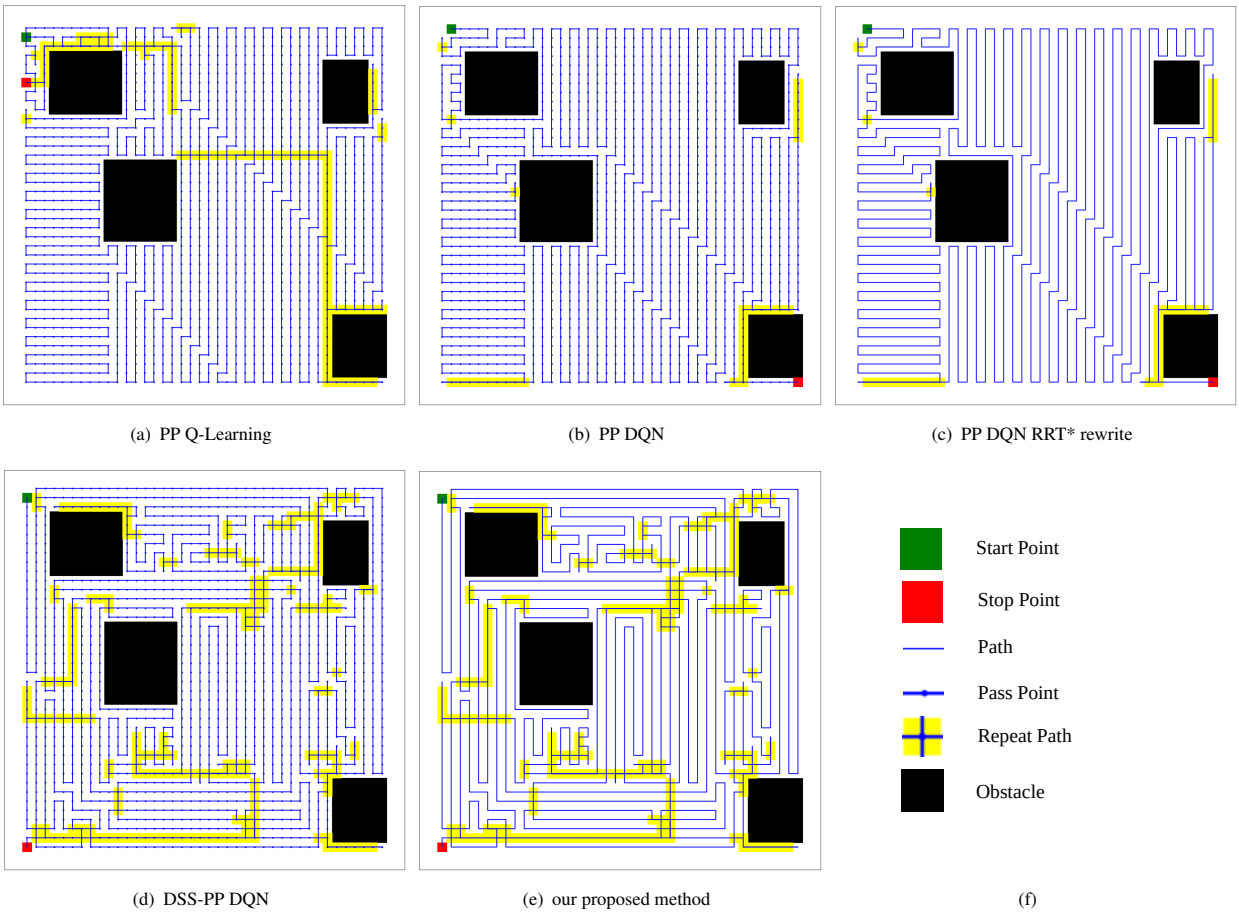


(a) PP Q-Learning (b) PP DQN (c) PP DQN RRT* rewrite

(d) DSS-PP DQN (e) our proposed method (f)

FIGURE 7 Schematic diagram of different planning methods (40×40).

From the analysis of Tables 5 and 6, it was evident that PP DQN significantly optimized the coverage path planning, notably reducing the number of path waypoints and turning occurrences while maintaining a stable path length and repetition rate. This optimization was crucial for minimizing the agent's turning frequency and overall coverage time. Specifically, as detailed in Table 5, on a 20×20 map, our proposed algorithm achieved a remarkable 75% reduction in the number of path waypoints and a 31% decrease in turning occurrences compared to PP Q-Learning. Similarly, on the same 20×20 map, our algorithm



(a) PP Q-Learning (b) PP DQN (c) PP DQN RRT* rewrite

(d) DSS-PP DQN (e) our proposed method (f)

FIGURE 8 Schematic diagram of different planning methods (40 × 40).

TABLE 5 Comparison of Evaluation Metrics (20 × 20)

Algorithm Name	Lp	Rr	Pn	Tn
PP Q-Learning(M. Zhang et al., 2023)	422	14	422	154
PP DQN	405	10	405	122
PP DQN RRT* Rewrite	405	10	122	122
DSS-PP DQN	425	14	425	107
Our Proposed Method	425	14	107	107

TABLE 6 Comparison of Evaluation Metrics (40 × 40)

Algorithm Name	Lp	Rr	Pn	Tn
PP Q-Learning(M. Zhang et al., 2023)	1514	7	1514	300
PP DQN	1447	3	1447	279
PP DQN RRT* Rewrite	1447	3	271	279
DSS-PP DQN	1583	11	1583	275
Our Proposed Method	1583	11	275	275

demonstrated a 12% reduction in path waypoints and an impressive 74% decrease in turning occurrences compared to PP DQN. Moving to the 40 × 40 map, as indicated in Table 6, our proposed algorithm showcased an 82% reduction in path waypoints and an 8% decrease in turning occurrences compared to PP Q-Learning. Likewise, on the 40 × 40 map, our algorithm achieved an 81% reduction in path waypoints and a 1% decrease in turning occurrences compared to PP DQN.

However, in Table 6, our proposed algorithm shows a 4% increase in path length and a 57% increase in repetition rate compared to PP Q-Learning. To address the issues of increased path length and repetition rate, we optimized the weighting factors of the proposed four reward functions.

5.3 | Optimization of weight factors

To address the issues of increased path length and higher repetition rates, we employed Non-dominated Sorting Genetic Algorithm II(NSGA-II) to optimize the weighting factors in our proposed algorithm.

The objectives of the NSGA-II are to simultaneously minimize the path length, repetition rate, and number of turns. Through NSGA-II selection, blend crossover, and Gaussian mutation applied to an initial population of 50 individuals, we selected the best solutions from the Pareto front as $[w1 = 0.32, w2 = 0.73, w3 = 0.29, w4 = 0.31]$. Finally, the evaluation metrics are presented in Tables 7 and 8.

TABLE 7 Optimized Evaluation Metrics (20×20)

Algorithm Name	Lp	Rr	Pn	Tn
Our Proposed Method	425	14	107	107
Our Proposed Method Optimizes Weight Factors	418	13	96	96

TABLE 8 Optimized Evaluation Metrics (40×40)

Algorithm Name	Lp	Rr	Pn	Tn
Our Proposed Method	1583	11	275	275
Our Proposed Method Optimizes Weight Factors	1583	11	243	243

From the analysis of Tables 7 and 8, it was evident that after optimizing the weighting factors, all four evaluation metrics exhibited a decrease. As presented in Table 7, on the 20×20 map, there was a 2% reduction in the total path length, a 7% decrease in the repetition rate, a 10% reduction in the number of path waypoints, and a 10% decrease in the number of turns. In Table 8, on the 40×40 map, while the total path length and repetition rate remained unchanged, there was a 12% decrease in the number of path waypoints and a 12% reduction in the number of turns. Thus, optimizing the weighting factors proved to be meaningful for enhancing the coverage path optimization for underwater robots.

6 | FIELD EXPERIMENTAL VERIFICATION

The experimental deep-sea net cage utilized in this study was the "Blue Diamond II," situated at the deep-sea net of Shandong Laizhou Mingbo Fish Farm. This facility constituted a large-scale intelligent deep-sea net encompassing a perimeter of 160 m and a water volume of up to 20,000 m^3. It featured two offshore multifunctional platforms covering an area of 600 m^2 and was equipped with substantial pneumatic feeding apparatus, underwater robots, among other amenities. Its primary purpose was the ecological and healthful breeding of high-quality fish species, such as Epinephelus fuscoguttatus.

The hybrid-driven biomimetic robotic fish was deployed for surface coverage and water quality inspection tasks within the "Blue Diamond II." This involved monitoring water quality indicators, including pH value, dissolved oxygen, temperature, etc., to enhance the overall quality of aquaculture. The experiments were conducted in the past to validate the effectiveness of the proposed coverage path planning algorithm in real-world underwater environments.

Given the specifications of the water quality monitoring zone and the turning radius constraints of the hybrid-driven biomimetic robotic fish, the deep-sea net "Blue Diamond II" in this experiment was partitioned into a grid map. Each grid had a side length of 7 m. Employing the proposed algorithm, a coverage path was meticulously planned. Subsequently, the biomimetic fish autonomously navigated through the predetermined waypoints, effectively fulfilling the aquatic inspection task. This

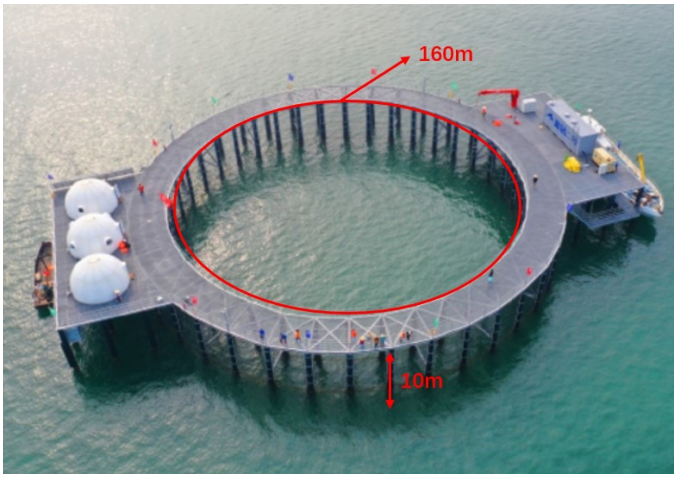
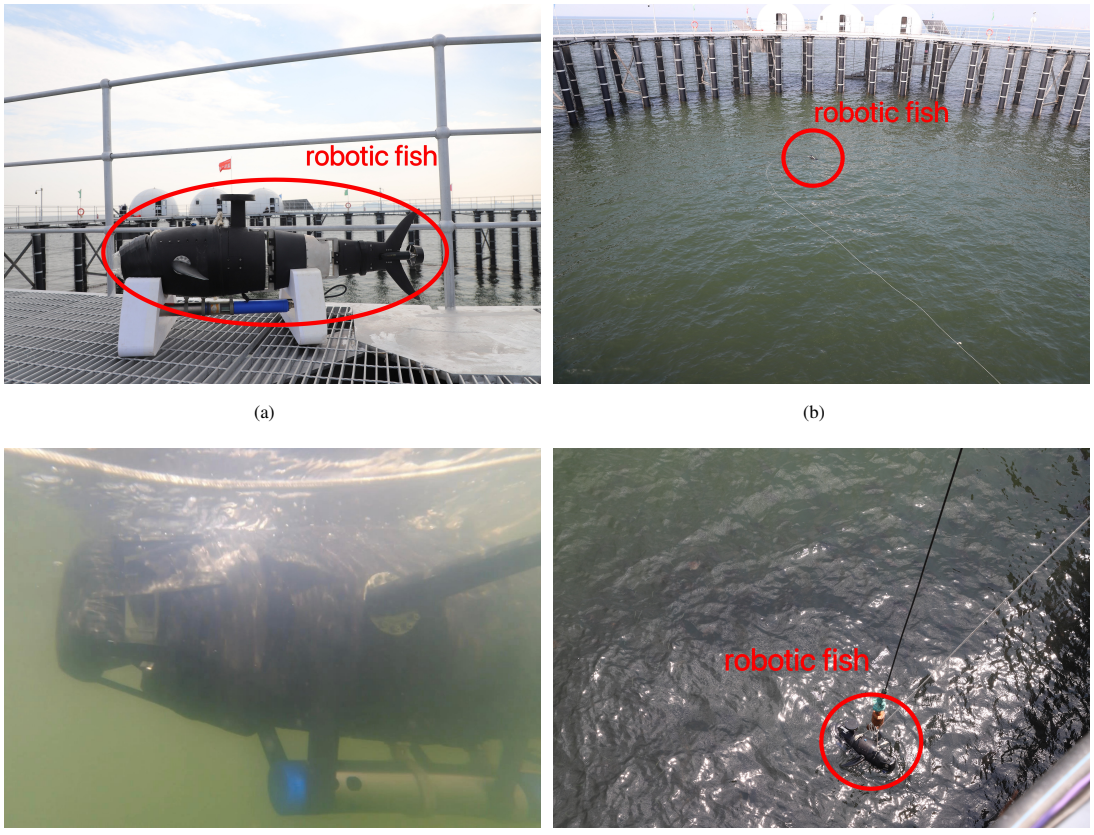


FIGURE 9 The "Blue Diamond II" deep-sea net cage.

experimental setup and execution occurred in the past as part of the validation process for the proposed coverage path planning algorithm in real-world underwater environments. The on-site experiment is shown in Figure 10.

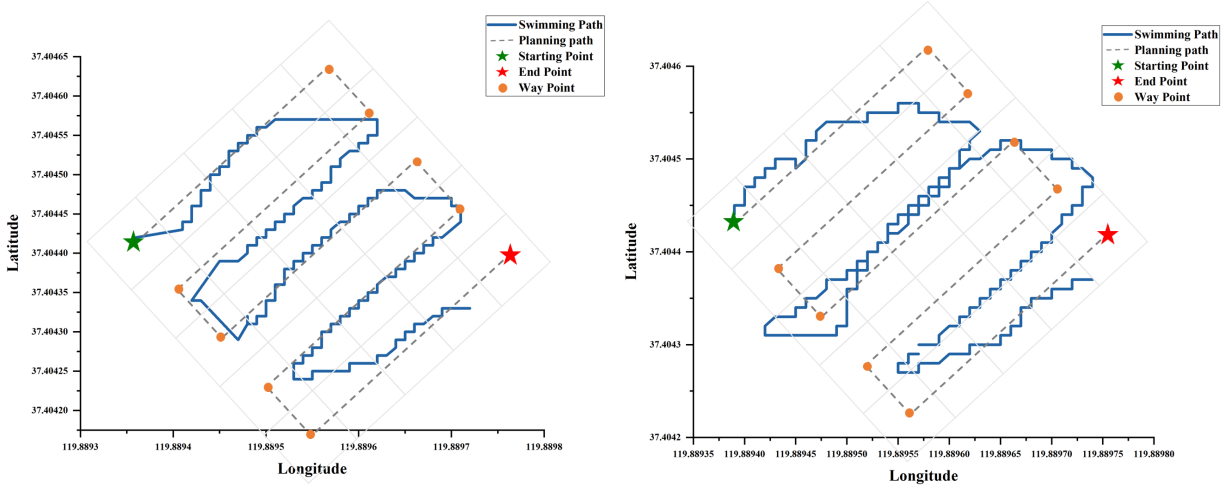


(a) (b)

(c) (d)

FIGURE 10 The on-site experiment images.

In order to cope with adverse offshore weather conditions, water quality monitoring experiments were conducted, including scenarios with sea winds reaching force four. A comparison between the planned path of the bionic fish and its actual swimming path is illustrated in Figure 11.

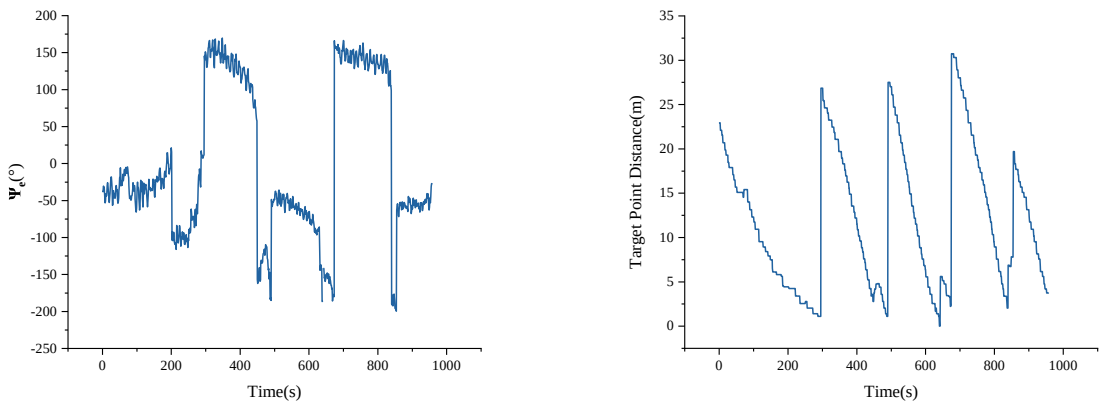


(a) The robot tracks along the predetermined path under normal wind and waves.

(b) The robot exhibits tracking deviation under wind and waves.

FIGURE 11 Comparison between the planned path of the bionic fish and its actual swimming path.

From Figure 12, it can be observed that under normal wind and wave conditions, the bionic fish can complete path tracking along the planned route. In the case of sea winds reaching force 4, the bionic fish exhibits yawing issues but still manages to track each target point. Therefore, it can be concluded that the biomimetic robotic fish is capable of completing the full coverage path tracking task, although improvements in its resistance to strong winds are needed. The heading angle tracking errors during the navigation through 9 target points are depicted in Figure 12(a) during the biomimetic robotic fish's movement. When maintaining a straight-line trajectory, the heading angle tracking error fluctuates around 0, while during turning maneuvers, the heading angle tracking error momentarily increases and tends towards 0 upon completing the turn. Figure 12(b) illustrates the distance error between the bionic fish and the next upcoming target point during the path tracking process, with an error of 2 meters.



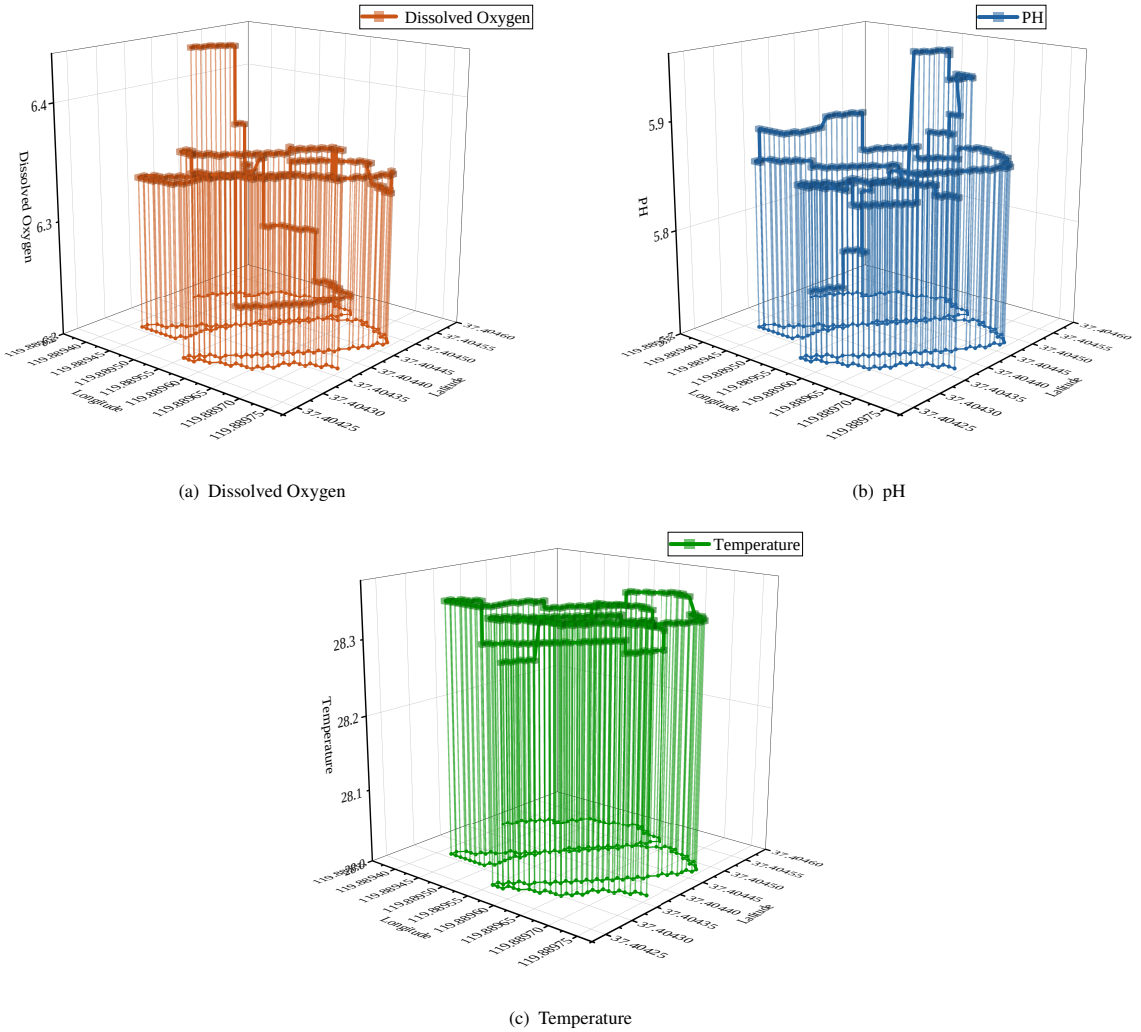
(a) Heading Angle Tracking Error

(b) Distance to Target Point

FIGURE 12 Path following errors.

Simultaneously, multi-parameter water quality sensors carried by the biomimetic robotic fish collected water quality data such as pH, temperature, and dissolved oxygen in the "Blue Diamond II" net. The distribution of water quality data is illustrated

in Figure 13. By further analyzing the time-series characteristics of water quality parameters and the spatial distribution of water quality in the deep-sea net, dynamic and three-dimensional spatiotemporal warnings for deep-sea net water quality can be achieved.



(a) Dissolved Oxygen

(b) pH

(c) Temperature

FIGURE 13 Water quality data distribution.

7 | DISCUSS

The proposed coverage path planning algorithm based on deep reinforcement learning in this paper reduces the number of path waypoints and turns by introducing three core modules: DQN, reward function, and rewrite strategy inspired by RRT*. The comparative experiments in Tables 5 and 6 also demonstrate the effectiveness of these three core modules. The algorithm introduces a dynamic step size reward function, which increases the maximum dimension of the action space, slightly raising the computational cost of the algorithm. However, this increase has a minor impact on the overall computational cost of the algorithm. As shown in Table 5, compared to Algorithm 3 without dynamic step size reward function, our proposed algorithm reduces the number of turns by 12%.

The proposed algorithm has been successfully applied to the coverage path planning for aquatic net monitoring of biomimetic robotic fish in practical experiments conducted at the Shandong Laizhou Mingbo Fish Farm. As the experimental platform is 10

meters above the sea surface, a crane is required to lift the biomimetic robotic fish. Therefore, as seen in Figure 10, there is a white rope on the biomimetic robotic fish to facilitate lifting. This rope does not interfere with the biomimetic robotic fish's swimming and does not affect path tracking performance. Despite being affected by wind and waves, the biomimetic robotic fish's swimming path may deviate but can still return to the correct tracking path, demonstrating the good performance of the proposed tracking algorithm and the biomimetic robotic fish.

8 | CONCLUCTION

This paper proposes a coverage path planning algorithm based on deep reinforcement learning to address the problem of coverage path planning for underwater biomimetic robots. The algorithm, while ensuring the total length and repetition rate of the path, reduces the number of path points and turning times, thereby reducing the path coverage time and energy consumption of underwater robots. The proposed algorithm consists of three core modules: DQN, reward function, and rewrite strategy inspired by RRT*. The output of DQN determines not only the current action direction but also influences the choice of the current step length to minimize turning times. The reward function is divided into four parts: dynamic step size reward function, predation avoidance reward function, smoothness reward function and boundary reward function, to optimize the coverage path. The rewrite strategy inspired by RRT* replans path points, reduces the number of nodes on the path, and lowers the robot's path coverage time.

The proposed coverage path planning algorithm is validated through simulation experiments. The generated coverage paths on 20×20 and 40×40 grid maps demonstrate that compared to the existing Predator-Prey Reward Based Q-Learning Coverage Path Planning(M. Zhang et al., 2023), the proposed algorithm reduces the number of path points and turning times by 75% and 31%, respectively, while ensuring the total length and repetition rate of the path. Furthermore, the proposed algorithm is successfully applied in the practical coverage path planning of biomimetic robotic fish for deep-sea net water quality monitoring, providing strong scientific support for water quality management, pollution source control, and environmental planning.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Programs of China(Grant No. 2022YFE0107100) and Nationa Natural Science Foundation of China(Grant No.62273351).

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

References

Bai, L. (2022). Research on path planning of intelligent mowing system based on computer vision. In *2022 international conference on computing, robotics and system sciences (icrss)* (p. 10-14). IEEE.

Cao, Y., Cheng, X., & Mu, J. (2022). Concentrated coverage path planning algorithm of uav formation for aerial photography. *IEEE Sensors Journal*, *22*(11), 11098-11111.

Gan, X., Huo, Z., & Li, W. (2023). Dp-a*: For path planing of ugv and contactless delivery. *IEEE Transactions on Intelligent Transportation Systems*.

Hassan, M., & Liu, D. (2019). Ppcpp: A predator–prey-based approach to adaptive coverage path planning. *IEEE Transactions on Robotics*, *36*(1), 284-301.

Huang, Y., Xu, J., Shi, M., & Liu, L. (2022). Time-efficient coverage path planning for energy-constrained uav. *Wireless Communications and Mobile Computing*, *2022*, 1-15.

Ji, Y., Wei, Y., Liu, J., & An, D. (2023). Design and realization of a novel hybrid-drive robotic fish for aquaculture water quality monitoring. *Journal of Bionic Engineering*, *20*(2), 543-557.

Kim, J. S., & Kim, B. K. (2010). Minimum-time grid coverage trajectory planning algorithm for mobile robots with battery voltage constraints. In *Iccas 2010* (p. 1712-1717). IEEE.

Le, A. V., Veerajagadheswar, P., Thiha Kyaw, P., Elara, M. R., & Nhan, N. H. K. (2021). Coverage path planning using reinforcement learning-based tsp for htetran—a polyabolo-inspired self-reconfigurable tiling robot. *Sensors*, *21*(8), 2577.

Liu, J., Liu, Z., & Yu, J. (2021). Line-of-sight based three-dimensional path following control for an underactuated robotic dolphin. *Science China Information Sciences*, *64*, 1-12.

Liu, J., Wu, Z., Yu, J., & Tan, M. (2018). Sliding mode fuzzy control-based path-following control for a dolphin robot. *Sci. China Inf. Sci.*, *61*(2), 024201:1-024201:3.

Liu, J., Wu, Z., Yu, J., & Xue, Z. (2019). Cooperative target tracking in aquatic environment using dual robotic dolphins. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*(8), 4782-4792.

Lu, J., Zeng, B., Tang, J., Lam, T. L., & Wen, J. (2023). Tmstc*: A path planning algorithm for minimizing turns in multi-robot coverage. *IEEE Robotics and Automation Letters*.

Luis, S. Y., Reina, D. G., & Marín, S. L. T. (2020). A deep reinforcement learning approach for the patrolling problem of water resources through autonomous surface vehicles: The ypacarai lake case. *IEEE Access*, *8*, 204076-204093.

Luo, M., Tian, Y., Li, E., Chen, M., & Tan, M. (2024). A local obstacle avoidance and global planning method for the follow-the-leader motion of coiled hyper-redundant manipulators. *IEEE Transactions on Industrial Informatics*.

Maini, P., Gonultas, B. M., & Isler, V. (2022). Online coverage planning for an autonomous weed mowing robot with curvature constraints. *IEEE Robotics and Automation Letters*, *7*(2), 5445-5452.

Masmitja, I., Martin, M., O Reilly, T., Kieft, B., Palomeras, N., Navarro, J., & Katija, K. (2023). Dynamic robotic tracking of underwater targets using reinforcement learning. *Science robotics*, *8*(80), eade7811.

Miao, X., Lee, H.-S., & Kang, B.-Y. (2020). Multi-cleaning robots using cleaning distribution method based on map decomposition in large environments. *IEEE Access*, *8*, 97873-97889.

Mitschke, M., Uchiyama, N., & Sawodny, O. (2018). Online coverage path planning for a mobile robot considering energy consumption. In *2018 ieee 14th international conference on automation science and engineering (case)* (p. 1473-1478). IEEE.

Nørremark, M., Nilsson, R. S., & Sørensen, C. A. G. (2022). In-field route planning optimisation and performance indicators of grain harvest operations. *Agronomy*, *12*(5), 1151.

Piardi, L., Lima, J., Pereira, A. I., & Costa, P. (2019). Coverage path planning optimization based on q-learning algorithm. In *Aip conference proceedings* (Vol. 2116, p. 220002). AIP Publishing LLC. 1.

Ramesh, M., Imeson, F., Fidan, B., & Smith, S. L. (2022). Optimal partitioning of non-convex environments for minimum turn coverage planning. *IEEE Robotics and Automation Letters*, *7*(4), 9731-9738.

Shakeri, R., Al-Garadi, M. A., Badawy, A., Mohamed, A., Khattab, T., Al-Ali, A. K., ... Guizani, M. (2019). Design challenges of multi-uav systems in cyber-physical applications: A comprehensive survey and future directions. *IEEE Communications Surveys & Tutorials*, *21*(4), 3340-3385.

Tan, C. S., Mohd-Mokhtar, R., & Arshad, M. R. (2021). A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. *IEEE Access*, *9*, 119310-119342.

Van Pham, H., Asadi, F., Abut, N., & Kandilli, I. (2019). Hybrid spiral stc-hedge algebras model in knowledge reasonings for robot coverage path planning and its applications. *Applied Sciences*, *9*(9), 1909.

Wagner, N., Kirk, R., Hanheide, M., & Cielniak, G. (2021). Efficient and robust orientation estimation of strawberries for fruit picking applications. In *2021 ieee international conference on robotics and automation (icra)* (p. 13857-13863). IEEE.

Wang, Y., He, Z., Cao, D., Ma, L., Li, K., Jia, L., & Cui, Y. (2023). Coverage path planning for kiwifruit picking robots based on deep reinforcement learning. *Computers and Electronics in Agriculture*, *205*, 107593.

Wijegunawardana, I. D., Muthugala, M. V. J., Samarakoon, S. B. P., Hua, O. J., Padmanabha, S. G. A., & Elara, M. R. (n.d.). Insights from autonomy trials of a self-reconfigurable floor-cleaning robot in a public food court. *Journal of Field Robotics*.

Xie, Z., Yuan, F., Liu, J., Tian, L., Chen, B., Fu, Z., ... He, X. (2023). Octopus-inspired sensorized soft arm for environmental interaction. *Science Robotics*, *8*(84), eadh7852.

Yan, S., Wu, Z., Wang, J., Li, S., Tan, M., & Yu, J. (2023). Towards unusual rolled swimming motion of a bioinspired robotic hammerhead shark under negative buoyancy. *IEEE/ASME Transactions on Mechatronics*.

Yu, H., Wang, P., Wang, J., Ji, J., Zheng, Z., Tu, J., ... Shen, S. (2023). Catch planner: Catching high-speed targets in the flight. *arXiv preprint arXiv:2302.04387*.

Zhang, C., Liu, Z., Wei, Y., An, D., & Liu, J. (2022). Improved rrt*-a*-based three-dimensional path planning algorithm for the robotic dolphin. In *2022 ieee international conference on real-time computing and robotics (rcar)* (p. 81-86). IEEE.

Zhang, G., Ji, C., Wu, Q., Liu, H., Zhou, Y., & Fu, J. (2022). Study on path planning of mechanized harvesting of ratoon rice in the first season based on the capacitated arc routing problem model. *Frontiers in Plant Science*, *13*, 963307.

Zhang, M., Cai, W., & Pang, L. (2023). Predator-prey reward based q-learning coverage path planning for mobile robot. *IEEE Access*, *11*, 29673-29683.

Zhang, P., Wu, Z., Chen, D., Tan, M., & Yu, J. (2023). Autonomous dynamic hitch-hiking control of a bionic robotic remora. *IEEE Transactions on Industrial Electronics*.

Zhou, Y., Li, P., Ye, Z., Yue, L., Gui, L., Jiang, X., ... Liu, Y. H. (2022). Building information modeling-based 3d reconstruction and coverage planning enabled automatic painting of interior walls using a novel painting robot in construction. *Journal of Field Robotics*, *39*(8), 1178-1204.

Zhu, D., Tian, C., Sun, B., & Luo, C. (2019). Complete coverage path planning of autonomous underwater vehicle based on gbnn algorithm. *Journal of Intelligent & Robotic Systems*, *94*, 237-249.

Zhu, J., White, C., Wainwright, D. K., Di Santo, V., Lauder, G. V., & Bart-Smith, H. (2019). Tuna robotics: A high-frequency experimental platform exploring the performance space of swimming fishes. *Science Robotics*, *4*(34), eaax4615.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.