



Article

Target-Following Control of a Biomimetic Autonomous System Based on Predictive Reinforcement Learning

Yu Wang¹, Jian Wang^{2,3} , Song Kang^{2,3} and Junzhi Yu^{2,4,*}

- ¹ Department of Automation, Tsinghua University, Beijing 100084, China; yu_wang@tsinghua.edu.cn
² The Laboratory of Cognitive and Decision Intelligence for Complex System, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China; jianwang@ia.ac.cn (J.W.)
³ The School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
⁴ The State Key Laboratory for Turbulence and Complex Systems, Department of Advanced Manufacturing and Robotics, College of Engineering, Peking University, Beijing 100871, China
* Correspondence: junzhi.yu@ia.ac.cn

Abstract: Biological fish often swim in a schooling manner, the mechanism of which comes from the fact that these schooling movements can improve the fishes' hydrodynamic efficiency. Inspired by this phenomenon, a target-following control framework for a biomimetic autonomous system is proposed in this paper. Firstly, a following motion model is established based on the mechanism of fish schooling swimming, in which the follower robotic fish keeps a certain distance and orientation from the leader robotic fish. Second, by incorporating a predictive concept into reinforcement learning, a predictive deep deterministic policy gradient-following controller is provided with the normalized state space, action space, reward, and prediction design. It can avoid overshoot to a certain extent. A nonlinear model predictive controller is designed and can be selected for the follower robotic fish, together with the predictive reinforcement learning. Finally, extensive simulations are conducted, including the fix point and dynamic target following for single robotic fish, as well as cooperative following with the leader robotic fish. The obtained results indicate the effectiveness of the proposed methods, providing a valuable sight for the cooperative control of underwater robots to explore the ocean.

Keywords: biomimetic motion; biomimetic autonomous system; target following; deep reinforcement learning; predictive control



Citation: Wang, Y.; Wang, J.; Kang, S.; Yu, J. Target-Following Control of a Biomimetic Autonomous System Based on Predictive Reinforcement Learning. *Biomimetics* **2024**, *9*, 33. <https://doi.org/10.3390/biomimetics9010033>

Academic Editor: Jinyou Shao

Received: 13 November 2023

Revised: 16 December 2023

Accepted: 2 January 2024

Published: 4 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid development of science and technology, the field of biomimetics has witnessed substantial advance in recent years, garnering widespread attention from researchers globally. With a long history of natural evolution, biological fish have acquired remarkable motion capabilities. They demonstrate proficiency in executing various acrobatic maneuvers, including rapid start and stop, inter-media leaping, and other sophisticated actions. Additionally, natural fish can achieve hydrodynamic drag reduction and energy efficiency through the utilization of diverse biological mechanisms.

Inspired by nature, the biomimetic robotic fish has attracted the attention of many scientists and engineers [1–6]. Katzschmann et al. designed a new kind of soft robotic fish named Sofi, which can interact with remote control personnel through ultrasonic communication module and realize the close observation of underwater organisms with a maximum diving depth of 18 m [1]. By mimicking yellowfin tuna, White et al. developed a series of high-speed biomimetic robotic tuna [7,8], and achieved tail-fin flapping frequencies of up to 15 Hz. To quantify the role of body flexibility in high-speed swimming, they further presented Tunabot Flex, which could achieve a high swimming speed of 4.6 BL/s (body length per second, BL/s) at a swing frequency of 8 Hz. Yu et al. designed a robotic dolphin. Based on the motion control strategies, a high swimming speed of 2.05 m/s (2.85 BL/s)

was achieved, and the action of repetitive leaping similar to that of biological dolphins was completed [9]. Compared with the traditional autonomous underwater vehicles (AUVs), the biomimetic robotic fish has the advantages of high maneuverability, strong concealment, and better biocompatibility. These attributes underscore their considerable potential for application across diverse domains.

More importantly, by learning the motion mechanism of biological fish, the movement performance of the robotic fish can be further improved, e.g., by using fish schooling movement to save energy. It is hypothesized that fish schooling movements can improve the hydrodynamic efficiency via a certain swimming mode [10]. In a school of fish, the leader fish tend to consume more energy than the follower fish. The eddy current generated by the tail of leader fish can provide a certain amount of water power to the follower, thus achieving the effect of energy saving. This movement has a similar phenomenon in birds. As a result, biological fish tend to travel in flocks for long-distance voyages. Many researchers have investigated the fish schooling mechanism. Li et al. focused on how the fish plan their movement to save energy and achieve larger thrust from the vortices generated by others [11,12]. They designed some bionic robotic fish to measure the energy consumption when the robotic fish swim in the pool. Further, a vortex phase-matching strategy was obtained, indicating that the schooling fish exhibited a tailbeat phase difference that varied linearly with front-back distance. They also found that when the fish swim side by side, an individual could improve its efficiency if they changed the tailbeat phase to a certain angle, such as 0.25π . By measuring the actual movement of 15 fish, Marras et al. found that compared with the individual swimming, the schooling fish in any position can save energy, and the fish swimming behind the neighbours showed the best performance [13]. Thandiackal et al. conducted an interesting experiment to observe the movement of natural trout when it interacted with the thrust wakes via a robotic mechanism. The results illustrated that the trout exhibited reduced pressure drag, further proving the energy saving [14]. Li et al. investigated the pressure and vorticity fields between a single fish and a pair of fish, and offered some results. However, there are insufficient conclusions about motion benefits of fish schooling [15]. Dai et al. investigated a variety of stable formations with the schooling of two, three and four self-propelled fish-like swimmers, and examined the energy efficiency of each formation [16]. Verma et al. explained the energy-saving mechanism in the schooling behavior of fish. By combining deep reinforcement learning and fluid simulation, an energy-saving strategy was proposed, which enabled followers to save energy by using vortices in the leader's wake [17]. More studies can be found in [18].

With regard to the target following control, there are many related results. Dai et al. [19] designed a robust tube model predictive controller (MPC) with an extended Kalman filter target observer for an underwater vehicle-manipulator system, specifically tailored to address the challenge of capturing moving targets. Cui et al. [20] proposed an optimal trajectory tracking method for AUVs, applying reinforcement learning techniques with critic and action neural networks. He et al. [21] formulated asynchronous multithreading proximal policy optimization-based algorithms to tackle issues related to path planning and trajectory tracking in unmanned underwater vehicles. Jiang et al. [22] introduced a model-free attention-based, model-agnostic meta-learning algorithm for AUVs, demonstrating efficacy in achieving high-precision tracking tasks. Zou et al. [23] designed an image-guided motion controller which consists of a genetic algorithm-based linear quadratic regulator velocity controller and direction controller to realize mobile target following for micro-robotic swarms. Yan et al. [24] designed a reinforcement learning and orthogonal fractional factorial design-based tracking controller for AUVs to enhance the scalability of uncertainty evaluation. Shi et al. [25] applied a hybrid actors-critics architecture to improve following control accuracy of AUVs. Gao et al. [26] introduced a fixed-time resilient cooperative edge-triggered estimation and control framework designed to facilitate cooperative target tracking for unmanned surface vehicles (USVs). Wai et al. [27] constructed an adaptive following control scheme with a dynamic recurrent fuzzy neural network that allowed

the vision-based mobile robot to track the moving target. Huang et al. [28] designed a homography-based visual servo controller that allowed the unmanned aerial vehicle to track the moving ship trajectory. Lin et al. [29] designed an image-based visual servoing geometric controller for quadrotors for tracking the desired trajectory.

Our work is motivated by the collective motion observed in fish schooling, where a phenomenon is known to enhance motion efficiency and reduce energy consumption. The studies on fish schooling movement can be theoretically categorized into kinematic and behavioral levels. At the kinematic level, the studies primarily focus on the macroscopic distances and swimming postures among fish. By closely observing the swimming patterns within fish schools, optimal distances and directions can be discerned. Further, at the behavioral level, greater emphasis may be placed on the individual body swimming postures of fish. For instance, in the context of tail-wagging fish swimming, it becomes imperative to synthesize parameters such as tail-wagging frequency, amplitude, and phase differences within the fish school. In this paper, we focus on the kinematic level, wherein we translate the complexities of fish schooling swimming into the pursuit of target position and attitude, thereby providing foundational support for biomimetic research at the behavioral level. The primary contributions of this paper can be concluded in three aspects.

- Inspired by fish schooling movement, we focus on the kinematic level and propose a target following control framework, including a predictive deep deterministic policy gradient controller (PDDPG) and nonlinear model predictive controller (NMPC).
- Aiming to address the hysteresis characteristics of following control for the robotic fish, we introduce the predictive concept into the deep deterministic policy gradient method. By predicting the future state and adding it to the buffer pool, we effectively mitigate the overshooting phenomenon during the tracking process. Furthermore, the state space is intentionally designed in a normalized manner, concurrently featuring a multi-objective optimization reward function.
- Taking the kinematic and dynamic models as the predictive model, we derive the nonlinear model predictive control law with full consideration of the stage cost and terminal cost. Extensive simulations are carried out to verify the effectiveness of the proposed PDDPG and NMPC methods.

The subsequent sections of this paper are structured as follows. Section 2 provides an exposition of the problem statement and the control framework. Section 3 delves into the target-following control methodologies, encompassing the predictive deep deterministic policy gradient controller and the nonlinear model predictive controller. Furthermore, Section 4 presents simulation results, followed by a comprehensive analysis. Finally, Section 5 offers concluding remarks to summarize the paper.

2. Problem Statement and Control Framework

In this section, we commence by introducing the target-following task and succinctly present the kinematic and dynamic models pertaining to the underwater robot. Subsequently, aligned with the specified task, we formulate a target-following framework for the robotic fish. Additionally, an alternative movement strategy is proposed, which integrates both the predictive deep deterministic policy gradient controller and the nonlinear model predictive controller.

In view of the observations that fish schooling movement in nature can save energy consumption and improve movement efficiency, this paper aims to design a target-following control method for the robotic fish. As shown in Figure 1, by selecting a robotic fish as the leader, a path cruise task that involves setting target points can be executed. $P_l = (x_l, y_l)$ denotes the real-time position of the leader robotic fish while φ_l is its yaw angle. d_g and φ_g indicate the Euclidean distance and relative direction between the leader robotic fish and the target points, respectively. $\varphi_{lg} = \varphi_l - \varphi_g$ illustrates the attitude direction difference between the leader robotic fish and the target points.

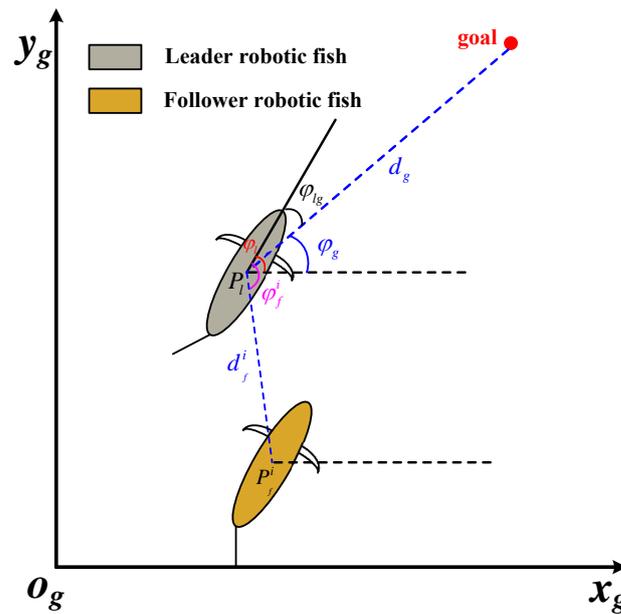


Figure 1. The illustration of the target-following task and coordinate system definition.

Furthermore, we set some robotic fish as followers. $P_f^i = (x_f^i, y_f^i)$ denotes the real-time position of the follower robotic fish, where $i = 1, 2, \dots, n$. The purpose of the following task is to maintain the set target distance d_f^i and direction φ_f^i between the follower and leader robotic fish. Given the aforementioned variables, the target position point for the follower can be obtained as follows:

$$\begin{aligned} x_f^i &= x_l + d_f^i \cos(\varphi_f^i - \varphi_l) \\ y_f^i &= y_l + d_f^i \sin(\varphi_f^i - \varphi_l) \end{aligned} \tag{1}$$

It should be noted that the target point undergoes real-time variations with the movements of the leader. Consequently, for the follower, this task is characterized as a dynamic following mission. One of the principal improvements of this study lies in the approach based on a deep reinforcement learning framework. During training, static target-following scenarios are employed, yet the method is endowed with the capability to dynamically follow targets. Besides, for each single robotic fish, we offer the kinematic model for the following control, which can be formulated by

$$\dot{x} = u \cos \varphi - v \sin \varphi, \tag{2}$$

$$\dot{y} = u \sin \varphi + v \cos \varphi, \tag{3}$$

$$\dot{\varphi} = r, \tag{4}$$

where $p(t) = (x(t), y(t), \varphi(t))$ represent the position and yaw angle with respect to inertia frame, respectively. (u, v, r) denote the linear and angular velocities with respect to body frame, respectively. Thereafter, the dynamic model can be derived as follows:

$$\dot{u} = \frac{1}{m_{11}}(\tau_u + m_{22}vr - d_{11}u), \tag{5}$$

$$\dot{v} = \frac{1}{m_{22}}(-m_{11}ur - d_{22}v), \tag{6}$$

$$\dot{r} = \frac{1}{m_{33}}(\tau_r + (m_{11} - m_{22})uv - d_{33}r), \tag{7}$$

where (m_{11}, m_{22}, m_{33}) and (d_{11}, d_{22}, d_{33}) are the mass and damping parameters larger than zero. τ_u and τ_r indicate the thrust force and yaw moment, respectively.

Therefore, based on the aforementioned problem statement, this paper proposes a target-following control framework, as illustrated in Figure 2. The framework comprises a biomimetic autonomous system consisting of one leader robotic fish (referred to as Agent_0) and multiple follower robotic fish. Initially, a cruising target point is set for the leader robotic fish. Utilizing the proposed PDDPG controller, real-time control force and moment can be generated to achieve cruising control. It is noteworthy that although these target points are discrete, if they are relatively close, an effect similar to continuous path following can be achieved. Subsequently, through a task allocation module based on the principles of natural fish, real-time target following positions can be determined for each follower robotic fish. Furthermore, for each follower robotic fish, a controller selector is designed, corresponding to the proposed PDDPG controller and NMPC, to output real-time control force and moment for target following. It is essential to emphasize the dual purposes of designing the selector. On one hand, the PDDPG controller exhibits strong environmental adaptability and scalability, suggesting that it has an advantage over NMPC when introducing random obstacle avoidance tasks in this mission. However, NMPC features stable solution finding and smooth motion output, contributing to improved control stability. On the other hand, in this paper, the two controllers are independent and are switched through a toggle switch. In practice, they can be organically integrated based on certain principles, such as event-triggered mechanisms, according to different task scenarios.

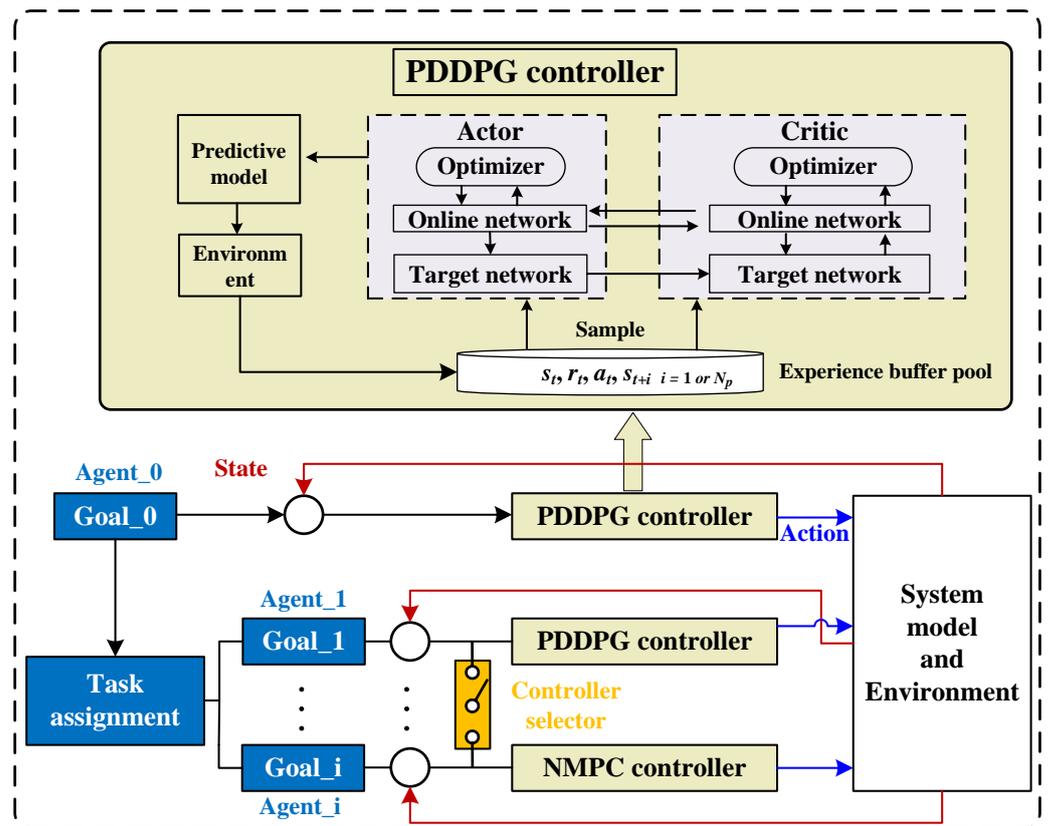


Figure 2. The target following control framework.

3. The Methodology of Target Following Control

3.1. The Predictive Deep Deterministic Policy Gradient Controller

In consideration of the dynamic characteristics and strong interference in underwater environments, this section introduces a following controller based on a predictive deep

deterministic policy gradient. Firstly, the target-following problem is formulated as a multi-objective optimization issue through the design of network architecture, state space, action space, and reward function. Secondly, to enhance the training performance, normalization scaling is applied to the designed state variables, and reward values are standardized. More importantly, due to the highly nonlinear nature exhibited by biomimetic robotic fish, traditional control methods often result in following overshooting. To address this issue, the predictive approach is incorporated into the conventional deep deterministic policy gradient (DDPG) method. Specifically, when certain conditions are met, state variables and reward values after N_p steps are calculated and stored in a buffer pool. During the testing stage, actions can be output from the network based on the state variables after N_p steps. This technique can expand the training space to some extent, effectively avoiding overshooting without increasing the state space, thereby reducing network complexity.

First, the target-following task can be regarded as a Markov decision process. The tuple data comprises the state variable $s \in \mathcal{S}$, the action variable $a \in \mathcal{A}$, the reward function $\mathcal{R}(s)$, the state transition function $\mathcal{F} : (s, a) \rightarrow s'$, and the discount factor γ . By inputting the current state into a neural network, the action variable can be outputted, leading to the transition to the next state. The optimization objective is to maximize the reward function, driving parameter updates in the neural network. In recent years, DDPG has come to stand out as a reinforcement learning algorithm that has garnered substantial interest in recent years due to its efficacy in addressing challenges associated with continuous action space problems. Its applications span diverse domains, including robotics, control, and various other fields. DDPG amalgamates concepts from both value-based and policy-based reinforcement learning, utilizing a dual-neural network architecture comprising the Critic and the Actor. In this paper, we introduce certain enhancements to the conventional DDPG framework to accomplish the target-following task.

For deep reinforcement learning algorithms, the selection of appropriate state and action spaces, along with the design of a suitable reward function, stands as a pivotal determinant of network performance. In the subsequent section, specific design methodologies will be elucidated.

3.1.1. State Space

In pursuit of reducing the complexity of deep neural networks, we exclusively focus on the design of two state variables as follows:

- d_{sg} : The state variable considered in this context pertains to the distance between the current position of the robot and the target point. This variable primarily ensures that the robot consistently approaches the target point at a predetermined velocity.
- φ_{sg} : The state variable involves the angular separation between the robot's current position and the target point. This variable is crucial for ensuring the robot's sustained alignment towards the target, serving the dual purpose of minimizing travel distance and maintaining a stable motion posture.

To enhance training performance and expedite convergence, we propose a normalization and scaling technique for a specific set of state variables. Firstly, we determine the maximum values for two state variables. In this study, the Euclidean distance between the robot's starting point (x_o, y_o) and the target point (x_g, y_g) is designated as the maximum value for d_{sg} , as follows:

$$d_{\max} = \sqrt{(x_o - x_g)^2 + (y_o - y_g)^2}. \quad (8)$$

As for φ_{\max} , we set it to 2π . Additionally, after normalization, we introduce a scale-up factor for two primary purposes. Firstly, post-normalization, the values fall within the $[0, 1]$ range, which might be unsuitable for network convergence. Hence, amplification is applied. Secondly, considering the difference in the physical interpretations of d_{sg} and φ_{sg} , it is necessary to balance their magnitudes for faster convergence when inputting into the network. It is noteworthy that the selection of d_{\max} is not a fixed value due to the real-time

variability of the follower robot's target. Hence, by dynamically updating the target point (x_g, y_g) for the follower robot, more effective action values can be obtained. Considering these aspects, the formulated state variables are expressed as follows:

$$\begin{aligned} \tilde{d}_{sg} &= k_1 \frac{d_{sg}}{d_{\max}} \\ \tilde{\varphi}_{sg} &= k_2 \frac{\varphi_{sg}}{\varphi_{\max}} \end{aligned} \quad (9)$$

3.1.2. Action Space

Based on the kinematic and dynamic models of biomimetic robotic fish, we define forward thrust and yaw moment as action variables. In contrast to conventional kinematic navigation approaches, we directly employ control quantities as actions, implying that it is necessary to traverse two layers of non-linear models, namely kinematics and dynamics, which increases the learning complexity. Additionally, building upon our previous work, we set the action ranges for forward thrust and yaw torque to $[0, 6 \text{ N}]$ and $[-6 \text{ Nm}, 6 \text{ Nm}]$, respectively. Moreover, since this study does not involve information exchange among robot swarms, the leader robotic fish does not decelerate when the distance between the leader and follower robotic fish is substantial. Hence, it is essential to ensure that the follower robotic fish possesses the ability to catch up with the leader, leading to the specification of the maximum forward thrust range for the follower robotic fish as $[0, 8 \text{ N}]$.

Furthermore, the forward thrust of the robotic fish is constrained to be consistently greater than zero, while the yaw torque exhibits bilateral symmetry. Therefore, a bilateral correction is applied to the forward thrust. Specifically, during both the network output and replay buffer storage phases, the range of forward thrust is adjusted to $[-3 \text{ N}, 3 \text{ N}]$. When inputted into the training environment, the forward thrust outputted by the network is increased by 3 N , rendering it unilaterally positive, and subsequently fed into the motion model to update the environmental information.

3.1.3. Reward Function

The reward function constitutes a pivotal element in deep reinforcement learning. With full consideration of path smoothness and length factors, a multi-objective optimization reward function is proposed as follows:

$$R = \sum_{i=1}^3 c_i r_i, \quad (10)$$

where c_i denotes the weight coefficients and r_i represents the different reward forms. There are three principles.

- The principle of minimum distance: it is expressed as $r_1 = -|\tilde{d}_{sg}|$, primarily employed to minimize the length of the motion path.
- The principle of directional convergence: it is expressed as $r_2 = -|\tilde{\varphi}_{sg}|$, with the aim of guiding the robot to orient itself towards the target point during motion. This principle not only contributes to the reduction in path length but also serves to ensure a certain degree of stability in the output yaw moment.
- The path-smoothing principle: it is characterized by $r_3 = -|\tilde{\varphi}_{sg} - \tilde{\varphi}'_{sg}|$, where $\tilde{\varphi}'_{sg}$ represents the previous time step's $\tilde{\varphi}_{sg}$. This principle primarily aims to enhance control stability by minimizing the yaw angle difference between consecutive time steps, thereby smoothing the motion path.

3.1.4. Predictive Concept-Based Improvement

In light of the aforementioned state space, action space, and reward function, it is evident that the learning objective of the proposed method is to achieve rapid and smooth target following. However, in practical tracking scenarios, due to the highly nonlinear

characteristic of the system, the steering of robotic fish exhibits a certain degree of lag, leading to the occurrence of overshooting phenomena. To address this issue, the most direct solution is to introduce angular velocity as an additional state variable and incorporate it into the learning network. Nevertheless, this approach encounters two primary challenges. Firstly, the introduction of angular velocity increases the complexity of the state space, thereby escalating the training difficulty. Secondly, in real-world applications, angular velocity information is typically obtained from inertial measurement unit (IMU) sensor modules, making it susceptible to external environmental factors and noise, manifesting notable information instability such as abrupt fluctuations. Therefore, angular velocity is deemed unsuitable as a state variable. With these considerations, this paper integrates a predictive approach into the traditional DDPG framework, presenting a novel training architecture. This structure effectively mitigates overshooting phenomena without introducing additional state variables.

In the conventional DDPG algorithm, the replay buffer stores a series of experience tuples (s_t, a_t, s_{t+1}, r_t) , including the representation of the state at the next time step s_{t+1} , i.e., $s_{t+1} = f(s_t, a_t)$. Here, f denotes a composite motion model of kinematics and dynamics. The key improvement in the proposed method lies in the replacement of the current state with the state quantity obtained after N_p steps when the steering angular velocity exceeds a predefined threshold. Both the state after N_p steps and the current state are then stored in the replay buffer, i.e., $(s_t, a_t, s_{t+N_p}, r_t)$. This process generates a sequence of states as

$$S = (s_{t+N_p}, s_{t+N_p-1}, \dots, s_{t+1}, s_t) \quad (11)$$

$$s_{t+i} = f(s_{t+i-1}, a_t)$$

Based on the above illustration, the calculation process of PDDPG is presented in Algorithm 1.

Algorithm 1 Algorithm for PDDPG

- 1: Initialize the parameters of Actor network and Critic network.
 - 2: Initialize the experience replay buffer pool.
 - 3: **for** episode = 1 to N **do**
 - 4: Reset the control system, and obtain the initial states s_0 .
 - 5: **for** step = 1 to M **do**
 - 6: According to the trained strategy, select the output action with the added noise information.
 - 7: Perform actions in the model environment.
 - 8: **if** $\dot{\varphi} > 20^\circ/s$ **then**
 - 9: Apply the predictive model, and calculate s_{t+N_p} .
 - 10: Put $(s_t, a_t, s_{t+N_p}, r_t)$ into buffer pool.
 - 11: **else**
 - 12: Based on the motion model, calculate s_{t+1} .
 - 13: Put (s_t, a_t, s_{t+1}, r_t) into buffer pool.
 - 14: **end if**
 - 15: Sample a subset of data from the experience replay buffer for network updating.
 - 16: Update the Critic network according to the loss.
 - 17: Update the Actor network based on deterministic policy gradient followed by the target Actor network.
 - 18: **end for**
 - 19: **end for**
-

3.2. The Nonlinear Model Predictive Controller

In recent years, the utilization of MPC has become increasingly prevalent within the domain of robotics. MPC proves valuable not only in tackling the intricacies of systems with multiple inputs and outputs, but also in addressing and managing control constraints

effectively. The control approach employed in this paper can be considered as a means of achieving setpoint stabilization to a certain extent. This is achieved by designing a controller that effectively stabilizes a predefined stationary setpoint.

Although the approach proposed in this paper also allows for dynamic target points to be followed, the switching of these dynamic targets is governed by specific triggering rules. On one hand, for the leader robotic fish, the target point switching criterion is based on the condition that the robotic fish is within a certain threshold distance from the target. As a result, this type of target point switching can be considered as setpoint stabilization on a time scale. On the other hand, for the follower robotic fish, the target points change in real time with the position of the leader robotic fish. However, due to the time-independence characteristic of target point locations during the controller design process, these changes can be simplified as setpoint stabilization. Furthermore, in order to achieve the following control, we need to stabilize the key variables, including the planar position and the yaw attitude. Hence, based on the kinematic and dynamic models of the robotic fish, the state variables can be selected as $P_f = (x, y, \varphi)$. Correspondingly, we consider the forward thrust and yaw moment as control variables, i.e., $u_f = (\tau_u, \tau_r)$.

This design implies the incorporation of both the kinematic and dynamic models as components of the predictive model, which simplifies the controller design process. However, the combination of two nonlinear models may introduce a degree of computational complexity and elevate optimization challenges. Iterative solution-seeking is necessary, and parameter adjustments are implemented to address these complexities. Therefore, by defining the reference $\Lambda_f = (x_d, y_d, \varphi_d)$ and error item e , we can consider the cost function as follows:

$$J(e(t_k), u_f) = \int_{t_k}^{t_k+T} L(e(\tau), u_f) d\tau + g(e(t_k + T)), \quad (12)$$

where L indicates the stage cost while g is the terminal penalty. T is the prediction horizon. Furthermore, the optimal control problem addressed at each sampling instant can be structured as follows:

$$\min_{u_f} J(e(t_k), u_f), \quad (13)$$

Subject to

$$e(t_k) = \tilde{\zeta}(x(t_k), y(t_k), \varphi(t_k), R_f), \quad (14)$$

$$L = e(\tau|t_k)^T Q e(\tau|t_k) + u_f(\tau|t_k)^T R u_f(\tau|t_k), \quad (15)$$

$$g = e(t_k + T|t_k)^T K e(t_k + T|t_k), \quad (16)$$

$$u_f \in [u_{f \min}, u_{f \max}], \quad (17)$$

where $\tilde{\zeta}$ can be calculated by the motion model of the robotic fish. Q , R , and K represent the coefficient matrix. Through the resolution of the optimization problem outlined above, the optimal control sequence can be derived. Subsequently, solely the control sequence up to the next sampling instant is considered, and the optimization process is reiterated in a receding horizon way.

4. Simulation and Analysis

In this section, extensive simulation tests were conducted to validate the effectiveness of the proposed target following method. Firstly, we constructed a simulated pool environment and performed network training using Pytorch 1.12.1. In pursuit of real-time performance, a four-layer fully connected structure was chosen for the architecture of the Actor and Critic networks. The neuron counts in the intermediate two hidden layers are set to 400 and 300, respectively. For key training parameters, the discount factor was set to 0.9, the learning rate to 1×10^{-5} , the target smoothing coefficient to 0.005, the minimum batch size to 256, the maximum training episodes to 3000, and the maximum training steps per

episode to 300. The control period is 0.1 s. The other key parameters of the motion model and control system can be seen in more detail in Table 1.

Table 1. Parameters of the following control system.

Item	Value	Item	Value	Item	Value
m_{11}	9.9 kg	m_{22}	14.5 kg	m_{33}	1.8 kg
d_{11}	17.2 kg/s	d_{22}	19.3 kg/s	d_{33}	1.1 kg·m/s ²
Q	$diag\{50, 50, 0.2\}$	R	$diag\{0.005\}$	K	$diag\{0.5\}$
c_1	0.4	c_2	0.4	c_3	0.2
k_1	10	k_2	20	T	10
$u_{f \min}$	$\tau_u = 0, \tau_r = -6$	$u_{f \max}$	$\tau_u = 8, \tau_r = 6$		

4.1. Training Results and Analysis

In this section, the neural network constructed based on the proposed reinforcement learning method was trained. When the prediction step size was set to five, Figure 3 presented the results of six training sessions, including rewards, Actor loss, and Critic loss. The shaded area in the figure represents the range of the maximum and minimum values for each round in the six training sessions. The obtained results indicate that the proposed method exhibits a relatively rapid convergence rate in the initial stages, achieving preliminary convergence at around 500 steps. By the time the training steps reach 3000, complete convergence is achieved, resulting in satisfactory training outcomes.

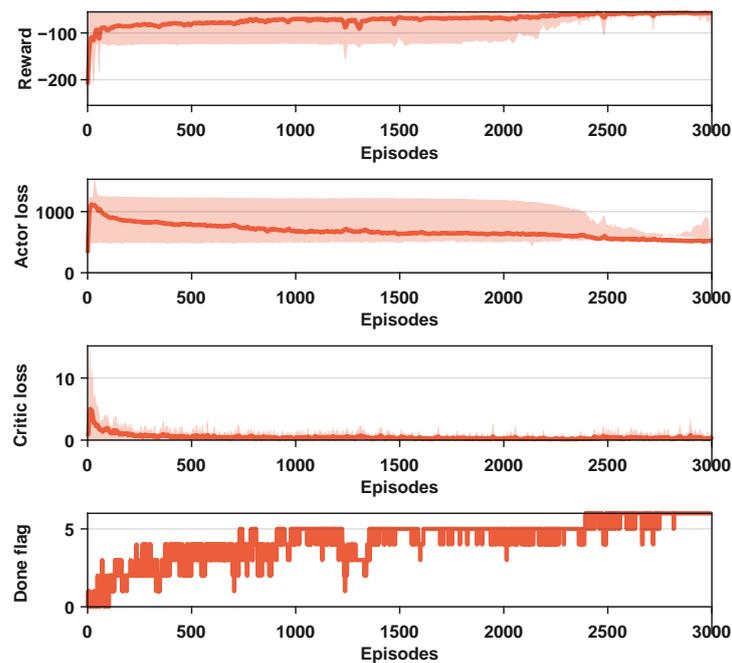


Figure 3. The training results of PDDPG when $N_p = 5$.

It should be noted that in Figure 3, the Done flag is used to indicate whether the target point is reached in each training round. In each training session, if the target point was reached within the current round, the variable was set to 1; otherwise, it was set to 0. The Done flag represents the cumulative sum of these binary values in the six training sessions, with a minimum value of zero and a maximum value of six. Therefore, the results suggest that with an increase in the number of training steps, the Done flag exhibited an overall upward trend, indicating that convergence was ultimately achieved in each training session. As a whole, from the reward and loss results, it can be seen that the values reached a satisfactory level after 3000 iterations, and the curves appeared relatively smooth. In terms

of completion, the Done flag was essentially at the maximum value around 3000 steps, indicating successful attainment of the target point in each training instance.

To investigate the impact of different prediction steps on training outcomes, we conducted relevant simulation experiments. Initially, we set the parameter N_p to values of 0, 3, 5, 8, and 10, employing identical training parameters. The training results are illustrated in Figure 4a. The findings indicate optimal training performance when N_p is set to 5, followed by $N_p = 3$. Notably, with $N_p = 5$, not only did rapid convergence occur, but a certain degree of training stability was also observed. Subsequently, we performed an extension of training for the case where N_p was set to 5, reaching 5000 steps. The results demonstrate that the reward has stabilized without significant fluctuations.

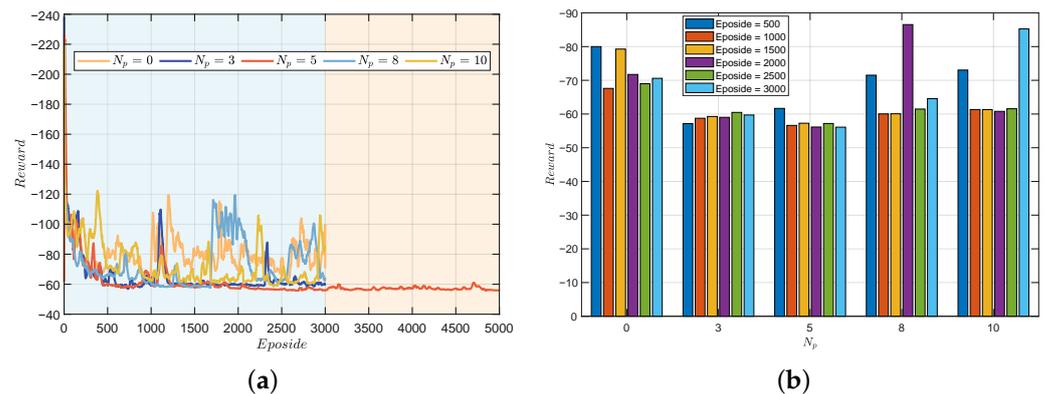


Figure 4. The reward comparison of testing results. (a) Under different N_p . (b) Under different episodes.

Further analysis is presented in Figure 4b, depicting comparative results of training with fixed steps under different N_p . Overall, PDDPG exhibits superior training performance compared to traditional DDPG, with $N_p = 3$ and $N_p = 5$ showcasing particularly outstanding results. However, it is noteworthy that temporary divergence phenomena were observed during the training processes with $N_p = 8$ and $N_p = 10$. This suggests that when the prediction step size is too small or too large, the training outcomes are unsatisfactory. This phenomenon can be attributed to two main factors. Firstly, a too small prediction interval implies an ineffective restriction on overshooting, leading to frequent changes in system attitude and triggering substantial penalties. Secondly, influenced by the inaccuracy of the motion model, longer prediction intervals may render the system more sensitive to model uncertainty, since model predictions over extended periods may accumulate errors. This could result in a decrease in the robustness of control performance, particularly in the presence of uncertainty or environmental changes.

Hence, the selection of an appropriate prediction interval is crucial for enhancing model training effectiveness, as both excessively small and large prediction intervals may impact the stability and performance of training results. The obtained results support the selection of N_p as 5 during the training process, as it exhibits superior performance in terms of rapid convergence and training stability. These findings provide valuable insights into the optimization of prediction step parameters for effective model training.

4.2. Testing Results and Analysis

4.2.1. Fix-Point Target Following under Single Robotic Fish

To further validate the effectiveness of the proposed method, simulation tests were conducted in this section. Firstly, fix-point target following tests under single robotic fish were performed by setting the initial position, initial attitude, and target point. The trained network results were evaluated based on different prediction horizons. Figure 5a illustrates the motion trajectories of the robotic fish under different prediction horizons. Figure 5b depicts the motion trajectories of the robotic fish at different episodes when $N_p = 5$. Motion

data results, including distance to the target point, the yaw angle difference, forward thrust, and yaw moment, are presented in Figure 6.

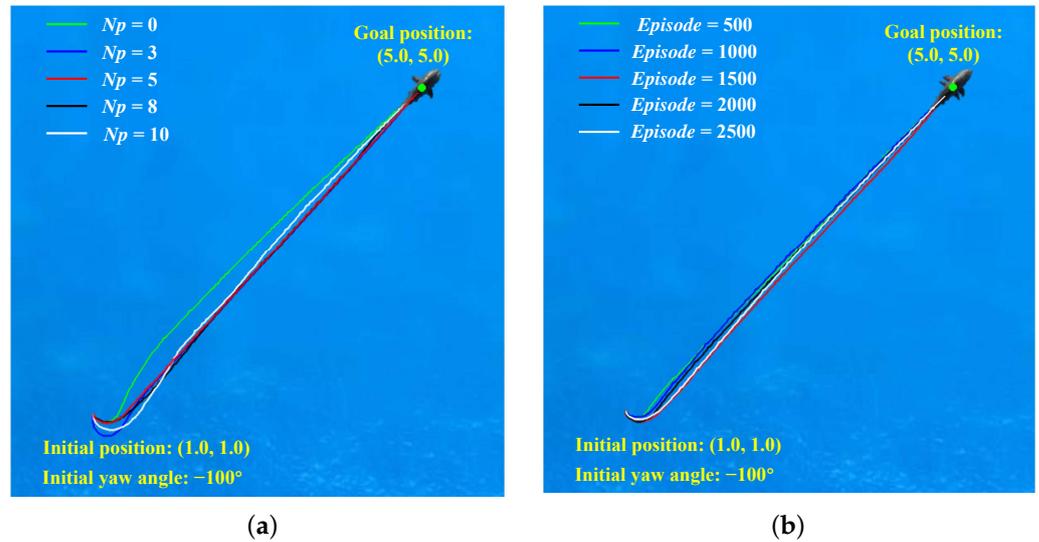


Figure 5. The trajectories comparison of testing results. (a) Under different prediction horizons. (b) Under different episodes.

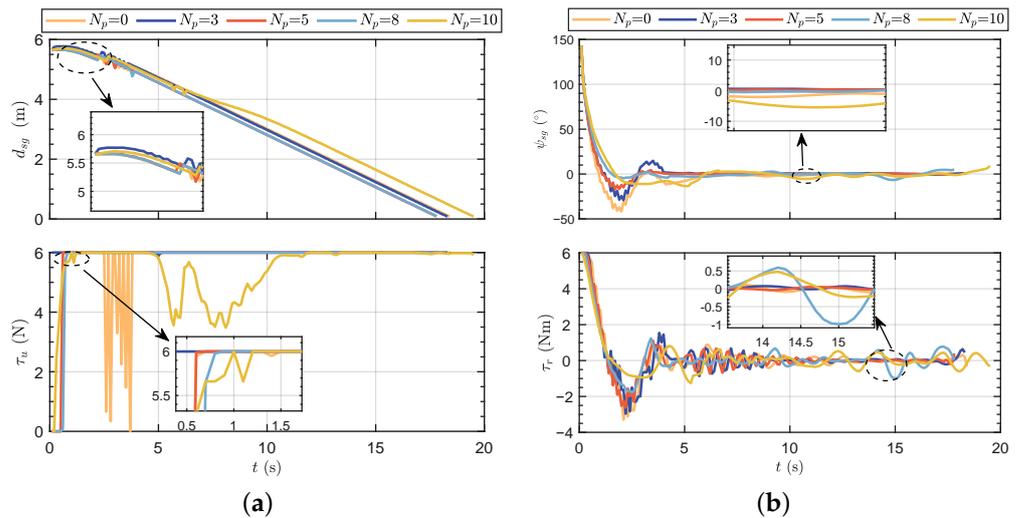


Figure 6. The motion data results of testing results. (a) Distance to the target point and forward thrust. (b) Yaw angle difference and yaw moment.

It can be seen that the trajectories reveal noticeable overshoot phenomenon in the early stages, when the initial pose set at -100° . Without a prediction horizon, the control performance exhibits poor performance. As the prediction horizon increases, the overshoot is effectively suppressed. However, when the prediction horizon reaches 8, some motion instability phenomena begin to emerge. Particularly in the case of $N_p = 10$, the yaw angle curve is unsmooth; the reason for this may be that the increased prediction horizon can lead to a chaotic learning process. Specially, although the performance of $N_p = 3$ is superior to $N_p = 8$ from a training perspective, the motion trajectories suggest that the performance of $N_p = 8$ seems more favorable. The reasons for this phenomenon may be identified in Figure 6b. It can be observed that the test results for $N_p = 8$ show small oscillations in the yaw moment even after entering the steady-state following process, indicating an instability in its swimming posture. The obtained results indicate that there are interactions between the prediction horizon and control performance, and it also influences overshoot suppression and post-steady-state stability in following motion.

4.2.2. Dynamic Target Following under Single Robotic Fish

To assess the effectiveness of the proposed methods in dynamic target following, a standard circle was employed for testing, establishing a foundation for collaborative following. Initially, the circle's center was positioned at (4, 4) with a set radius of 2. The circle was then partitioned into 200 points and followed sequentially. The robotic fish transitioned to the next target point when its distance to the target point fell below 0.3 m.

Figure 7 presents the following path, while Figure 8a,b provide insights into the following data. The results demonstrate the successful implementation of the proposed method for standard circle following, yielding some conclusions. Initially, during the stable following process, the distance between the robotic fish and the dynamic target point is maintained at 0.32 m. According to the switch criterion for dynamic target following, the proposed method can be validated to accomplish following promptly and effectively. Subsequently, the yaw attitude during the following process exhibits relative stability but with subtle oscillations. The forward thrust remains at its maximum value for the majority of the time, which is primarily attributed to the consistent distance between the robotic fish and the dynamic target point. In the context of turning motion, a certain degree of lateral movement is induced, resulting in a lateral velocity. Finally, consistent with the performance of yaw attitude, both the yaw moment and turning angular velocity display slight oscillations characterized by small amplitudes, without significantly affecting the system stability.

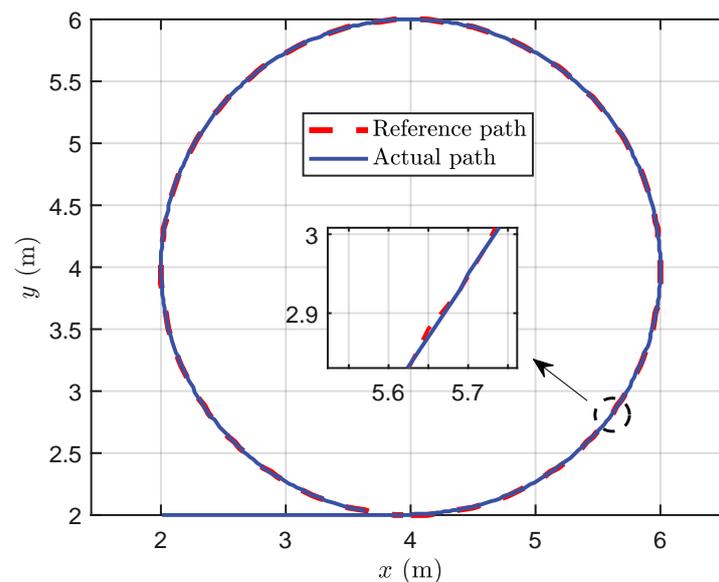


Figure 7. The simulation results of circle following trajectory.

4.2.3. Cooperative Following Control under Multiple Robotic Fish

In this section, inspiration from the efficient mechanisms of fish schooling motion is applied to multi-robot cooperative motion. Studies indicate that maintaining a certain distance and direction between multiple fish swimming together can enhance hydrodynamic performance. For instance, swimming in a side-by-side following manner can reduce energy consumption. Therefore, we emulate the principles of fish swarm motion at the locomotion level, laying the foundation for research on robotic fish swarms. It should be noted that the proposed methods in this paper can be directly applied to any agent in the cluster, and are not limited to the number of agents. To better demonstrate simulation results, this section takes the example of a following task with two biomimetic robotic fish to validate the effectiveness of the proposed methods.

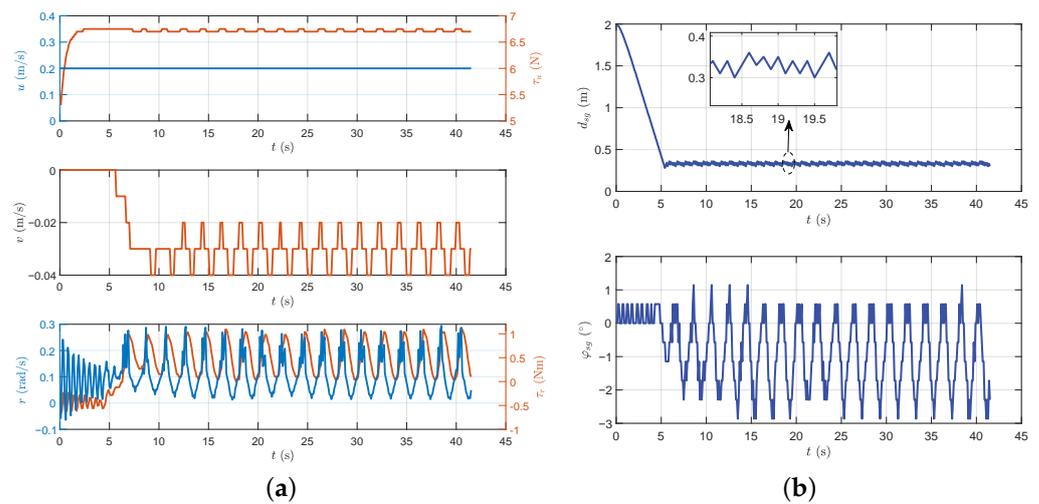


Figure 8. The motion data results of dynamic target following control. (a) The velocity illustration and control force/moment. (b) The following distance and yaw attitude.

Throughout the cooperative following involving multiple robotic fish, the dynamic repositioning of the leader robotic fish prompts a corresponding adjustment in the target-following coordinates P_f^i for the follower robotic fish. Hence, if the initial position of the follower robotic fish remains fixed, d_{max} is a variable which will introduce a notable drawback. In the scenario where the leader robotic fish is engaged in a single-point following task, the ongoing following motion results in a progressive increase in the distance P_f^i from the initial point, leading to an augmentation of the follower robotic fish’s d_{max} , as indicated in Equation (9). Despite the continuous pursuit of the leader by the follower robotic fish, it is indicated that there is potential for a decreasing d_{max} , ultimately causing a gradual reduction in pursuit speed until it becomes insufficient for successful following. To mitigate this challenge, both d_{max} for the leader and follower robotic fish are intentionally maintained as constants, specifically set at 6 during the testing phase in this section.

Furthermore, we outline a task for a biomimetic autonomous system. The task involves a leader robotic fish guiding a collective of follower robotic fish in the exploration of a predefined area. The exploration process is facilitated by establishing search target points for the leader robotic fish, with the other follower robotic fish collaboratively tailing the leader in the exploration endeavor. Besides, the distances and orientations during the following process can be determined based on biological mechanisms observed in natural fish. Consequently, this section presents the simulation testing of leader–follower cooperative following control. First, the corner points of a square are defined as search target points for the leader robotic fish, specifically at (6, 2), (6, 6), (2, 6), and (2, 2). The leader robotic fish aligns its movement towards these target points, switching to the next target point when the distance from the current target falls below 0.3 m. Further, the follower robotic fish dynamically follows the movement of the leader robotic fish in real time.

In alignment with the biological mechanisms of collaborative motion [16], we stipulate a distance of $d_f^i = 0.5$ m and a target orientation of $\varphi_f^i = 135^\circ$ concerning the leader robotic fish. Thus, based on Equation (1), we can derive the real-time target position for the follower robotic fish. Figure 9 provides snapshot sequences of the cooperative following control, encompassing leader trajectory generated by PDDPG, and the follower trajectories generated by PDDPG and NMPC. The obtained results demonstrate that the leader robotic fish successfully completes the standard square path search task with minimal overshooting. Notably, the follower robotic fish, under the control of both PDDPG and NMPC methods, successfully accomplishes the following task.

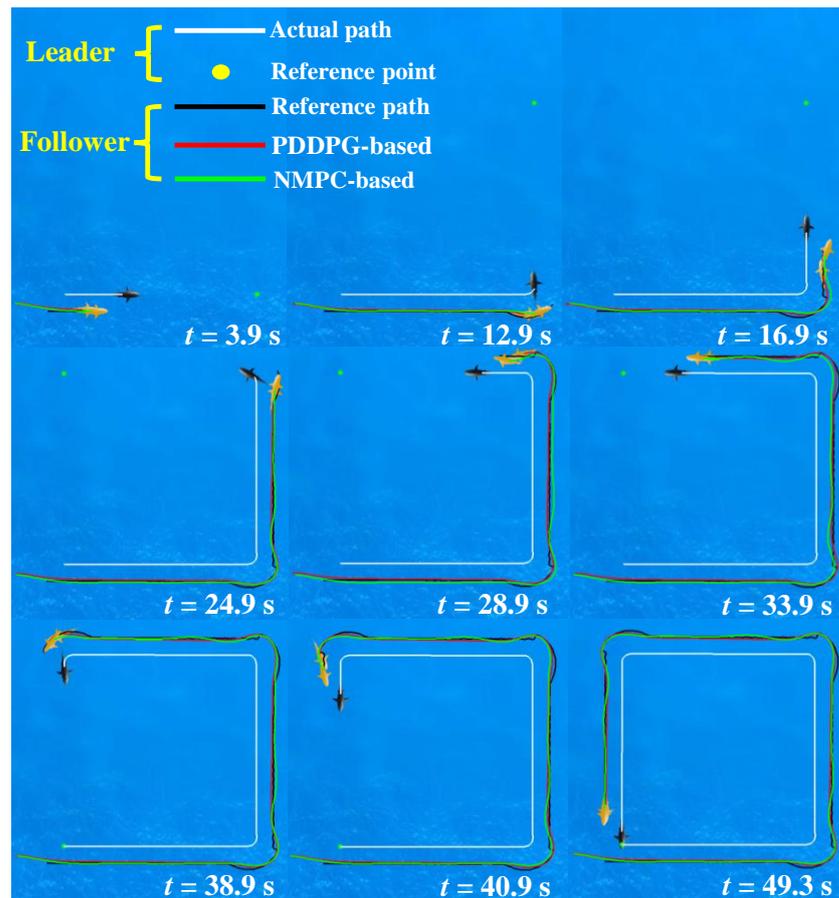


Figure 9. The snapshot sequences of cooperative following control.

Figure 10 depicts the motion data results for cooperative following. Based on d_{sg} , it can be seen that NMPC displays a marginally superior performance in following distance compared to PDDPG, indicating a capacity for faster target following. However, concerning φ_{sg} , PDDPG significantly outperforms NMPC, especially when the leader robotic fish switches target points for a right-angle turn. The distinct weak overshooting characteristic of PDDPG is conspicuously manifested, while NMPC exhibits a noticeable degree of overshooting, resulting in unstable yaw attitude. To further verify the superiority of the proposed method, some quantitative comparison results are offered. On one hand, the total lengths of the following path generated by PDDPG and NMPC are 18.34 m and 18.36 m, respectively, which indicates the slim margin for PDDPG. On the other hand, when the robotic fish turns at a right angle, the overshoot phenomenon is obvious. Taking the time interval of $t = [10 \text{ s}, 20 \text{ s}]$ as an example, the root-mean-square error (RMSE) of φ_{sg} for PDDPG and NMPC are 8.3° and 45.6° , respectively. Additionally, the mean absolute error (MAE) of φ_{sg} for PDDPG and NMPC are 7° and 25.5° , respectively. Therefore, the obtained results illustrate that the proposed PDDPG shows more satisfactory performance.

Moreover, Figure 11 provides insight into the control quantities of forward thrust and yaw moment. In terms of forward thrust, PDDPG seldom reaches its maximum value, whereas NMPC consistently maintains a near-maximum swimming speed throughout the pursuit process. This may be attributed to PDDPG learning, which revealed that that overshooting is prone to occur during high-speed turns, prompting the model to avoid utilizing maximum thrust to prevent this phenomenon. Regarding yaw moments, consistent with yaw attitudes during turning, NMPC outputs a more substantial moment amplitude during turning phases. Nevertheless, it can be observed that in the steady-state tracking phase, NMPC's yaw moment is more stable, while PDDPG exhibits slight oscillations, resembling the swimming behavior of real tail-flapping fish.

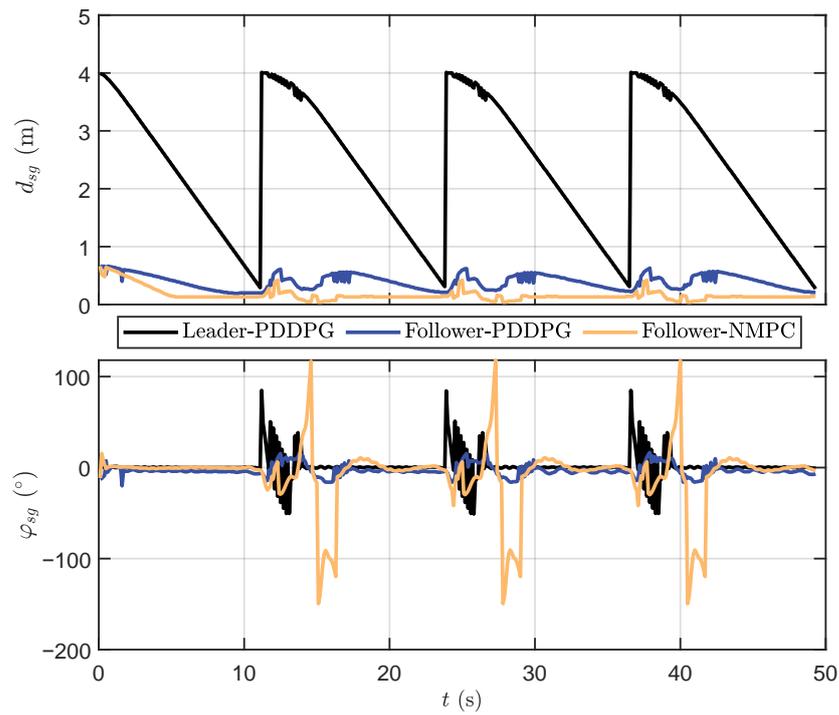


Figure 10. The motion data results of cooperative following distance and yaw difference.

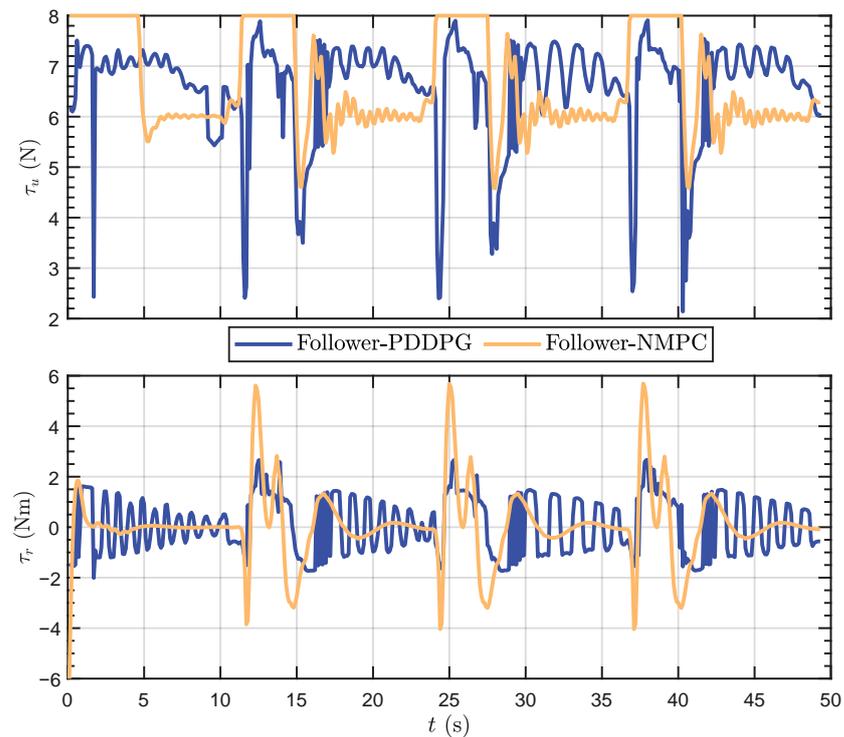


Figure 11. The motion data results of control quantities.

4.3. Discussion

Drawing inspiration from the schooling movement of biological fish, a target-following method based on deep reinforcement learning is proposed, leading to successful implementation of cooperative following control. On one hand, by incorporating predictive thinking into the traditional DDPG algorithm, the system overshooting is effectively reduced. Notably, the method utilizes static target-following scenarios during training but

demonstrates the ability to follow dynamic targets. On the other hand, as an auxiliary control, a target-following controller based on NMPC is designed.

When adjusting the parameters of neural networks and hyperparameters of deep reinforcement learning, we adhere rigorously to the principle of cross-validation to identify the most suitable parameter combinations for a specific task while mitigating issues such as overfitting or underfitting. With the consideration of model complexity, we explore multiple combinations of neurons, conduct cross-validation, and ultimately select appropriate parameters. With regard to the discount factor, learning rate, smoothing coefficient, and batch size, we conduct preliminary tests based on conventional DDPG parameters and further engage in simulation testing to finalize the parameters. Concerning certain parameter settings in the reward function of deep reinforcement learning, we adjust the parameters from the aspects of objectives significance and simulation test results. Regarding generalization performance, we employed several strategies to enhance generalization, including the normalization and scaling technique for the state variables, adding the noise for action generation, and adjusting the target network update frequency. Further, in dynamic target following and cooperative following control simulations, even with real-time changes to the target point, the proposed method consistently demonstrated effective following capabilities, providing further evidence of its robust generalization performance.

Furthermore, extensive simulations are conducted. First, the training results reveal optimal performance with a prediction step of $N_p = 5$. Excessively large or small prediction periods yield unsatisfactory performance. This conclusion is further validated through stationary target-following tests. Second, to assess dynamic target-following performance, the proposed PDDPG algorithm successfully follows a circular trajectory. Finally, by setting up a cooperative following task, the proposed method accomplishes cooperative exploration in a quadrilateral environment, concluding with a performance comparison between PDDPG and NMPC, confirming the effectiveness and superiority of the proposed method.

Despite the successful implementation of cooperative following control, there are still some limitations. On one hand, this paper places particular emphasis on the motion of individual robotic fish, with the goal of directly transferring learned target-following capabilities to swarm control. Consequently, considerations such as inter-swarm motion constraints or obstacle avoidance are omitted. By incorporating neighboring robotic fish states into the state space and devising appropriate reward functions, this issue may be addressed. On the other hand, we focus on the kinematic level of biological fish schooling movement, without delving into behavioral level [30]. To this end, it is required that the joint dynamic models and biomimetic motion control algorithms should be introduced, which is our ongoing endeavor.

5. Conclusions and Future Work

In this paper, we have developed a target-following control framework, including a predictive deep deterministic policy gradient controller and a nonlinear model predictive controller. Inspired by the mechanism of hydrodynamic efficiency improvement observed in fish schooling movement, we aim to investigate a target following method that can be applied to achieve a cooperative following task. In view of the nonlinear characteristics in the motion model of the robotic fish, the predictive modeling concept is incorporated into the conventional DDPG algorithm. On this basis, the training framework is developed, including the normalization of the state space, action space, and the standardization of the reward function. Additionally, we introduce an auxiliary controller based on a nonlinear predictive model, offering an alternative for cooperative following control of the follower robotic fish. Finally, extensive simulations are conducted, demonstrating the effectiveness of the proposed method.

In future work, we plan to further investigate the mechanistic aspects of the behavioral level in the biological fish schooling movement. By incorporating inter-cluster motion constraints, more intelligent cooperative following can be achieved. Furthermore, how

to realize three-dimensional cooperative following control is another issue worthy of in-depth study.

Author Contributions: Conceptualization, Y.W. and J.Y.; methodology, Y.W. and J.W.; software, Y.W. and S.K.; validation, Y.W., J.W. and J.Y.; formal analysis, Y.W. and S.K.; investigation, Y.W. and S.K.; resources, Y.W. and J.Y.; data curation, Y.W. and J.Y.; writing—original draft preparation, Y.W. and J.Y.; writing—review and editing, Y.W., J.W., S.K. and J.Y.; visualization, S.K.; supervision, J.Y.; project administration, Y.W. and J.Y.; funding acquisition, Y.W., J.W. and J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62303260, Grant 62203436, Grant 62233001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data generated during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Katzschmann, R.K.; DelPreto, J.; MacCurdy, R.; Rus, D. Exploration of underwater life with an acoustically controlled soft robotic fish. *Sci. Robot.* **2018**, *3*, eaar3449. [\[CrossRef\]](#)
- Yu, J.; Wang, T.; Chen, D.; Meng, Y. Quantifying the leaping motion using a self-propelled bionic robotic dolphin platform. *Biomimetics* **2023**, *8*, 21. [\[CrossRef\]](#) [\[PubMed\]](#)
- Shao, H.; Dong, B.; Zheng, C.; Li, T.; Zuo, Q.; Xu, Y.; Fang, H.; He, K.; Xie, F. Thrust improvement of a biomimetic robotic fish by using a deformable caudal fin. *Biomimetics* **2022**, *7*, 113. [\[CrossRef\]](#)
- Wang, J.; Wu, Z.; Dong, H.; Tan, M.; Yu, J. Development and control of underwater gliding robots: A review. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 1543–1560. [\[CrossRef\]](#)
- Cao, Q.; Wang, R.; Zhang, T.; Wang, Y.; Wang, S. Hydrodynamic modeling and parameter identification of a bionic underwater vehicle: Robdact. *Cyborg Bion. Syst.* **2022**, *2022*, 9806328. [\[CrossRef\]](#) [\[PubMed\]](#)
- Zhang, C.; Zhang, Y.; Wang, W.; Xi, N.; Liu, L. A manta ray-inspired biosyncretic robot with stable controllability by dynamic electric stimulation. *Cyborg Bion. Syst.* **2022**, *2022*, 9891380. [\[CrossRef\]](#)
- Zhu, J.; White, C.; Wainwright, D.K.; Di Santo, V.; Lauder, G.V.; Bart-Smith, H. Tuna robotics: A high-frequency experimental platform exploring the performance space of swimming fishes. *Sci. Robot.* **2019**, *4*, eaax4615. [\[CrossRef\]](#)
- White, C.; Lauder, G.V.; Bart-Smith, H. Tunabot Flex: A tuna-inspired robot with body flexibility improves high-performance swimming. *Bioinspir. Biomim.* **2020**, *16*, 026019. [\[CrossRef\]](#)
- Yu, J.; Wu, Z.; Su, Z.; Wang, T.; Qi, S. Motion control strategies for a repetitive leaping robotic dolphin. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 913–923. [\[CrossRef\]](#)
- Weih, D. Hydromechanics of fish schooling. *Nature* **1973**, *241*, 290–291. [\[CrossRef\]](#)
- Li, L.; Nagy, M.; Graving, J.M.; Bak-Coleman, J.; Xie, G.; Couzin, I.D. Vortex phase matching as a strategy for schooling in robots and in fish. *Nat. Commun.* **2020**, *11*, 5408. [\[CrossRef\]](#)
- Li, L.; Ravi, S.; Xie, G.; Couzin, I.D. Using a robotic platform to study the influence of relative tailbeat phase on the energetic costs of side-by-side swimming in fish. *Proc. R. Soc. A* **2021**, *477*, 20200810. [\[CrossRef\]](#) [\[PubMed\]](#)
- Marras, S.; Killen, S.S.; Lindström, J.; McKenzie, D.J.; Steffensen, J.F.; Domenici, P. Fish swimming in schools save energy regardless of their spatial position. *Behav. Ecol. Sociobiol.* **2015**, *69*, 219–226. [\[CrossRef\]](#) [\[PubMed\]](#)
- Thandiackal, R.; Lauder, G. In-line swimming dynamics revealed by fish interacting with a robotic mechanism. *eLife* **2023**, *12*, e81392. [\[CrossRef\]](#) [\[PubMed\]](#)
- Li, G.; Kolomenskiy, D.; Liu, H.; Thiria, B.; Godoy-Diana, R. On the interference of vorticity and pressure fields of a minimal fish school. *J. Aero Aqua-Bio* **2019**, *8*, 27–33. [\[CrossRef\]](#)
- Dai, L.; He, G.; Zhang, X.; Zhang, X. Stable formations of self-propelled fishlike swimmers induced by hydrodynamic interactions. *J. R. Soc. Interface* **2018**, *15*, 20180490. [\[CrossRef\]](#) [\[PubMed\]](#)
- Verma, S.; Novati, G.; Koumoutsakos, P. Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 5849–5854. [\[CrossRef\]](#)
- Liu, Y.; Jiang, H. Research development on fish swimming. *Chin. J. Mech. Eng.* **2022**, *35*, 114. [\[CrossRef\]](#)
- Dai, Y.; Yu, S.; Yan, Y.; Yu, X. An EKF-based fast tube MPC scheme for moving target tracking of a redundant underwater vehicle-manipulator system. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 2803–2814. [\[CrossRef\]](#)
- Cui, R.; Yang, C.; Li, Y.; Sharma, S. Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 1019–1029. [\[CrossRef\]](#)

21. He, Z.; Dong, L.; Sun, C.; Wang, J. Asynchronous multithreading reinforcement-learning-based path planning and tracking for unmanned underwater vehicle. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 2757–2769. [[CrossRef](#)]
22. Jiang, P.; Song, S.; Huang, G. Attention-based meta-reinforcement learning for tracking control of AUV with time-varying dynamics. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 6388–6401. [[CrossRef](#)]
23. Zou, Q.; Du, X.; Liu, Y.; Chen, H.; Wang, Y.; Yu, J. Dynamic path planning and motion control of microrobotic swarms for mobile target tracking. *IEEE Trans. Autom. Sci. Eng.* **2023**, *20*, 2454–2468. [[CrossRef](#)]
24. Yan, J.; Li, X.; Yang, X.; Luo, X.; Hua, C.; Guan, X. Integrated localization and tracking for AUV with model uncertainties via scalable sampling-based reinforcement learning approach. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 6952–6967. [[CrossRef](#)]
25. Shi, W.; Song, S.; Wu, C.; Chen, C.P. Multi pseudo Q-learning-based deterministic policy gradient for tracking control of autonomous underwater vehicles. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3534–3546. [[CrossRef](#)] [[PubMed](#)]
26. Gao, S.; Peng, Z.; Liu, L.; Wang, D.; Han, Q.L. Fixed-time resilient edge-triggered estimation and control of surface vehicles for cooperative target tracking under attacks. *IEEE Trans. Intell. Veh.* **2023**, *8*, 547–556. [[CrossRef](#)]
27. Wai, R.J.; Lin, Y.W. Adaptive moving-target tracking control of a vision-based mobile robot via a dynamic petri recurrent fuzzy neural network. *IEEE Trans. Fuzzy Syst.* **2013**, *21*, 688–701. [[CrossRef](#)]
28. Huang, Y.; Zhu, M.; Zheng, Z.; Low, K.H. Homography-based visual servoing for underactuated VTOL UAVs tracking a 6-DOF moving ship. *IEEE Trans. Veh. Technol.* **2022**, *71*, 2385–2398. [[CrossRef](#)]
29. Lin, J.; Wang, Y.; Miao, Z.; Fan, S.; Wang, H. Robust observer-based visual servo control for quadrotors tracking unknown moving targets. *IEEE/ASME Trans. Mechatron.* **2023**, *28*, 1268–1279. [[CrossRef](#)]
30. Godoy-Diana, R.; Vacher, J.; Raspa, V.; Thiria, B. On the fluid dynamical effects of synchronization in side-by-side swimmers. *Biomimetics* **2019**, *4*, 77. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.