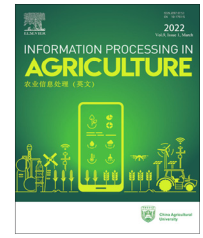
Available at www.sciencedirect.com

INFORMATION PROCESSING IN AGRICULTURE 9 (2022) 91–99

journal homepage: www.elsevier.com/locate/inpa



Path planning with multiple constraints and path following based on model predictive control for robotic fish

Yizhuo Mu [a,b], *Jingfen Qiao* [b,c], *Jincun Liu* [a,b], *Dong An* [a,b,*], *Yaoguang Wei* [a,b,*]

[a] *College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China*
[b] *National Innovation Center for Digital Fishery, China Agricultural University, Beijing 100083, China*
[c] *School of Computing, The Australia National University, Canberra 2600, Australia*

A R T I C L E I N F O

Article history:
Available online 30 December 2021

Keywords:
Path planning
Path following
Robotic fish
Beetle swarm optimization
Model predictive control

A B S T R A C T

This paper discusses the path planning and path following control problems of robotic fish. In order to avoid obstacles when robotic fish swim in a complex environment, a path planning method based on beetle swarm optimization (BSO) algorithm is developed. This method considers the influence of the robotic fish's volume and motion constraints on the path planning task, which can eliminate the collision risk and meet the constraint of the minimum turning radius when the robotic fish obtains the planned path. In constructing the path following controller, a multilayer perception based model predictive control (MPC) is adopted to design the optimal control method, and the objective function of the optimal control is dynamically adjusted according to the path curvature. The simulation results show that this proposed method can effectively overcome the complexity of robotic fish kinematics modelling and adapt well to the reference paths of different curvatures given by the path planner.

1. Introduction

With the development of underwater research, the study of underwater motion and navigation control has attracted more and more attention. Path planning and the following is an essential research content of underwater robotic fish motion and navigation control. The purpose of robotic fish path planning is to find an optimal path specified from the starting point to the target according to objectively feasible areas, such as the shortest path or the least energy consumption, while path following is responsible for bringing robotic fish along with the defined optimal path. Most of the research on the path planning of robotic fish only transplant the general path planning method of a mobile robot to robotic fish. Hu et al. [1] combined reinforcement learning with simulated annealing to achieve path planning for robotic fish in an unknown environment. However, their study did not consider the effects of the robotic fish's physical form and motion constraints on the complex underwater environments. To overcome these influences, Yu et al. [2] considered the irreversible characteristics of robotic fish and designed a path planning method based on a neural network to adapt to the complex underwater environments. Furthermore, Hou et al. [3] proposed a path planning method based on an ant colony algorithm to realize the motion control of robotic fish. In

* Corresponding authors at: College of Information and Electrical Engineering, China Agricultural University, 17 Tsinghua East Road, Beijing 100083, China.
E-mail addresses: andong@cau.edu.cn (D. An), wyg@cau.edu.cn (Y. Wei).
Peer review under responsibility of China Agricultural University.
https://doi.org/10.1016/j.inpa.2021.12.005

order to minimize the energy consumption and avoid sharp turnings that cannot meet the minimum turning radius of the robotic fish, the length and the curvature of the generated path are considered in the design of the fitness function. The sharp turnings that do not meet the minimum turning radius constraint in the generated path will be replaced by arcs. However, such a sharp turning approach may put the planned path at risk of collision with obstacles.

After the path planning method obtains the optimal reference path, the control system needs to make the robotic fish move along the reference path by the path following control. The underwater path following control is more challenging than other environmental control due to the complexity of the underwater environment and the strong coupling and nonlinear characteristics of underwater robotic fish. Taking the traditional control method PID as an example, it is found in [4] that the PID control method is often challenging to deal with dynamic strategies and complex underwater environments. This is because the parameter tunning in PID control is complex and is not competent for multi-objective optimization of multi-input and multi-output (MIMO) complex control systems. Other control methods have been used to control the motion of robotic fish, such as sliding mode control [5], auto disturbance rejection control [6], fuzzy logic control [7], and the line-of-sight method [8,9] is the most used in path following control. However, most of these traditional methods require a complex modelling process. In theory, the model predictive control (MPC) can achieve optimal control performance and easily cope with MIMO systems. But its performance is limited due to the low accuracy of the prediction model. The conventional method to improve the control performance is to build a more complex model or use a nonlinear optimization solver. This method has some problems, such as complex modelling and high computational complexity.

With the rapid development of the deep neural network, combining the deep neural network model and control technology has injected new vitality into traditional control methods. Learning and optimization can overcome the uncertainty of the model and improve the control system's performance [10]. Nagabandi et al. [11] combined the MPC framework with a neural network to realize robot path following control. The experimental results show that the dynamic path following the robot model can be trained by using the data collected by the robot acting randomly in the environment. The prediction model in this approach can be flexibly applied to various reference paths only by modifying the objective function. Furthermore, Xie et al. [12] proposed a neural network based MPC framework to solve dynamic ships' online collision avoidance problem. They interactively train the neural networks to approximate the optimal policy in the MPC framework for collision avoidance in real-time. Ma et al. [13] adopted the deep reinforcement learning to motion decision and path following control of robot fish. Experimental results show that the control method based on reinforcement learning has better path following ability than the traditional PID control method. Wang et al. [14] designed an iterative learning controller that can effectively follow the path of robotic fish and enhance the adaptability to the aquatic environment. The learning-based method makes the prediction model and controller adapt to the dynamic environment more precisely by interacting with the environment.

In this paper, a path planning and following method is proposed to control the motion of robotic fish in a complex environment. Based on establishing a two-dimensional environment model with obstacles, the path planning method based on the BSO algorithm was used to obtain the reference path of the robotic fish. On this basis, the path following of robotic fish is realized by the learning-based model predictive control method. The rest of the paper is organized as follows. The second section will first propose the simplified model of robotic fish and the path planning and following framework of robotic fish. Then, a path planning method based on a beetle swarm optimization algorithm is introduced in Section 2.3, and the MPC controller for path following control is designed in Section 2.4. After that, the overall simulation experiment and the performance of planning and control will be discussed in Section 3. The concluding remarks will be presented in the end of this paper.

2. Materials and methods

2.1. Simplified model of robotic fish

The body and/or caudal fin propulsion (BCF) mode is the swimming mode most fish in nature. Compared to the median and/or paired fin propulsion (MPF) mode, it offers unparalleled advantages in swimming speed, high-speed propulsion efficiency and acceleration performance. The fish swims in this mode mainly propelled by wave power from the resilient tail in the back third of the body, while the front two-thirds barely fluctuates. The research on BCF mode is the main means to realize high-speed swimming and efficient propulsion of robotic fish [15]. The simplified fish model can be divided into head, trunk, and tail (the tail includes caudal peduncle and caudal fin). Therefore, the robotic fish can be simplified into a multi-jointed chain physical model, as shown in Fig. 1.

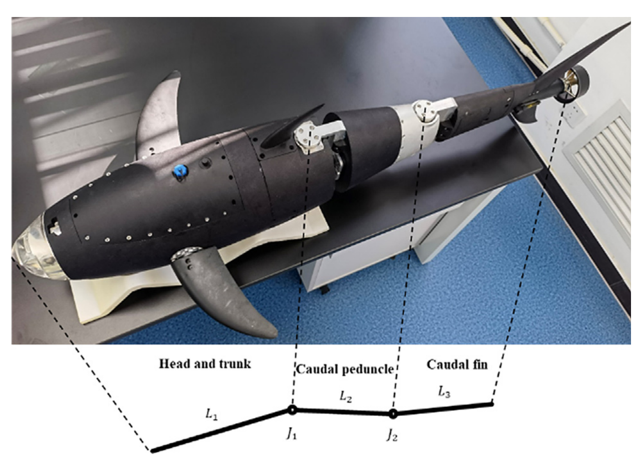


Fig. 1 – Simplified physical model of robotic fish. Head and trunk, caudal peduncle and caudal fin correspond to joint segments 1, 2 and 3, respectively. Joint segments are connected by joints.

We use the Swimmer model in MuJoCo [16] physical simulator as the simulation model for robotic fish path following control, as shown in Fig. 2. MuJoCo is a physics simulator that can be used for research such as robot control optimization. In the simulation model of robotic fish based on the MuJoCo physics simulator, the state space of robotic fish is represented by a 16-dimensional vector. The state-space describes the position and velocity information of the sliding joint, the angle and angular velocity information of the hinge joint, and the position and velocity information of the center of mass. In MuJoCo, the sliders are defined by a position and a sliding direction, and the sliding direction is set to the direction of the X-axis or the direction of the Y-axis in the Swimmer model. The hinges are defined by position and rotation axis and have one rotational degree of freedom, set to rotate about the Z-axis in the Swimmer model. The action space of the robotic fish is represented by a 2-dimensional vector, which describes the dynamic mechanism formed by the torques of the two hinge joints.

2.2. Path planning and following framework of robotic fish

In recent years, some researchers have tried to solve the problem of path planning and path following simultaneously. Sgorbissa [17] proposed an innovative, integrated solution for path planning, path following control, and obstacle avoidance, applicable to a two-dimensional and three-dimensional navigation. Shen et al. [18] attempted to develop a unified receding horizon optimization project for the integrated path planning and path following control of the autonomous underwater vehicle. Wang et al. [19] studied an autonomous underwater vehicle's planning and following control problems. The particle swarm optimization (PSO) algorithm and the MPC method are used to realize the path planning and the path following with dynamic uncertainty and wave disturbance, respectively. Ulyanov et al. [20] developed an event-based framework for an underactuated autonomous underwater vehicle, which contains two parts: the first part for online path planning and the second part for path following control. Castaño et al. [21] adopted an ergodic exploration method to optimize the trajectory of the tail-actuated robotic fish sensing operation, enabling robotic fish to achieve efficient exploration and energy-saving tracking control of the planned trajectory. A nonlinear model, predictive control algorithm, is proposed for trajectory tracking control. Lu et al. [22] proposed a vision-based path planning and path following system for robotic fish. The planning part uses the A* algorithm, and the PID controller is used in the following part.

Inspired by the above literature, this paper proposes a framework for path planning and path following of robotic fish, as shown in Fig. 3. Firstly, the reference path of the following task is obtained through environment modelling and optimization solution in the path planner. Then the objective function of optimal control is constructed based on the reference path. Finally, the prediction model of the MPC controller and rolling optimization are used to achieve the robotic fish path following the control task.

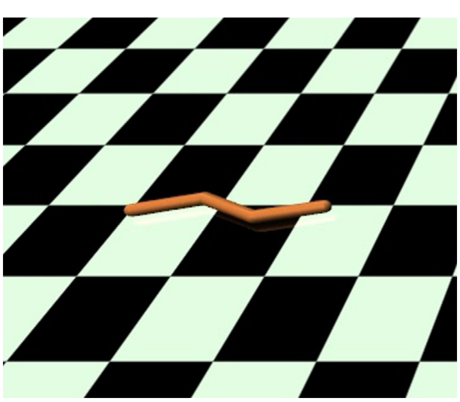


Fig. 2 – Physics simulator.

2.3. Path planning based on BSO algorithm

Inspired by the swarm intelligence algorithms, Wang et al. [23] proposed the BSO algorithm by combining the beetle antennae search (BAS) algorithm [24] based on the beetle foraging mechanism with the PSO algorithm. As the PSO algorithm's principle, each beetle's position in the BSO algorithm can be used as a potential solution to the optimization problem, and information can also be shared between different beetles. The speed update for each beetle depends on the moving trend of the beetle approaching individual extremes and the trend of the beetle swarm approaching the extreme global value. In contrast, the BSO algorithm differs from the PSO algorithm in that the position updating of each beetle depends not only on the speed updating but also on the fitness function value of the antenna detection position. The BSO algorithm is updated iteratively as follows:

$$x_{is}^{k+1} = x_{is}^{k} + \lambda v_{is}^{k} + (1-\lambda)\xi_{is}^{k}, \tag{1}$$

$$v_{is}^{k+1} = \omega v_{is}^{k} + c_1 r_1 (p_{is}^{k} - x_{is}^{k}) + c_2 r_2 (p_{gs}^{k} - x_{gs}^{k}), \tag{2}$$

$$\xi_{is}^{k+1} = \delta^{k} * v_{is}^{k} * sign(f(x_{rs}^{k}) - f(x_{ls}^{k})), \tag{3}$$

where $s = 1, 2, \ldots, S$, $i = 1, 2, \ldots, n$, k is the iteration time. ξ_{is} represents the position increment determined by the value of the fitness function of the antenna detection position. δ^k is the searching step length of the beetle. $f(x_{rs})$ and $f(x_{ls})$ are the fitness function values of the two antennae detection positions of the beetle. $X_i = (x_{i1}, x_{i2}, \ldots, x_{iS})^T$ represents the position attribute of individual beetle i in the S-dimensional search space, which is a potential solution of the optimization problem. $V_i = (v_{i1}, v_{i2}, \ldots, v_{iS})^T$ represents the speed attribute

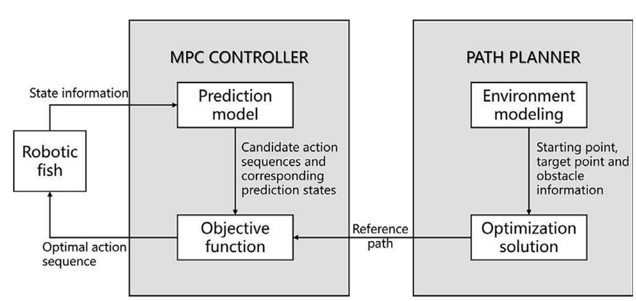


Fig. 3 – The overall framework of robotic fish path planning and path following control.

of individual beetle i. $P_i = (p_{i1}, p_{i2}, \ldots, p_{iS})^T$ represents the individual extreme, and $P_g = (p_{g1}, p_{g2}, \ldots, p_{gS})^T$ represents the global extreme. sign(.) is the symbol function. The positive constant λ in Eq. (1), inertia weight ω, parameter c_1 and c_2 in Eq. (2) are adjustable parameters. r_1 and r_2 are random numbers with a domain ranging between 0 and 1.

The iterative update strategy of the BSO algorithm combines the search mechanism of beetle single and the update strategy of the PSO algorithm. Compared with other swarm intelligent algorithms, the BSO algorithm has apparent advantages in solving optimization problems because of its fast iterative convergence speed and avoiding falling into local extremes.

In our previous work [25], we first applied the BSO algorithm to the three-dimensional path planning problem. Experimental results show that the iterative convergence speed of the path planning method is fast, the convergence result is stable, and it is not easy to fall into the optimal local solution. The controlled object is studied as a mass point in this work, but the actual robotic fish physical model is the two-joint chain physical model mentioned in Section 2.1. Therefore, it is necessary to consider the influence of the actual robotic fish volume and its motion constraints on the path planning task. In the proposed robotic fish path planning method, the fitness function considers the path length, obstacle avoidance constraints and the maximum curvature constraints of the path curve. The goal of considering the path length is to minimize the movement time and energy consumption as much as possible. The obstacle avoidance constraint aims to ensure that the robotic fish will not collide with obstacles. The purpose of considering the maximum curvature constraint of the path curve is to avoid generating sharp turnings which cannot meet the minimum turning radius of the robotic fish to facilitate the path following control.

Considering that robotic fish's planning and control modelling process in a three-dimensional environment is relatively complex, we plan to build a two-dimensional simulation environment model to verify our planning method. In addition, the subsequent path following control will be based on the two-dimensional reference path given by the planning task. In general, the path is determined by several control points, and then a series of coordinate points are obtained through cubic spline interpolation. These coordinate points are described as: (x_i, y_i), for $i = 0, 1, \ldots, n$. The path length is taken as the main part of the fitness function of BSO. The length of the path can be expressed as follows:

$$L = \sum_{i=1}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}. \tag{4}$$

The optimization problem of path planning is minimizing L.

2.3.1. *Obstacle avoidance constraint*

Circles describe the obstacles in our environmental model. Therefore, the collision can be determined by judging the relationship between the distance from each point on the path to the center of the obstacle and the radius of the obstacle. In other words, if the distance from the point on the path to the center of the obstacle is greater than the radius of the obstacle, there will be no collision. Otherwise, there will be a collision. Furthermore, considering the volume of the robotic fish itself, it is necessary to carry out corresponding expansion treatment for obstacles. In general, the center of mass of a robotic fish is located between 1/3 and 1/2 of the fish's body. Here we choose two-thirds of the body length of the robotic fish as the expansion radius of the obstacle. The obstacle expansion processing of the planning module can also help the path following controller to cover a certain number of errors in the following path following tasks.

The average degree of all coordinate points in the moving path of robotic fish in the obstacle is taken as the penalty term of the fitness function. The penalty function term is expressed as follows:

$$V = \sum_{i=1}^{n} \mathrm{mean}(v_{ik}), i = 0, 1, \ldots, n, k = 1, 2, \ldots, m, \tag{5}$$

$$v_{ik} = \max\left\{1 - \frac{\sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}}{r_k + \frac{2}{3}l}, 0\right\} \tag{6}$$

where r_k is the actual radius of obstacle k, and l is the body length of the robotic fish. x_k and y_k are respectively the horizontal and vertical coordinates of coordinate point i on the path, x_k and y_k are the coordinates of the center of obstacle k. max{.} is the maximum function, mean(.) is the mean function.

2.3.2. *Maximum curvature constraint*

In theory, the minimum turning radius r_{min} of robotic fish is its own property, so the maximum curvature $\kappa_{max} = \frac{1}{r_{min}}$ of the moving path of robotic fish is fixed. We define that the curvature of the coordinate point i on the robotic fish path is κ_i (The curvature of the only circle determined by the point i and its adjacent two coordinate points). As shown in Fig. 4, it can be calculated by Eq. (7).

$$\kappa_i = \frac{1}{r} = \frac{2\sin\theta}{c} \tag{7}$$

Like the obstacle avoidance constraint, the maximum curvature constraint is also added to the fitness function as a

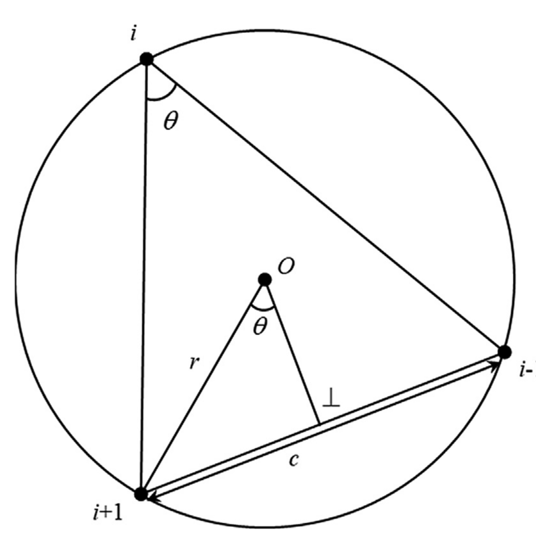


Fig. 4 – Diagram for calculating curvature.

penalty term. The penalty function term of maximum curvature constraint is designed as a step function:

$$H = \begin{cases} 1, \max\{\kappa_i\} > \kappa_{\max}, \\ 0, \max\{\kappa_i\} \leqslant \kappa_{\max}. \end{cases} \tag{8}$$

Add obstacle avoidance constraint and maximum curvature constraint as penalty function term to fitness function. The complete form of the fitness function with penalty function term can be obtained:

$$C = L + \alpha V + \beta H, \tag{9}$$

where, α is the penalty coefficient of the obstacle avoidance constraint penalty function term. β is the penalty coefficient of the maximum curvature constraint penalty function term.

2.4. *Path following based on MPC*

MPC is a model-based closed-loop optimal control strategy. Compared with the traditional PID control, the MPC can optimize and predict. MPC can overcome the velocity jump problem in the navigation of robotic fish to a certain extent and has a better control effect theoretically. The MPC based robotic fish path following method can compensate for the errors in the model and iteratively replan and correct the errors. The core of the MPC algorithm is the dynamic prediction model, which can predict the future, the control strategy, which is optimized and implemented repeatedly online, and the feedback correction of model error. Therefore, the central part of the MPC framework is the predictive model and the objective function of optimal control.

2.4.1. *The prediction model*

Prediction model is the basis of MPC. Its main function is to predict the state of the robotic fish next time according to the current state and action. In this paper, the dynamic prediction model of the MPC method is realized by multilayer perceptron (MLP). By making the robotic fish move randomly in the simulated environment and recording the track of length T, the random track data set $\tau = (s_0, a_0, \ldots, s_{T-2}, a_{T-2}, s_{T-1})$ is obtained. The state s here includes the position and velocity information of the two joints of the robotic fish, as well as the position and velocity information of the robot fish's center of mass. a is the action of the robotic fish. The collected track $\{\tau\}$ is divided into training data pairs, the input (s_t, a_t) and the corresponding output label s_{t+1}. As shown in Fig. 5, the MLP takes the current state s_t and action a_t of the robotic fish as input, and then outputs the predicted state $\hat{s}_{t+1}$ at the next time. During the training, the stochastic gradient descent method is used to train the MLP by minimizing the error between the prediction model output and the label. The gait characteristics of robotic fish in the environment can be understood by learning the prediction model to overcome the complexity of the robotic fish motion model and the uncertainty of the underwater motion environment.

2.4.2. *Process of optimal control*

The purpose of rolling optimization is to find the optimal control solution, which is an online optimization. It optimizes the control input over a short period to minimize the gap between the predictive model output and the reference. The rolling optimization process enables the MPC method to deal robustly with the uncertainties in the learned dynamic model.

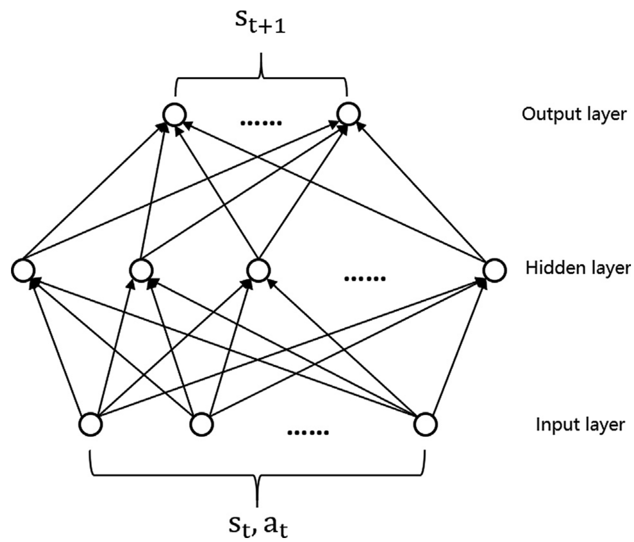


Fig. 5 – MLP prediction model.

In order to use the learned MLP prediction model to accomplish the specific task, we need to define an objective function to encode the task. For the robotic fish path following task, we construct an objective function, which encourages the robotic fish's center of mass to be close to the reference path, and encourages the robotic fish to move forward along the reference path with appropriate heading angle. We define the objective function of the time step t as:

$$F(\hat{s}_{t'}) = \lambda F_{\perp}(\hat{s}_{t'}) + \mu F_{\parallel}(\hat{s}_{t'}) + \nu F_{\angle}(\hat{s}_{t'}), t' = t, t+1, \cdots, t+n, \tag{10}$$

where, $F_{\perp}(\hat{s}_{t'})$ is the lateral error term, $F_{\parallel}(\hat{s}_{t'})$ is the longitudinal displacement term, and $F_{\angle}(\hat{s}_{t'})$ is the heading error term. Considering the characteristics of different reference paths, we set different weight coefficients for the above three terms according to the curvature of each coordinate point on the path. When the path curvature is small, the requirement for the robotic fish to approach the reference path is higher, so we set a larger value of λ. When the path curvature is large, that is, when the robotic fish needs to move in a steering motion, it will have higher requirements on the heading angle of the robotic fish, so we set a larger value of ν. By adjusting the weights according to path curvature, the MPC controller can better adapt to different reference paths.

The action of robotic fish is selected by a sample-based method. At each time step t, the state of robotic fish is s_t, and a finite horizon H in the future is planned by randomly generating several candidate action sequences. This process recalculates the control quantity according to the current error, predicts the results of these action sequences by using the learned MLP prediction model, and then selects the control sequence $A_t^{(H)}$ that optimizes the objective function.

$$A_t^{(H)} = (a_t, \cdots, a_{t+H-1}) = \arg F^*_{a_t,\cdots,a_{t+H-1}}(\hat{s}_{t'}), t' = t, t+1, \cdots, t+H-1. \tag{11}$$

Algorithm 1: Path following based on MPC.

1: Input the learned MLP prediction model M
2: **repeat**
3: Get the current state s_t.
4: Get the reference path obtained by the path planner.
5: Randomly generate several candidate action sequences $a_{t'}$.
6: $\hat{s}_{t'} = M(s_t, a_{t'})$
7: Calculate the mean curvature $\bar{\kappa_t}$ of the three nearest reference path coordinate points such $\bar{\kappa_t} = \frac{1}{3}(\kappa_t + \kappa_{t+1} + \kappa_{t+2})$
8: Adjust the weights (λ, μ, ν) according to the curvature $\bar{\kappa_t}$.
9: Select the optimal action sequence $A_t^{(H)}$ according to Eq. (10) and Eq. (11).
10: Perform the first action a_t in $A_t^{(H)}$.
11: **until** the target position is reached.

This optimization problem is solved at each time step t, and then only the first control action a_t in the control sequence is performed to transition to the next state s_{t+1}. Then in the next time step $t+1$, the updated actual state information of the robotic fish is used for the output of the predicted results of the prediction model, in order to prevent the mismatch of the prediction model or external interference from causing too large a gap between the control output and the expectation. A new control sequence is obtained through optimization. This process is repeated to solve the constrained optimization problem on the basis of rolling, to realize the continuous path following control of the robotic fish.

The MLP prediction model M and the optimized objective function F are used to construct the MPC controller. The pseudocode of the path following based on MPC is given in Algorithm 1.

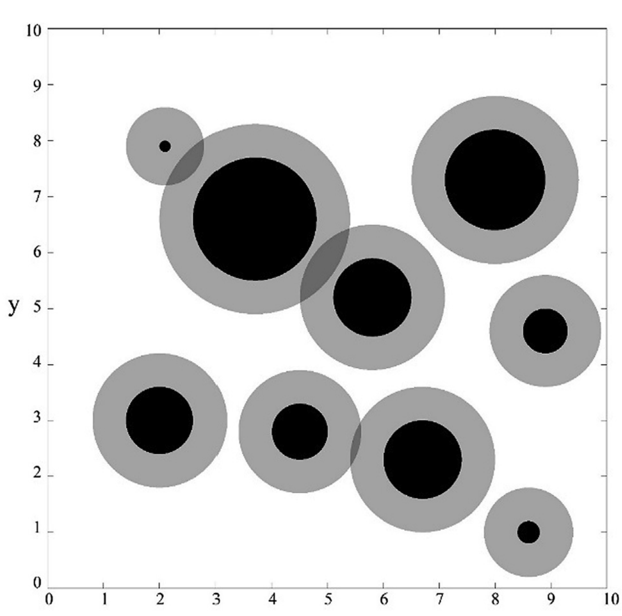


Fig. 6 – Environment model.

3. Results and discussion

3.1. *Path planning experiment results*

A 10 × 10 two-dimensional environment model is established for the simulation experiment of path planning of robotic fish, as shown in Fig. 6. The obstacles in the environmental model are described by circles. The starting position is set to (0, 0), and the target position is set to (10, 10).

The parameters of the BSO algorithm are set as $\omega = 1, c_1 = c_1 = 1.5, \lambda = 0.5, \delta^k = 0.8, d = 3$. Assume that the

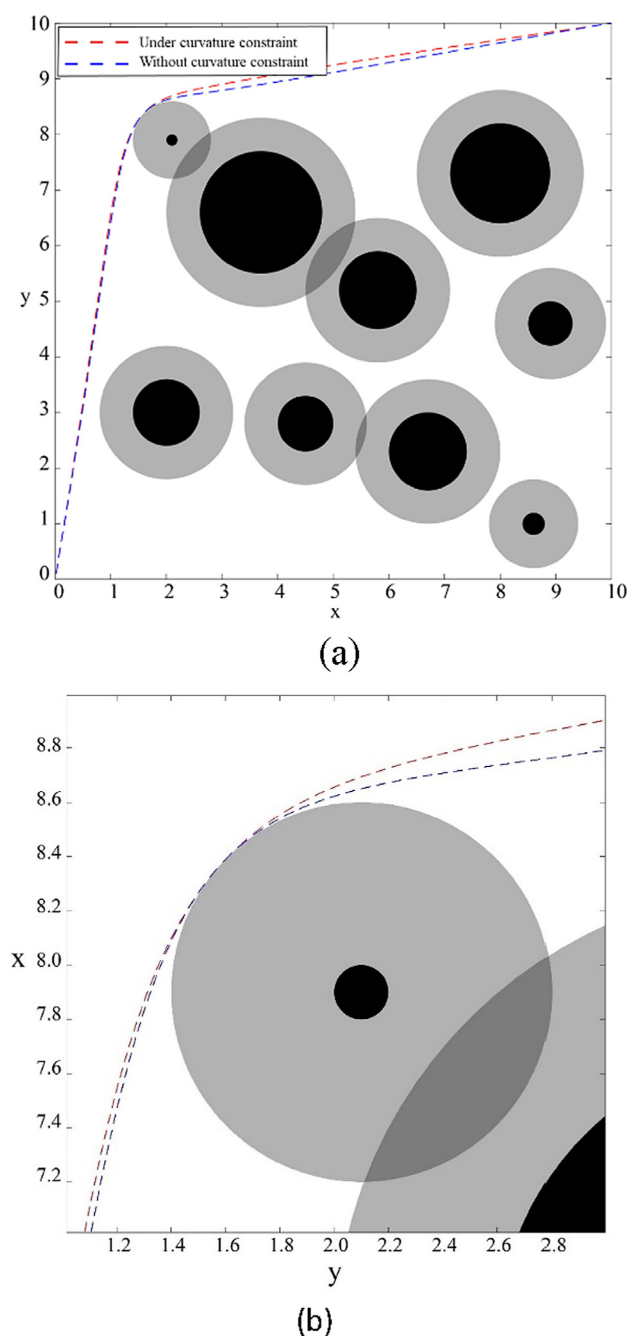


Fig. 7 – Path planning results. (a) Path planning results under and without curvature constraint; (b) Locally enlarged view of (a).

maximum curvature of the moving path of the robotic fish is $\kappa_{\max} = 1$.

The results of path planning are shown in Fig. 7, Fig. 8, and Fig. 9. The planning results show that the path planner can obtain the feasible path. Although the path length without the curvature constraint is slightly smaller than the path length under the curvature constraint, it cannot meet the minimum turning radius constraint of robotic fish, which is not beneficial to the subsequent completion of the robotic fish path following control task. Therefore, the planning result obtained under the curvature constraint is selected as the reference path of the path following task.

3.2. *Path following experiment results*

In this part, the path following experiments are carried out under the condition of fixed parameters of the optimal control objective function and dynamic adjustment of parameters according to the curvature of reference path.

The results of the path following experiments are shown in Fig. 10. Experimental results show that the proposed path following the controller can follow the reference path within a certain error range. In addition, it can be seen that the path following controller whose objective function parameters are dynamically adjusted according to the reference path curvature has a better following effect.

3.3. *Discussion*

It can be seen from Fig. 11 and Fig. 12 that the path following controller whose objective function parameters are dynamically adjusted according to the curvature of the reference path can better adapt to the part of the reference path with different curvatures. When the reference path is transferred from the straight part to the turning part, the dynamic adjustment objective function makes the controlled object robotic fish have a lower heading error. When the reference path changes from the turning part to the straight part, the robotic fish can quickly reduce the lateral error.

However, we notice that when the curvature of the reference path changes significantly, the path following effect of the MPC controller is not as good as that of the straight part with stable curvature. This may be because the experience-based adjustment of the optimization control objective func-

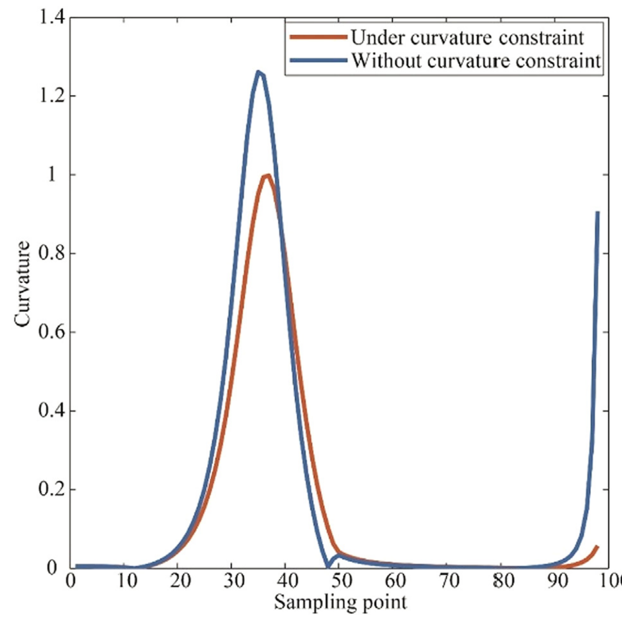


Fig. 9 – Curvature of planning results.

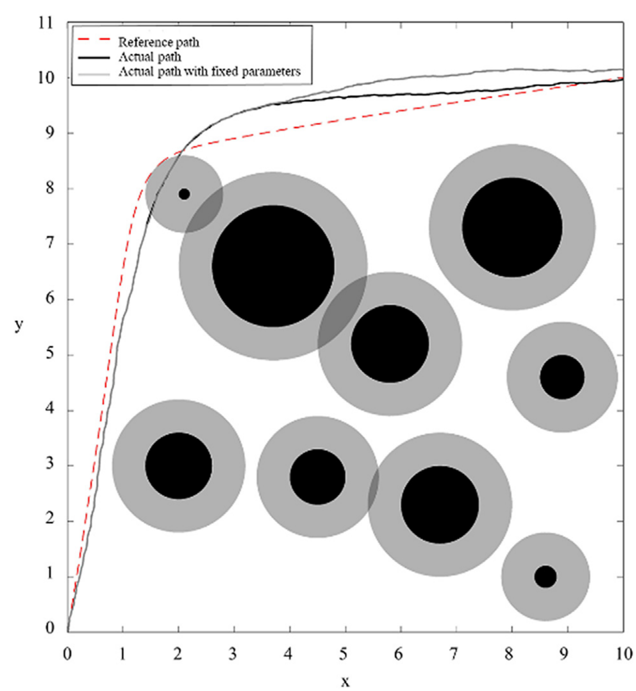


Fig. 10 – Path following results.

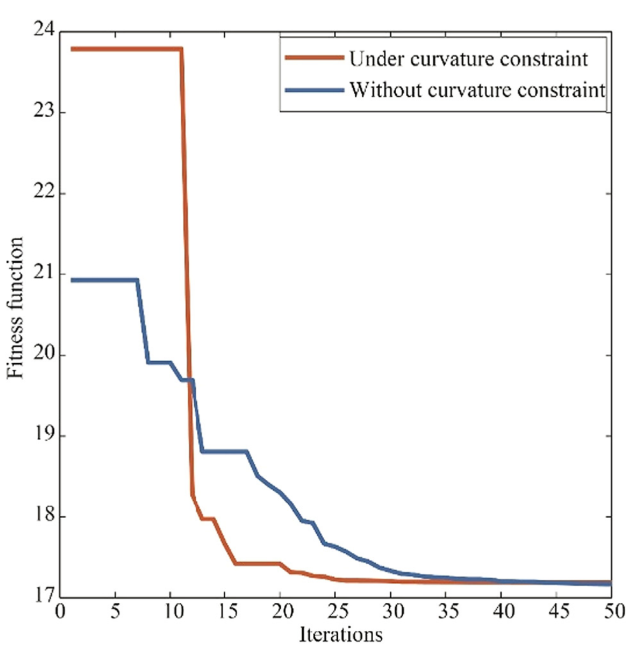


Fig. 8 – The fitness function converges.

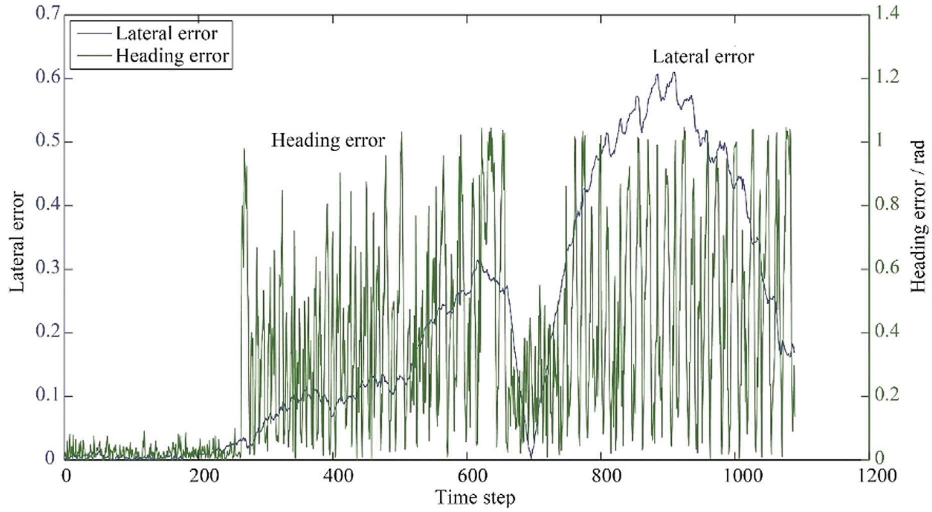


Fig. 11 – Path following lateral error and course error under fixed parameters of optimal control objective function.

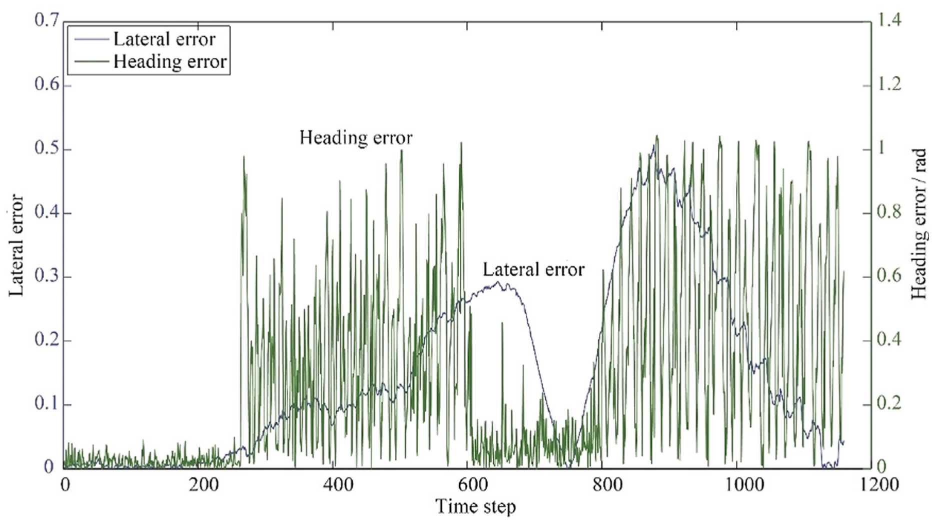


Fig. 12 – The lateral error and heading error of path following when the objective function parameters are dynamically adjusted according to the curvature of the reference path.

tion parameters is not intelligent. Therefore, it is necessary further to study the adaptive adjustment method of objective function parameters to make the MPC path following controller have better robustness and higher reliability.

4. Conclusion

This paper studies the path planning and path following control of robotic fish. The main innovations are as follows: A path planning method considering the volume and motion constraints of robotic fish is proposed, which can avoid collision risk and meet the constraint of the minimum turning radius of robotic fish. The optimal control method of learning-based model predictive control is adopted to dynamically adjust the optimal control objective function according to the curvature of the reference path to complete the robotic fish path following task. The simulation results show that the proposed planning and control method can effectively complete the path planning and path following control tasks of robotic fish in a two-dimensional environment. The physical morphology and motion characteristics of robotic fish are considered, thus avoiding the complex kinematics modelling process. However, the specific disturbance caused by ocean current and waves in the underwater environment, which will focus on future research, is not considered. In addition, the proposed control method can be further combined with adaptive control to improve the stability and robustness of the control system.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61903007, and in part by the National Key Research and Development Program of China under Grant 2019YFD0901000.

REFERENCES

[1] Hu J, Mei J, Chen D, Li L, Cheng Z. Path planning of robotic fish in unknown environment with improved reinforcement learning algorithm. In: International Conference on Internet and Distributed Computing Systems. Springer; 2018. p. 248–57.
[2] Yu Z, Tao J, Xiong J, Luo A, Yang SX. Neural-Dynamics-Based Path Planning of a Bionic Robotic Fish. In: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE; 2019. p. 1803–8.
[3] Hou N, Wang H, Yu M, Chen L, Cao Z, Zheng J, et al. Robotic fish path planning in complex environment. In: 2019 Chinese Control Conference (CCC). IEEE; 2019. p. 4519–24.
[4] Khan S, Javed S, Naeem N, Ali I. Performance analysis of PID and state-feedback controller on the depth control of a robotic fish. In: 2017 International Conference on Frontiers of Information Technology (FIT). IEEE; 2017. p. 7–11.
[5] Wang H, Mi C, Li Z, Hou N, Xie G. A novel data-assisted model and discrete-time sliding mode steering controller of robotic fish. In: 2018 IEEE International Conference on Mechatronics, Robotics and Automation (ICMRA). IEEE; 2018. p. 47–51.
[6] Song X, Gao S, Chen C, Gao Z. Enhanced Fireworks Algorithm-Auto Disturbance Rejection Control Algorithm for Robot Fish Path Tracking. Int J Comput Commun. 2019;14 (3):401–18.
[7] Bal C, Ozmen Koca G, Korkmaz D, Akpolat ZH, Ay M. CPG-based autonomous swimming control for multi-tasks of a biomimetic robotic fish. Ocean Eng 2019;189:106334. https://doi.org/10.1016/j.oceaneng.2019.106334.
[8] Du S, Zhou C, Yu J, Wu Z. A Modified Line-of-Sight Method for Path Tracking Applied to Robotic Fish. In: 2020 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE; 2020. p. 809–14.
[9] Liu J, Liu Z, Yu J. Line-of-sight based three-dimensional path following control for an underactuated robotic dolphin. Inform Sci 2021;64:1–112210.
[10] Yu J, Wang M, Dong H, Zhang Y, Wu Z. Motion control and motion coordination of bionic robotic fish: A review. J Bionic Eng 2018;15(4):579–98.
[11] Nagabandi A, Kahn G, Fearing RS, Levine S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2018. p. 7559–66.
[12] Xie S, Garofano V, Chu X, Negenborn RR. Model predictive ship collision avoidance based on Q-learning beetle swarm antenna search and neural networks. Ocean Eng 2019;193:106609. https://doi.org/10.1016/j.oceaneng.2019.106609.
[13] Ma L, Yue Z, Zhang R. Path Tracking Control of Hybrid-driven Robotic Fish Based on Deep Reinforcement Learning. In: 2020 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE; 2020. p. 815–20.
[14] Wang M, Zhang Y, Dong H, Yu J. Trajectory tracking control of a bionic robotic fish based on iterative learning. Sci China Inform Sci 2020;63:1–9.
[15] Yu JunZhi, Wen L, Ren ZiYu. A survey on fabrication, control, and hydrodynamic function of biomimetic robotic fish. Sci China Technol Sci 2017;60(9):1365–80.
[16] Todorov E, Erez T, Mujoco TY. A physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; 2012. p. 5026–33.
[17] Sgorbissa A. Integrated robot planning, path following, and obstacle avoidance in two and three dimensions: Wheeled robots, underwater vehicles, and multicopters. Int J Robot Res 2019;38(7):853–76.
[18] Shen C, Shi Y, Buckham B. Integrated path planning and tracking control of an AUV: A unified receding horizon optimization approach. IEEE/ASME Trans Mechatron 2017;22 (3):1163–73.
[19] Wang X, Yao X, Zhang L. Path planning under constraints and path following control of autonomous underwater vehicle with dynamical uncertainties and wave disturbances. J Intell Robot Syst 2020:1–18.
[20] Ulyanov S, Bychkov I, Maksimkin N. Event-Based Path-Planning and Path-Following in Unknown Environments for Underactuated Autonomous Underwater Vehicles. Appl Sci 2020;10(21):7894. https://doi.org/10.3390/app10217894.
[21] Castaño ML, Mavrommati A, Murphey TD, Tan X. Trajectory planning and tracking of robotic fish using ergodic exploration. In: 2017 American Control Conference (ACC). IEEE; 2017. p. 5476–81.
[22] Lu Y, Chen X, Wu Z, Yan Y, Yu J. A Vision-Based Path Planning and Following System for a Miniature Robotic Fish. In: 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR). IEEE; 2019. p. 408–13.
[23] Wang T, Yang L. Beetle swarm optimization algorithm: Theory and application. arXiv preprint arXiv:1808.00206; 2018.
[24] Jiang X, Li S. BAS: Beetle antennae search algorithm for optimization problems. arXiv 2017. arXiv preprint arXiv:1710.10724.
[25] Mu Y, Li B, An D, Wei Y. Three-dimensional route planning based on the beetle swarm optimization algorithm. IEEE Access 2019;7:117804–13.