drones

MDPI

Article

Intelligent Control Strategy for Robotic Manta via CPG and Deep Reinforcement Learning

Shijie Su [1,*], Yushuo Chen [1], Cunjun Li [2], Kai Ni [1] and Jian Zhang [1,*]

[1] College of Mechanical Engineering, Jiangsu University of Science and Technology, Zhenjiang 212100, China; 221210201104@stu.just.edu.cn (Y.C.); 231210201414@stu.just.edu.cn (K.N.)
[2] Zhoushan Institute of Calibration and Testing for Quality and Technology Supervision, Zhoushan 316021, China; licunjun@sina.com
* Correspondence: sushijie@just.edu.cn (S.S.); zhjian127@just.edu.cn (J.Z.)

Abstract: The robotic manta has attracted significant interest for its exceptional maneuverability, swimming efficiency, and stealthiness. However, achieving efficient autonomous swimming in complex underwater environments presents a significant challenge. To address this issue, this study integrates Deep Deterministic Policy Gradient (DDPG) with Central Pattern Generators (CPGs) and proposes a CPG-based DDPG control strategy. First, we designed a CPG control strategy that can more precisely mimic the swimming behavior of the manta. Then, we implemented the DDPG algorithm as a high-level controller that adaptively modifies the CPG's control parameters based on the real-time state information of the robotic manta. This adjustment allows for the regulation of swimming modes to fulfill specific tasks. The proposed strategy underwent initial training and testing in a simulated environment before deployment on a robotic manta prototype for field trials. Both further simulation and experimental results validate the effectiveness and practicality of the proposed control strategy.

Keywords: swimming mode; deep deterministic policy gradient; swimming task; Markov decision process

check for updates

Citation: Su, S.; Chen, Y.; Li, C.; Ni, K.; Zhang, J. Intelligent Control Strategy for Robotic Manta via CPG and Deep Reinforcement Learning. *Drones* **2024**, *8*, 323. https://doi.org/10.3390/drones8070323

Academic Editor: Abdessattar Abdelkefi

Received: 17 May 2024
Revised: 2 July 2024
Accepted: 11 July 2024
Published: 13 July 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

The marine environment is an essential part of the Earth and contains a wealth of resources. Due to their fish-like mode of propulsion and exceptional environmental compatibility, robotic fish have been used in underwater exploration [1,2], environmental monitoring [3], aquaculture [4], and other domains [5]. The swimming styles of robotic fish can be categorized into two groups based on their propulsion mechanisms: Body and/or Caudal Fin (BCF) propulsion and Median and/or Paired Fin (MPF) propulsion [6]. Robotic fish utilizing BCF propulsion are noted for their high maneuverability, whereas those employing MPF propulsion are recognized for their outstanding stability at low speeds [7]. The application of MPF propulsion, exemplified by the manta for their superior swimming efficiency, agility, and interference resistance, has garnered increasing interest from both industry and academia [8–13].

The mainstream swimming control strategies for robotic fish are the sinusoidal control and the Central Pattern Generator (CPG) control [14]. The sinusoidal control can generate a diverse array of swimming gaits continuously but it cannot smoothly and naturally handle changes in swimming frequency and amplitude. CPG control orchestrates rhythmic movements by producing stable periodic signals, offering significant advantages over sinusoidal control, such as stability, robustness, smooth transition, and adjustability. Therefore, the CPG control strategy has garnered growing interest and has been extensively applied in the motion control of diverse biomimetic robotics. Hao et al. [15] demonstrated the efficacy of the CPG control strategy by applying phase oscillators to a robotic manta. They achieved closed-loop heading control by integrating two yaw modes based on phase and amplitude

differences. Chen et al. [16] successfully realized static and moving obstacle avoidance for bionic fish using the CPG control strategy. Qiu et al. [17] developed an asymmetric CPG and explored a passive stiffness adjustment mechanism to adapt to different swimming states of robotic fish.

The intricate and unpredictable nature of hydrodynamics in robotic fish presents significant challenges in developing precise mathematical models for their swimming control. Researchers have turned to model-free approaches to address this issue and enable autonomous swimming for robotic fishes, including Proportional Integral Derivative (PID), Active Disturbance Rejection Control (ADRC), and fuzzy control. Morgansen et al. [18] developed a robotic fish featuring bi-articulated pectoral fins and designed a depth controller based on the PID control strategy. Wang et al. [19] designed an MPF-type robotic fish, RobCutt, utilizing the ADRC strategy for closed-loop swimming control. Cao et al. [20] developed a CPG algorithm combined with fuzzy control to achieve stable 3D swimming of a robot fish.

With the progression of artificial intelligence, many scholars are investigating how reinforcement learning (RL) algorithms can be utilized to enhance the autonomous movement of robotic fish. RL algorithms can learn optimal strategies directly from interactions with the environment, eliminating the necessity for predefined mathematical representations of system dynamics [21]. This contrasts with traditional control methods that depend on precise models to forecast system behavior. Instead, RL employs a data-driven approach, enabling it to adapt and continuously enhance its performance over time. The model-free characteristic of RL allows for its application to complex dynamic systems where constructing exact models may not be feasible or practical. The advent of RL techniques has catalyzed progress in research and development within the domain of robotic fish [22]. Zhang et al. [23] implemented an RL algorithm as a high-level controller, complemented by a CPG control strategy as the low-level controller, for path-tracking control of a BCF-type robotic fish. However, their approach employs the Actor–Critic (A2C) algorithm, encountering limitations, including low sample efficiency and difficulties in achieving a balance between exploration and exploitation, thus reducing its practical applicability.

Deep Reinforcement Learning (DRL) is a state-of-the-art algorithmic framework that combines the principles of RL with the representational capabilities of deep learning. DRL algorithms surpass traditional RL methods in terms of generalization and learning capabilities, particularly in complex environments and under conditions of high dimensionality. Woo et al. [24] introduced a DRL-based controller for Unmanned Surface Vehicle (USV) path tracking. However, USVs exhibit fewer complex dynamics than the intricate swimming patterns observed in robotic fish. Zhang et al. [25] validated the robustness of the Deep Deterministic Policy Gradient (DDPG) algorithm, and their study showed the applicability of DDPG in robot control.

Although various methods have been successfully applied to numerous robotic systems, the unique morphology and kinematic properties of the robotic manta require a specialized approach. Table 1 delineates prior studies on control strategies for robotic mantas.

Table 1. Research on control strategies for the robotic manta.

Item	Control Strategy	Performance
Biomimetic robot (2016) [26]	CPG/infrared sensor feedback	Simple autonomous obstacle avoidance
Bionic manta ray (2018) [10]	CPG controller	Open-loop swimming mode switching
Manta robot (2022) [15]	CPG/amplitude phase controller	Simple yaw control
Manta robot (2022) [27]	CPG/fuzzy controller	Gliding and slapping propulsion control
Manta robot (2022) [28]	S-plane fuzzy controller	Stable heading switching control
Robotic manta (2024) [29]	Sliding–mode fuzzy controller	Stable path-following control

Zhang et al. [26] implemented a simple autonomous obstacle avoidance control by using infrared sensor feedback information based on the design of a simple CPG network. Zhang et al. [10] designed a CPG controller for open-loop swimming control based on the

development of a robotic manta combining rigid and soft structures. Hao et al. [15] improved the basic swimming performance of the robotic manta by modifying the CPG model, which diversified the swimming modes. However, subsequent parameter adjustments increased the complexity of the closed-loop control strategy, making it more difficult to coordinate various parameters, leading to unstable yaw control. Zhang et al. [27] achieved the switch between the gliding and flapping propulsion modes of the robotic manta by combining the CPG network and the fuzzy controller. However, its control accuracy and propulsion efficiency were low. He et al. [28] achieved a smaller precision heading change control by combining the S-plane controller and the fuzzy controller. Meng et al. [29] designed a sliding-mode fuzzy controller based on the development of a new type of robotic manta, achieving stable path-tracking control of the robotic manta. However, due to the use of a sine curve for basic swimming control, the trajectory was not smooth when the robotic manta switched swimming modes. From these studies, we find that most of the current research on the motion control of robotic mantas is focused on basic swimming control strategies. There are still many shortcomings in simultaneously improving smooth transitions between swimming modes and enhancing adaptability, autonomy, and stability in unknown environments.

This study presents a CPG–DDPG control strategy to improve the smoothness of transition in the robotic manta swimming mode, and to boost its adaptability, autonomy, and stability in unpredictable dynamic environments. The proposed control strategy integrates a CPG as the fundamental mechanism, complemented by the DDPG algorithm functioning as the advanced regulatory strategy.

The main contributions of this paper are as follows:

(1) We have successfully developed a robotic manta and proposed a CPG control strategy that allows for smooth transitions between different swimming modes by adjusting a single parameter, significantly reducing the difficulty of parameter adjustment in closed-loop control strategies and enhancing stability. This CPG control strategy enables us to more accurately simulate the swimming behavior of manta, thus allowing for more precise and natural control of swimming movements.

(2) We have put forward a CPG-based DRL control strategy to adjust the CPG control parameters of the robotic manta via the DDPG algorithm. This strategy utilizes the learning capabilities of the DDPG algorithm and the stability of the CPG model to achieve more flexible and adaptive control. By making decisions based on the current state of the robotic manta, it effectively enhances the adaptability and swimming efficiency of the robotic manta in unknown environments.

(3) We conducted a series of simulations and real-world prototype experiments to validate the effectiveness of the proposed CPG–DDPG control strategy in the swimming control process of the robotic manta.

The remainder of this paper is structured as follows: Section 2 introduces the CPG model for the robotic manta swimming control. Section 3 provides a detailed discussion of the proposed CPG-DDPG control strategy. Section 4 describes the simulation environment and results for the robotic manta swimming task. Section 5 details the fundamental swimming experiments and swimming task experiments conducted with the robotic manta prototype. Finally, Section 6 summarizes the paper and outlines the contributions of the proposed CPG–DDPG control strategy.

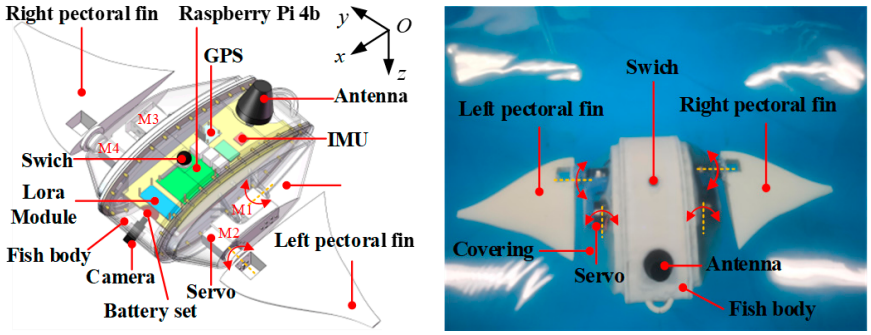
2. CPG Model for Robotic Manta Swimming Control

In this section, we present the design of the robotic manta and propose a CPG model for controlling its swimming.

2.1. Design of the Robotic Manta

Figure 1 presents the structure of the designed robotic manta (length 300 mm, width 690 mm, weight 3.8 kg). This robot comprises a central housing and two pectoral fins, all of which are manufactured via 3D printing techniques utilizing a material with a

density of nearly 1000 kg/m^3. A silicone skin, affixed to a transition plate, encloses the robot to ensure the servomotors are protected from water ingress. This skin is infused with insulating oil to adjust the robotic manta's buoyancy and enhance its resistance to underwater pressures. The design of the robot incorporates a streamlined contour, effectively minimizing fluid resistance.



(**a**) (**b**)

Figure 1. Structure of the robotic manta. (**a**) Conceptual design; (**b**) robotic manta prototype.

The central shell houses the electrical system, and two pectoral fins provide propulsion. A waterproof camera is mounted on the front of the shell, and a Raspberry Pi 4B control board housed in the center of the shell is used to handle various complex tasks. The wireless module is used for command and data transfer with the host computer. The Inertial Measurement Unit (IMU) and GPS provide attitude and position information for the robotic manta. Lithium batteries arranged along the bottom of the shell provide power to the robotic manta and enhance its stability in the water. Pressure sensors mounted at the end of the shell monitor the dive depth of the manta. The pectoral fins are controlled by four bus servos, two on each side, one for flapping motion and the other for rotary motion. The yellow dotted lines denote the axis of pectoral fin rotation, and the red arrows denote the direction of pectoral fin rotation (the same notation is used in subsequent figures). Each servomotor is controlled by a Pulse Width Modulation (PWM) signal generated by the Raspberry Pi 4B control board. The specific technical parameters are listed in Table 2.

Table 2. Main technical specification of the robotic manta.

Items	Characteristics
Dimension	~300 mm × 690 mm × 120 mm
Total mass	~3.8 kg
Control unit	Raspberry Pi 4B (8 GB)
Wireless module	LoRa (SX1268)
Sensors	IMU (JY901B, Witmotion, Shenzhen, China), Pressure sensor (MS5837, Rovmaker, Shanghai, China), GPS (WTGPS-300, Witmotion, Shenzhen, China)
Drive unit	Servomotors×4 (LX-824, Hiwonder, Shenzhen, China)
Power supply	7.4V (18650 Li-ion×8)

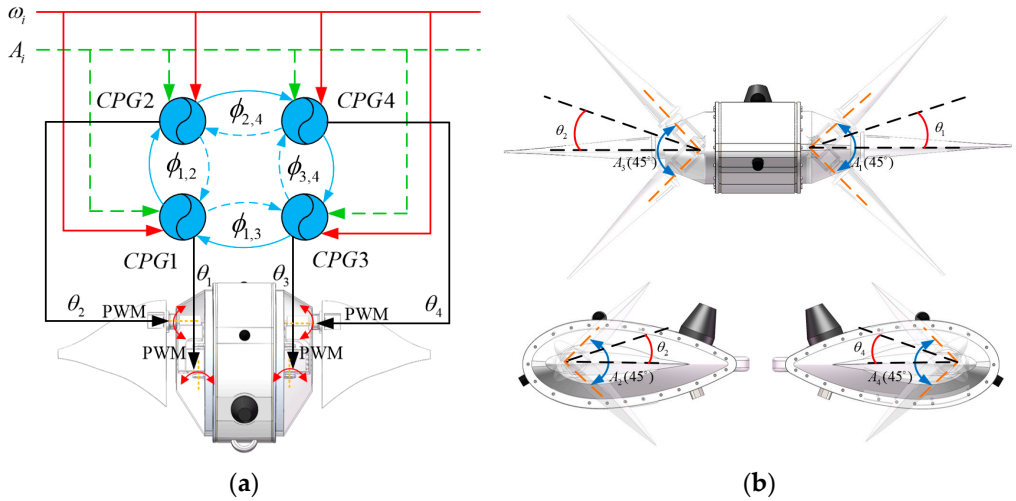
2.2. CPG Control

CPG is a biological neural network, extensively found in vertebrate and invertebrate species, capable of generating rhythmic signals without sensory feedback [30]. Studies have shown that periodic activity within the central nervous system produces movement of the fins and body of the fish. Inspired by these findings, scholars have developed various CPG algorithms to control the swimming patterns of robotic fish [31]. Currently, the most commonly used CPGs include recurrent neural oscillators, phase oscillators [32], and Hopf oscillators [33].

To ensure the generation of stable periodic oscillatory signals, CPG models with low parametric and computational complexity are usually preferred. Following this principle, we employ the Hopf oscillator as the basic unit for constructing the underlying motion control network. In the state space, Hopf oscillators exhibit stable limit cycles, signifying their capability to generate periodic oscillations with consistent waveforms from any non-zero initial conditions. The robotic manta features four articulated joints, each powered by a distinct Hopf oscillator to facilitate its propulsion. For coordinated swimming motion across the joints, it is imperative to establish an interconnection among the CPG units corresponding to each joint. The Hopf oscillator model incorporates a coupling term as described below:

$$\begin{cases} \dot{x}_i = \alpha(A_i{}^2 - r_i{}^2)x_i{}^2 - \omega_i(y_i - b_i) + \sum_{j=1}^{4}(\cos\phi_{i,j}x_j - \sin\phi_{i,j}(y_i - b_i)) \\ \dot{y}_i = \alpha(A_i{}^2 - r_i{}^2)(y_i - b_i)^2 - \omega_i x_i + \sum_{j=1}^{4}(\sin\phi_{i,j}x_j - \cos\phi_{i,j}(y_i - b_i)) \\ \theta_i = y_i \end{cases} \tag{1}$$

where i denotes the number of joints, α is the following velocity, A_i represents the swing amplitude, ω_i denotes the swing frequency, $\phi_{i,j}$ denotes the phase relationship between joints i and j, b_i is the offset angle of joints i, x_i and y_i represent the state variables of the CPG oscillator i, and $r_i = \sqrt{x_i{}^2 + y_i{}^2}$. The output signals θ_i of the CPG model represent the deflection angles of the corresponding joints i (Figure 2b), which are sent in the form of PWMs to the servo motors of each joint to control the flapping of the pectoral fins, thereby controlling the swimming of the robotic manta.



(**a**) (**b**)

Figure 2. CPG model and schematic of the pectoral fin servo angle. (**a**) CPG model; (**b**) schematic of the pectoral fin servo angle.

The orange dashed lines in Figure 2b denote the range of rotation of the pectoral fins, and the blue arrows denote the direction of rotation of the pectoral fins.

Although the CPG model (Figure 2a) can generate rhythmic output signals, the complexity introduced by an excessive number of input parameters significantly hindered the design of control strategies. To address this issue, we opted to fix certain parameters, including α, A, ϕ, and ω, and proposed a modified CPG model. This modified approach can generate a diverse range of output signals utilizing the singular control input β. In order to fix parameters α, A, ϕ, and ω, we adopted a systematic experimental approach. By individually varying the parameters and conducting systematic tests on control strategies, we determined the best parameter fixation scheme. This ensured the robustness and

applicability of the modified CPG model under different environmental conditions. The modified CPG model is described below:

$$\begin{cases} \phi_{i,j}, \beta = n \\ \omega_i, \beta = n \\ \dot{x}_i = \alpha({A_i}^2 - {r_i}^2){x_i}^2 - \omega_i(y_i - b_i) + \sum_{j=1}^{4}(\cos\phi_{i,j}x_j - \sin\phi_{i,j}(y_i - b_i)) \\ \dot{y}_i = \alpha({A_i}^2 - {r_i}^2)(y_i - b_i)^2 - \omega_i x_i + \sum_{j=1}^{4}(\sin\phi_{i,j}x_j - \cos\phi_{i,j}(y_i - b_i)) \\ \theta_i = y_i \end{cases} \tag{2}$$

where n =1, 2, 3, 4, 5, 6 and each value represents a specific swimming behavior, including straight swimming, fast straight swimming, left turning, right turning, floating, and diving. Recognizing the manta's capabilities for on-the-spot steering and rapid movement, we fine-tuned the oscillation frequency ω within the CPG model to reflect these swimming dynamics more accurately. Within the framework of closed-loop swim control, we employed a reduced frequency for turning maneuvers to enhance directional precision and mitigate the risk of overshooting. Conversely, for straight-line swimming, we increased the frequency to expedite the robotic manta's progress toward the designated target following course correction.

3. Design of the CPG–DDPG Control Strategy

Swimming control of robotic mantas is a significant challenge, requiring constant changes in swimming patterns for precise maneuvering. In this paper, by combining the CPG with the DDPG, we introduce a novel CPG–DDPG control strategy. This strategy dynamically adjusts the CPG network's parameters using the DDPG algorithm, facilitating the successful execution of the swimming task.

3.1. Control Problem and MDP Modeling of the Robotic Manta

The swimming control problem of a robotic manta can be regarded as a Markov Decision Process (MDP) [34], defined by a tuple $[s, a, p, r, \gamma]$, where s denotes the state space; a denotes the action space; p denotes the probability distribution of state transitions; r denotes the reward function; γ is defined as the discount factor. The goal is to learn an optimal policy [35] that maximizes the cumulative reward.

This paper models the swimming control problem of the robotic manta as an MDP task. Figure 3 illustrates the swimming task. Imagine a robotic manta is located at a random coordinate point (initial position) at a predetermined depth, which contains 2D coordinates (x_0, y_0) and relative information $(\delta d_0, \delta y_0)$, and targets for swimming to a target position (x_{tar}, y_{tar}). Here, the initial and target positions are randomly generated. Supposing the current position of the robotic manta is (x_t, y_t), the relative position data comprise the relative distance δd_t and the relative position angle δy_t.

To this end, we define the state space:

$$s = [\delta y_t, \delta d_t, \nu_t, \nu y_t] \tag{3}$$

where t is the time tag, δy_t denotes the error in the actual yaw $y_{\theta t}$ of the robotic manta relative to the target yaw $y_{\theta tar}$, δd_t denotes the relative distance, ν_t denotes the velocity of the robotic manta, and νy_t denotes the angular velocity of the robotic manta's yaw.

Define the action space a:

$$a = [\beta] \tag{4}$$

The upper controller monitors the current state of the robotic manta and adjusts the input parameter β of the CPG-based low-level controller to realize the swimming control of the robotic manta, which guides the robotic manta to swim to the target position.

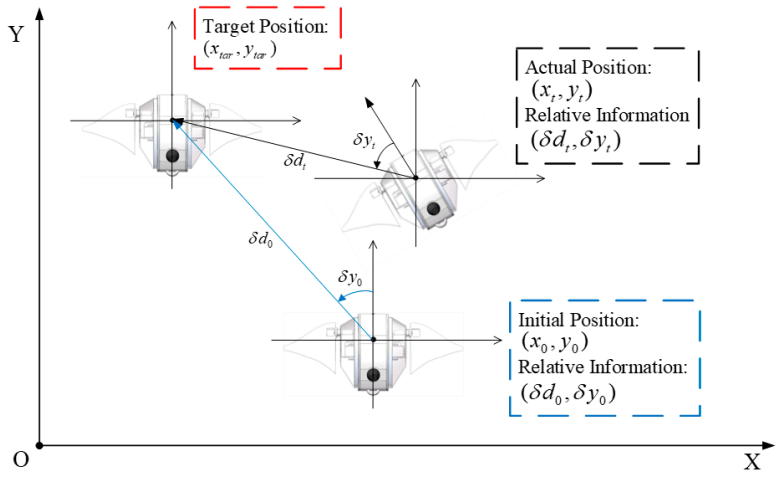


Figure 3. Schematic of position regulation for the robotic manta.

The reward function r is defined as:

$$\begin{cases} r = \rho_1 r_y + \rho_2 r_d + \rho_3 r_{vy} + \rho_4 r_v + \rho_5 r_a + \rho_6 r_b \\ r_y = -\frac{\delta y_t}{\pi} \\ r_d = -\frac{\delta d_t}{D} \\ r_{vy} = \begin{cases} -a, & vy_t \geq b \\ 0, & vy_t < b \end{cases} \\ r_v = \begin{cases} 0, & v_t \geq d \\ -c, & v_t < d \end{cases} \\ r_a = \begin{cases} 0, & d \leq e \\ -f, & d > e \end{cases} \\ r_b = \begin{cases} -g, & |x_i| > x_{\max},\ |y_i| > y_{\max},\ |z_i| > z_{\max} \\ 0, & |x_i| < x_{\max},\ |y_i| < y_{\max},\ |z_i| < z_{\max} \end{cases} \end{cases} \tag{5}$$

where r_y denotes the orientation reward, r_d denotes the position reward, r_{vy} denotes the angular velocity reward, r_v denotes the forward velocity reward, r_a denotes the task completion reward, r_b denotes the device destruction reward, ρ_1, ρ_2, ρ_3, ρ_4, ρ_5, and ρ_6 denote the weight coefficients, and D denotes the initial relative position error. When the set maximum steering speed b m/s is exceeded, $-a$ reward for overspeed is obtained. $-c$ reward is assigned when the robotic manta's forward velocity falls below d m/s. When the robotic manta's position in the world coordinate system falls within a cylindrical area of radius e centered on the target position, the task is considered complete; otherwise, it receives a $-f$ reward. If the robotic manta swims outside the boundaries of the maximum allowed movement area, terminate the current episode, and gain a $-g$ reward. r denotes the overall reward for the MDP task.

3.2. CPG–DDPG Strategy for Robotic Manta Swimming Control

The DDPG algorithm is not only capable of addressing continuous tasks but also discrete tasks, making it extensively utilized in robotics research. It leverages off-policy data and the Bellman equation to learn the Q-function, and it uses this Q-function to learn the policy. Furthermore, the DDPG algorithm utilizes the Actor–Critic (AC) framework, which comprises two distinct networks: the actor network $\mu(s|\theta^\mu)$, which selects actions, and the critic network $Q(s, a|\theta^Q)$, which assesses these actions by estimating their value and is used to update the weights θ^μ of the action network.

The action space of the robotic manta swimming task is discrete. Therefore, this study employs the DDPG algorithm as a high-level strategy and integrates it with the CPG algorithm, thus proposing the CPG–DDPG control strategy. The CPG–DDPG control strategy is illustrated in Figure 4 and Algorithm 1.

Algorithm 1. CPG–DDPG

Randomly initialize the actor network $\mu(s|\theta^{\mu})$ and the critic network $Q(s,a|\theta^{Q})$ with weights θ^{μ} and θ^{Q}
Initialize the target network μ' and Q' with weights $\theta^{\mu'} \leftarrow \theta^{\mu}$, $\theta^{Q'} \leftarrow \theta^{Q}$
Initialize replay buffer R
for each episode **do**
  Initialize a random process N for action exploration
Receive initial observation state S_0
  for each training step **do**
    Select action $a_t = \mu(s_t|\theta^{\mu}) + N_t$, according to the current policy and exploration noise
Take action a_t as β
Execute β to CPG controller
for $i, j = 1$ to 4 **do**

$$\begin{cases} \phi_{i,j}, \beta = n \\ \omega_i, \beta = n \\ \dot{x}_i = f_i(\phi_{i,j}, \omega, A_i, x_i, y_i, \alpha) \\ \dot{y}_i = f_i(\phi_{i,j}, \omega, A_i, x_i, y_i, \alpha) \\ \theta_i = y_i \end{cases}$$

Solve for the θ_i
end for
Servo rotation θ_i
Robotic manta swimming
Observe reward ^{r_t} and observe new state s_{t+1}
Store $transition(s_t, a_t, r_t, s_{t+1})$ in R
Sample a random minibatch N from R
    Update the critic network through minimizing the loss $L_C(\theta^{Q})$
Update the actor policy: $\nabla_{\theta^{\mu}} L_A(\theta^{\mu})$
Update the target networks: $\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$, $\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$
  end for
end for

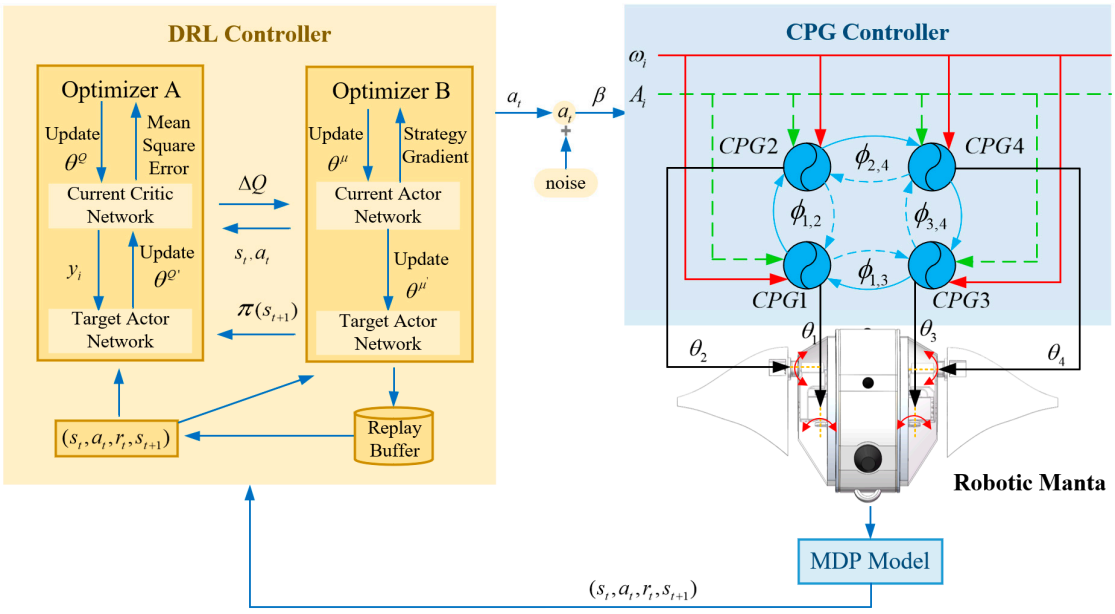


Figure 4. General framework diagram of the CPG–DDPG control strategy.

The DDPG algorithm serves as the decision-making core for the robotic manta, overseeing its current motion state. It processes these parameters and generates control signals β for the CPG-based motion controller. Subsequently, the CPG controller adjusts the phase differences $\phi_{i,j}$ and frequencies ω of its oscillators and outputs the corresponding rotation

angles θ_i to the servo motors. This coordination enables the robotic manta to modulate its swimming direction and speed, facilitating navigation towards designated target points.

4. Simulation

To verify the effectiveness of the proposed control strategy, this section centers on the swimming task of a robotic manta like a case study. An agent was trained in a simulation environment, and the control performance was evaluated post-training.

4.1. Simulation Setup

As a typical model-free closed-loop control method, the PID controller is widely used in modern control systems, while the sinusoidal motion controller is often used as a typical basic locomotion controller for robotic mantas. Therefore, in order to comprehensively evaluate the performance of the CPG–DDPG control strategy proposed in this paper in swimming control, we combine CPG with PID controllers to form the CPG–PID control strategy, combine the SIN controller with the DDPG algorithm to form a SIN–DDPG control strategy, and compare the control performance of the two strategies with the CPG–DDPG control strategy through swimming task testing in a simulation environment.

The main framework of the CPG–PID control strategy is shown in Figure 5. The closed-loop input signal is the target position point, and the feedback signal consists of the actual position (x_t, y_t) and actual yaw angle $y_{\theta t}$. The deviation signal δy_t is used as the input for the PID controller. By applying the PID controller, the input parameters β of the CPG model are obtained, as expressed below:

$$\beta = K_p \times \delta y_t + K_i \times \int_0^t \delta y_t dt + K_d \times \frac{d\delta y_t}{dt} \tag{6}$$

where K_p, K_i, and K_d are the proportional, integral, and derivative coefficients of the PID controller, respectively. We conducted tests on the swimming performance of the robot under different PID parameters and optimized the control parameters. Finally, we set K_p, K_i, and K_d as fixed values. Consequently, the CPG outputs the steering angle θ_i for the servo based on the input parameters β, thereby adjusting the heading of the robotic manta.

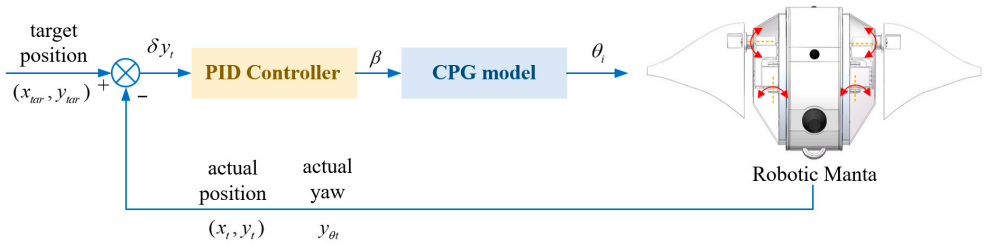


Figure 5. Main framework of the CPG–PID control strategy.

When designing the SIN–DDPG control strategy, we introduce the velocity control parameter k_v and the direction control parameter k_t in the SIN controller, which are used to adjust the swimming speed and direction of the robotic manta, respectively. The upper-level controllers of SIN–DDPG and CPG–DDPG have the same state space and reward function. The difference lies in the DDPG algorithm in SIN–DDPG, which makes decisions after observing the current state of the robotic manta and outputs two control parameters, k_v and k_t, to the SIN controller. These parameters adjust the amplitude to change the thrust generated by the pectoral fins, thereby adjusting the swimming direction and speed of the robotic manta. The SIN controller outputs the rotation angle of the servo. The specific calculation formula of the sine motion controller is as follows:

$$\begin{cases} \alpha_l = \alpha_0 \cdot \sin(\omega t) \\ \theta_l = k_{tl} \cdot k_v \cdot \theta_0 \cdot \sin(\omega t + \phi) \\ \alpha_r = \alpha_0 \cdot \sin(\omega t) \\ \theta_r = k_{tr} \cdot k_v \cdot \theta_0 \cdot \sin(\omega t + \phi) \end{cases} \tag{7}$$

$$k_{tl} = \begin{cases} 1 + k_t, \ k_t < 0 \\ 1, \ k_t > 0 \end{cases} \tag{8}$$

$$k_{tr} = \begin{cases} 1 + k_t, \ k_t > 0 \\ 1, \ k_t < 0 \end{cases} \tag{9}$$

where α_l represents the tracking curve of the left flutter servo, θ_l represents the tracking curve of the left rotary servo, α_r represents the tracking curve of the right flutter servo, and θ_r represents the tracking curve of the right rotary servo. k_v denotes the velocity coefficient, k_{tl} denotes the left direction coefficient, and k_{tr} denotes the right direction coefficient.

The construction and training of the SIN–DDPG and CPG–DDPG control strategies relies on PyTorch, utilizing Python as the programming language. The simulation process unfolds as follows: Initially, the neural networks are established in PyTorch. Subsequently, the neural networks' weights are continuously updated by the DDPG algorithms using the amassed training data. Once these algorithms converge, the necessary control strategies can be derived, and the weights of the neural network represent the training outcomes. The training process for this network consists of 1000 episodes, with each episode consisting of 500 training steps. The replay buffer size is set to 10,000, and the batch size is set to 62. The learning rates for both the actor and critic networks are fixed at 0.0001, with the discount factor $\gamma = 0.99$. Random noises are added in the training phase and used to enhance the generalization ability of the model. Table 3 presents the hyperparameters [36] used in the proposed SIN–DDPG and CPG–DDPG control strategies.

Table 3. Training hyperparameters' settings.

Parameter	Value
Learning rate	0.0001
Discount rate	0.99
Size of the replay buffer	10,000
Batch size	62
Episode	1000
Training step	500

The simulation platforms for SIN–DDPG, CPG–DDPG, and CPG–PID control strategies all use Webots and are integrated with PyCharm to carry out collaborative simulation activities. Figure 6 depicts the simulation environment of the robotic manta. The main parameters of the Webots simulation environment are shown in Table 4.

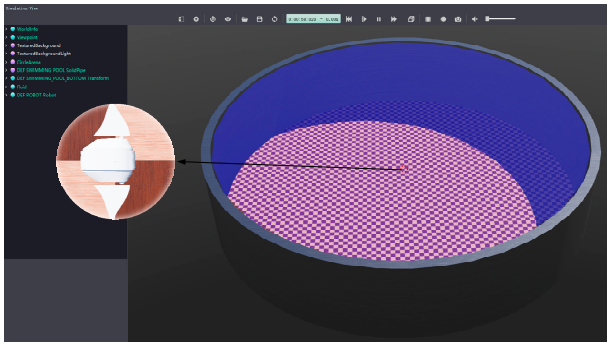


Figure 6. Robotic manta simulation environment.

Table 4. Simulation parameters.

Parameter	Value
Pool radius	50 m
Pool depth	20 m
Fluid depth	20 m
Fluid density	1000 kg/m^3
Drag force coefficients	10
Robotic manta length	0.3 m
Robotic manta width	0.69 m
Robotic manta density	1000 kg/m^3
Each servo rotation range	$(-45°, 45°)$

This study compared the performance of SIN–DDPG, CPG–DDPG, and CPG–PID control strategies in simulated swimming tasks. The evaluation focused on task success rate, path length during swimming, task duration, and trajectory smoothness.

4.2. Analysis of Simulation Results

Figure 7 presents the convergence curves of the SIN–DDPG and CPG–DDPG control strategies under simulation conditions. The learning outcomes are derived from multiple training, utilizing mean and standard deviation for analysis. The results indicate that the CPG–DDPG control strategy surpasses the SIN–DDPG control strategy in terms of convergence speed and average reward value.

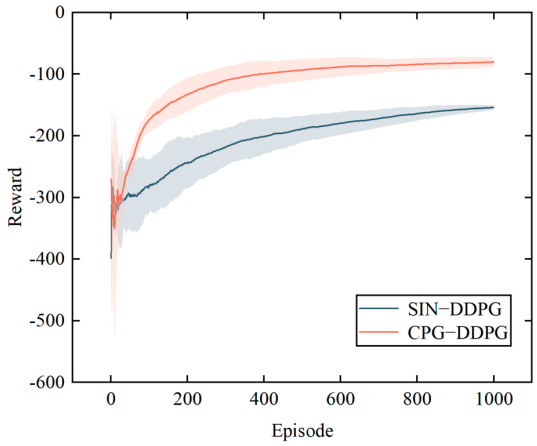


Figure 7. Comparison of convergence curves of CPG–DDPG and SIN–DDPG control strategies.

After training, we tested the SIN–DDPG, CPG–DDPG, and CPG–PID control strategies in a simulated environment for 100 swimming tasks, respectively. The initial coordinates were set to (4.6, 0.4) within a global coordinate system, with the destination point coordinates being randomly selected. A task was considered successful if the robotic manta reached the specified destination within 500 steps; failure was considered otherwise. Figure 8 presents a comparison of the success rates and average rewards.

Through 100 tests, the CPG–PID control strategy successfully reached the target point 89 times, achieving a success rate of 89%. The SIN–DDPG control strategy had a success rate of 94%, with an average reward of −40. By contrast, the CPG–DDPG control strategy was the only control strategy to achieve a 100% success rate in the swimming tasks, and its average reward of −26.4 was significantly higher than that of the SIN–DDPG control strategy.

In order to further evaluate the effectiveness of the control strategies, we selected path length and time consumption as indicators. The three mentioned control strategies were employed to conduct a series of ten simulated swimming control tasks, each targeting five distinct points.

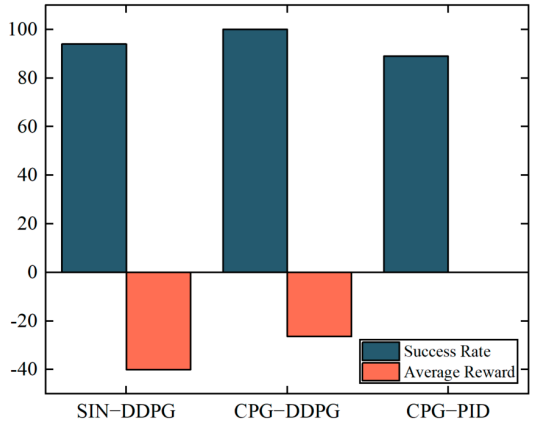


Figure 8. Comparison of success rate and average reward.

The experimental results are presented in Table 5, where the abbreviations S–D, C–D, and C–P denote SIN–DDPG, CPG–DDPG, and CPG–PID control strategies, respectively. Analysis of the data reveals that the average path lengths for the SIN–DDPG, CPG–DDPG, and CPG–PID strategies were 16.72 m, 7.94 m, and 15.28 m, respectively. The corresponding average durations to complete the path were 166.2 s for SIN–DDPG, 37.4 s for CPG–DDPG, and 93.6 s for CPG–PID. These durations translated to average swimming speeds of 0.10 m/s for SIN–DDPG, 0.21 m/s for CPG–DDPG, and 0.17 m/s for CPG–PID. Across all five trials, each of the three control strategies successfully executed the assigned swimming control tasks.

Table 5. Experimental data for swimming tasks using the SIN–DDPG, CPG–DDPG, and CPG–PID control strategies.

Project	Experiment 1			Experiment 2			Experiment 3			Experiment 4			Experiment 5		
control strategy	S–D	C–D	C–P	S–D	C–D	C–P	S–D	C–D	C–P	S–D	C–D	C–P	S–D	C–D	C–P
target point		(5, 5)			(−4, 5)			(4, −5)			(−3, 3)			(1, 5)	
distance (m)		5			10.3			5.3			8.5			6.6	
traveling path (m)	12.8	5.7	5.6	22.8	11.3	19.6	20	6.8	7	18	9.5	24.7	10	6.4	19.5
time taken (s)	114	29	27	227	51	106	159	34	37	219	44	176	112	29	122
average speed (m/s)	0.11	0.2	0.21	0.10	0.22	0.18	0.13	0.2	0.19	0.08	0.22	0.14	0.09	0.22	0.16

Figure 9a shows the comparison of the average path lengths of the three control strategies in the five experiments. The results indicate that the average path length of the CPG–DDPG control strategy was always less than half of the SIN–DDPG control strategy. In Experiment 1, the average path length of the CPG–DDPG control strategy slightly exceeded that of the CPG–PID control strategy. However, in the subsequent four experiments, the CPG–DDPG control strategy maintained a shorter path length. In addition, the standard deviation of the average path length for the CPG–DDPG control strategy was significantly shorter than that of the SIN–DDPG and CPG–PID control strategies.

Figure 9b compares the average time consumption of the three control strategies. Compared with the SIN–DDPG control strategy, the CPG–DDPG control strategy significantly reduced the average time consumption. Moreover, only in the first and third experiments did the average time consumption of the CPG–DDPG control strategy approach that of the CPG–PID control strategy. In the remaining three experiments, the average time consumption of the CPG–DDPG control strategy was significantly lower than the CPG–PID control strategy. Similar to the results of the average path length, the standard deviation of the average time consumption of the CPG–DDPG control strategy was still significantly lower than the other two control strategies.

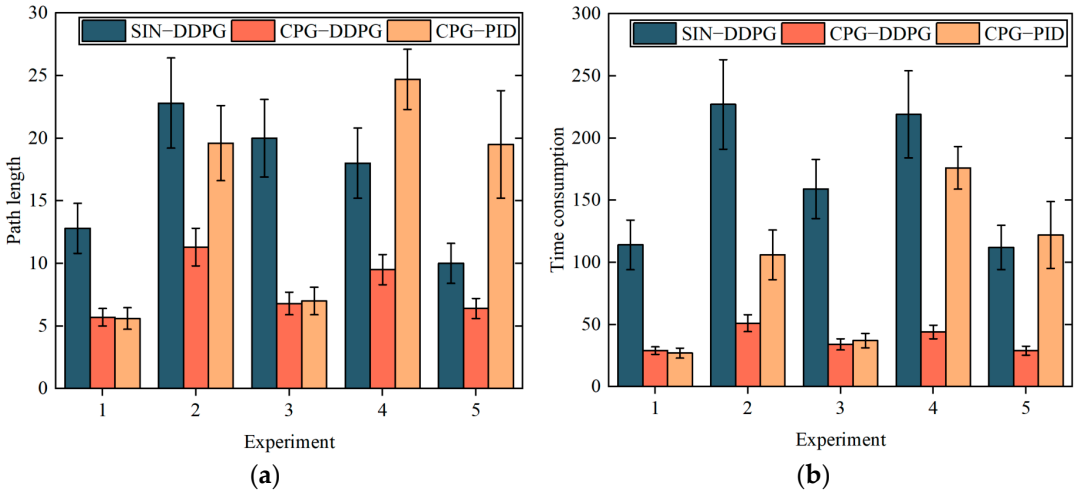


Figure 9. Comparison of control effect of three control strategies. (**a**) Comparison of path lengths; (**b**) comparison of time consumption.

In Experiments 2 and 5, a swimming trajectory of the robotic manta was randomly selected, as shown in Figure 10a,b, where the starting point is indicated by "●" and the end point is denoted by "★", and the task was completed when the robotic manta swam inside the red dashed circles. Figure 10c,d correspond to the distances between the center of the robotic manta and the target point during these two experiments, respectively. Compared with the other two control strategies, the CPG–DDPG control strategy achieved shorter and smoother trajectories due to its ability to quickly and smoothly adjust the swimming pattern of the robotic manta. It quickly corrected for heading angle errors, aligned itself with the target, and then moved quickly toward the target.

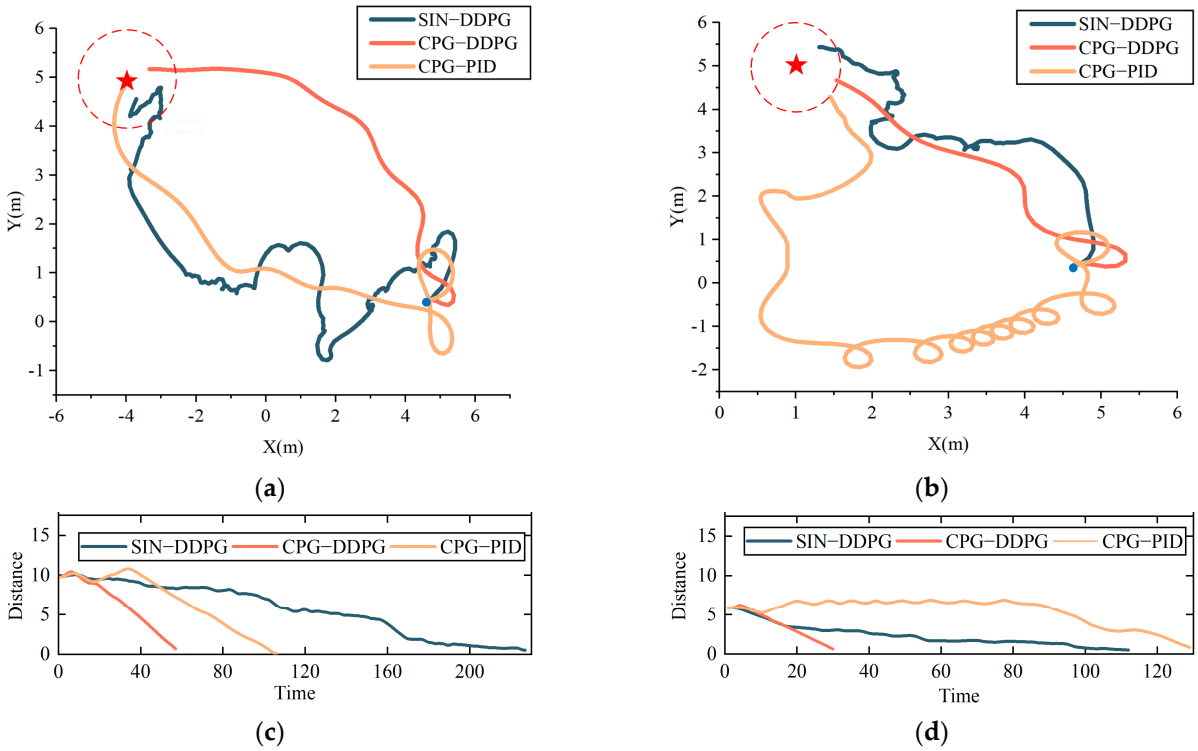


Figure 10. Comparison of swimming paths and distances to the target point under three control strategies. (**a**) Swimming paths for Experiment 2; (**b**) swimming paths for Experiment 5; (**c**) distance to the target point in Experiment 2; (**d**) distance to the target point in Experiment 5.

In contrast, the SIN–DDPG control strategy resulted in a longer path and a less smooth trajectory. This was due to the tendency for discontinuous acceleration and deceleration when adjusting the direction using the SIN control strategy as the base swimming control strategy. Moreover, the SIN control strategy's high sensitivity to the coordination of two control parameters for modulating the speed and direction of the robotic manta led to less stability compared with the CPG control strategy, which smoothly transitioned between swimming modes with just a single input parameter. Consequently, the SIN–DDPG control strategy exhibited increased time consumption for the swimming task.

While the trajectory of the CPG–PID control strategy was smooth, the high sensitivity of the PID control strategy to parameter variations resulted in lower accuracy when adjusting the direction and speed of the robotic manta. This inaccuracy caused the robotic manta to swim in circles, thereby increasing the path length and time expenditure of the swimming task.

In conclusion, the CPG–DDPG control strategy outperformed the SIN–DDPG and CPG–PID control strategies in metrics such as success rate, path length, trajectory smoothness, and time consumption.

5. Experiments and Analysis

In this section, we present a series of experiments performed on the robotic manta prototype. First, we conducted open–loop experiments to evaluate the swimming ability of the prototype. Subsequently, to verify the proposed CPG–DDPG control strategy's performance even more, we deployed the control strategy on a real–world experimental platform and compared it with the CPG–PID control strategy.

5.1. Experimental Platform

In this study, we developed an experimental platform, as shown in Figure 11. The host computer sent task commands and other relevant data to the control center of the robotic manta via a wireless communication module. After receiving these messages, the onboard control unit adjusted the swimming modes of the robotic manta through the control strategy to complete the swimming task.

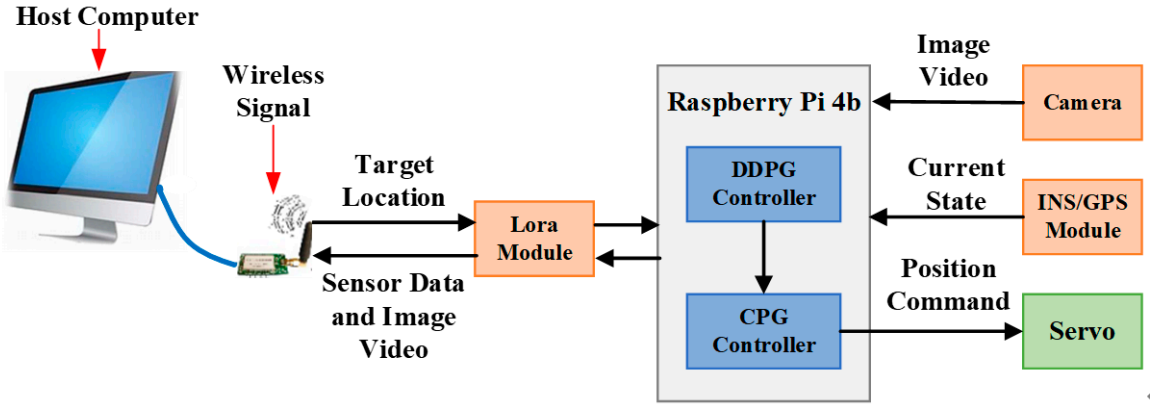


Figure 11. Experimental platform of the robotic manta.

5.2. Basic Performance Test of the Robotic Manta Prototype

The low-level CPG-based swimming control strategy was validated by sending commands for different swimming modes from the host computer and observing the response of the robotic manta. As shown in Figure 12, a straight-line swimming experiment was performed in a pool with a diameter of 1.8 m. The prototype took about 4 s to swim from one end of the pool to the other, with an average speed of 0.4 m/s.

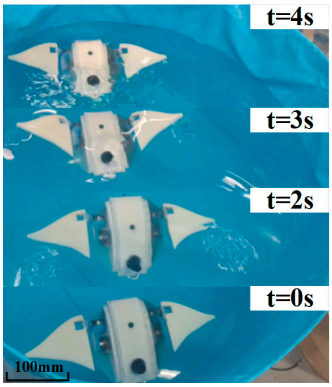
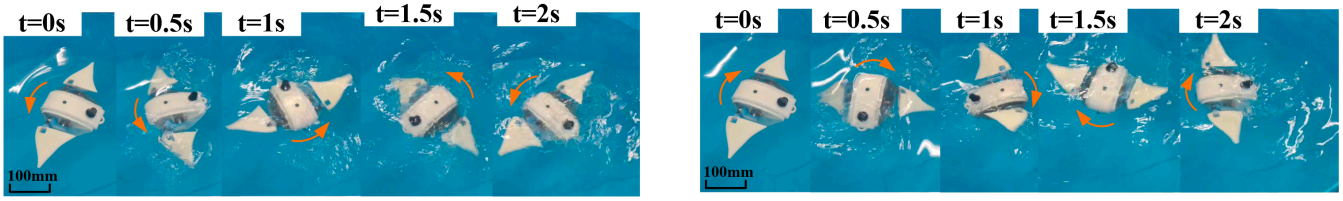


Figure 12. Straight-line swimming experiment.

Figure 13 depicts the experimental results of the left and right turns of the robotic manta prototype. It completed the in- situ turns with a maximum turn speed of 2 s/turn, showing excellent maneuverability.



(**a**) (**b**)

Figure 13. Turning experiment results. (**a**) Experimental effect of left turn; (**b**) experimental effect of right turn.

The experimental results of surfacing and diving are presented in Figure 14. The depth of the pool was about 0.6 m, the whole process took 2 s, and the ascent and descent speed of the robotic manta was about 0.3 m/s.



(**a**) (**b**)

Figure 14. Surfacing and diving experiment results. (**a**) Surfacing experiment; (**b**) diving experiment.

The experiments validated the feasibility of the CPG model on the robotic manta prototype, demonstrating its effectiveness across various swimming modes.

5.3. Swimming Task Experiments of the Robotic Manta Prototype

In simulation tests, we observed that the CPG–PID control strategy outperformed SIN–DDPG, despite having a slightly lower success rate. However, it showed improvements in trajectory smoothness, swimming path length, and task completion time. Considering overall performance and stability, we chose to compare the CPG–DDPG control strategy with the superior-performing CPG–PID control strategy. By controlling the robotic manta prototype to complete swimming tasks, we further assessed the performance of the two control strategies in real-world scenarios. The goal of this task was for the robotic manta to swim from the origin (0, 0) to the target position at coordinates (12, −1), covering a total distance of 12.05 m. Each control strategy was tested ten times. We used average path length

and average time consumption as metrics to evaluate the experimental results. Similar to the simulation results, the CPG–DDPG control strategy resulted in shorter swimming paths and less time consumption compared with the CPG–PID control strategy. Specifically, the average swimming paths for CPG–DDPG and CPG–PID control strategies were 17.2 m and 21.3 m, respectively, with average time consumptions of 92 s and 138 s, and average swimming speeds of 0.19 m/s and 0.16 m/s.

A randomly selected swimming trajectory of the robotic manta is shown in Figure 15a,b, where the center point of the robotic manta is indicated by a red "•", the starting point is the origin, and the endpoint is marked with a "★". Figure 15c corresponds to the distances between the center of the robotic manta and the target point.

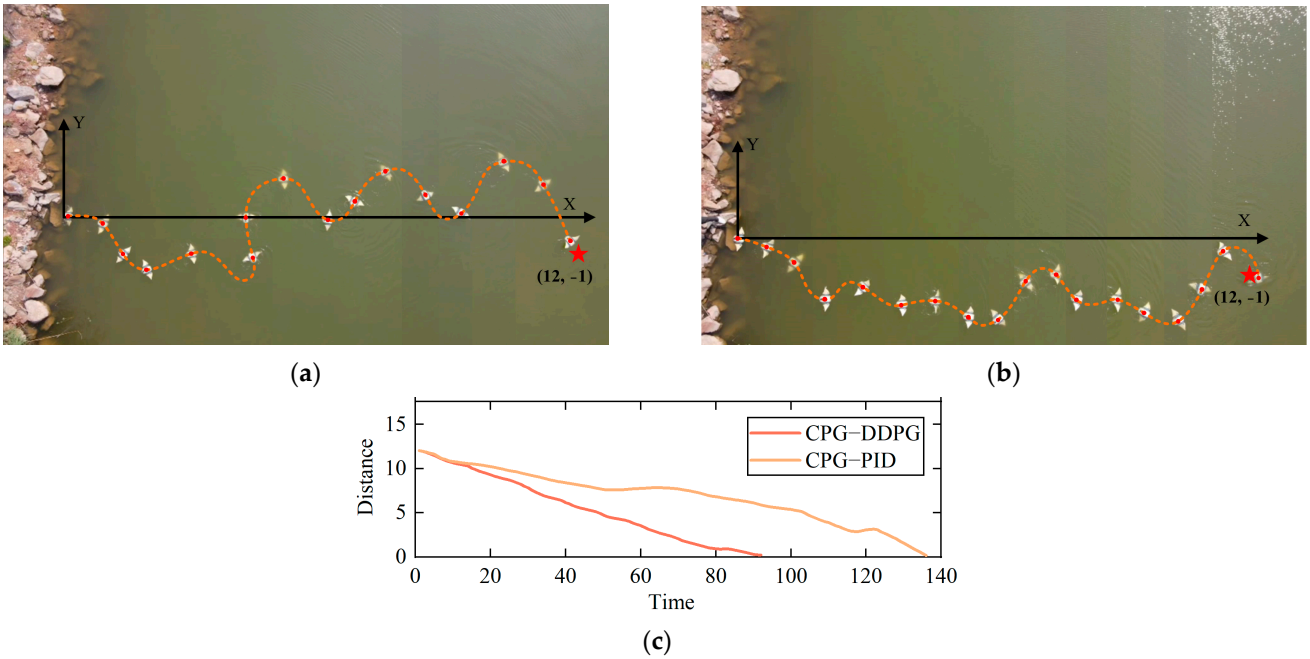


Figure 15. Comparison of the swimming paths and distances to the target point under the two control strategies. (**a**) Path with CPG–PID control strategy; (**b**) path with CPG–DDPG control strategy; (**c**) distance to the target point for the robotic manta.

The experimental results indicate that, in the same physical world environment, the travel path of the robotic manta controlled by the CPG–DDPG control strategy was significantly shorter than that controlled by the CPG–PID control strategy.

Comparing Figures 10 and 15 reveals the impact of simulated and real-world environments on the control performance of the robotic manta. Figure 10 shows the trajectories of three control strategies in a simulated environment with minimal external interference, allowing for a clear evaluation of each strategy. In contrast, Figure 15 shows the trajectories of the CPG–DDPG and CPG–PID control strategies in a real-world environment, including factors such as water flow and sensor noise. The CPG–DDPG control strategy exhibited smoother and more stable trajectories in both environments, highlighting its superior adaptability and robustness. It achieved higher precision with minimal deviation in the simulated environment and effectively resisted disturbances in the real-world environment, maintaining small deviations. Conversely, the CPG–PID control strategy was more sensitive to real-world disturbances, resulting in poorer stability and accuracy.

6. Conclusions

This paper introduces a novel control strategy for robotic manta, termed the CPG–DDPG, which integrates the DDPG algorithm with a CPG network. The CPG–DDPG control strategy consists of a low-level CPG network and a high-level DDPG algorithm,

which adaptively adjusts the control parameters of the CPG network according to the actual attitude and speed of the robotic manta, and then adjusts the swimming pattern to realize the desired swimming control objectives finally. This control strategy offers stable, adaptive, and smooth modality transitions in response to environmental changes. It can be applied to multi-joint robotic fish of different types and degrees of freedom, requiring only some modifications in the CPG model section.

Simulation experiments were conducted on the swimming tasks of the robotic manta using three distinct control strategies: SIN–DDPG, CPG–DDPG, and CPG–PID. The results demonstrated that the proposed CPG–DDPG control strategy outperformed both SIN–DDPG and CPG–PID in success rate and terms of efficiency. Specifically, compared with SIN–DDPG and CPG–PID, CPG–DDPG accomplished the swimming task with 6% and 11% higher success rates and 77% and 62% higher efficiency, respectively. Additionally, the swimming trajectories under the CPG–DDPG control strategy were notably smoother than those of SIN–DDPG and CPG–PID. The results of further experiments on a robotic manta swimming task using CPG–DDPG and CPG–PID control strategies in a real-world environment show that the CPG–DDPG control strategy reduced the task consumption time by 33% compared with the CPG–PID control strategy.

Both simulation and experimental validations confirmed the superior performance of the CPG–DDPG control strategy. Compared with SIN–DDPG, CPG–DDPG more accurately simulated the swimming behavior of manta rays and provided smoother transitions between the various swimming modes. Furthermore, unlike CPG–PID, the CPG–DDPG control strategy eliminated the need for tedious parameter adjustments, and its exceptional learning ability allowed the robotic manta to navigate proficiently in unknown underwater environments. The CPG–DDPG control strategy can quickly and smoothly adjust the direction of the robotic manta to deal with deviations. For distant targets, a high-frequency swimming mode is used for rapid approach, while a low-frequency mode ensures accuracy upon arrival. This control strategy has great potential to improve the maneuverability, efficiency, stability, and adaptability of the robotic manta. However, the proposed control strategies focus solely on 2-D target point swimming control. Future research will improve the depth regulation device in hardware and combine 2-D swimming control with depth control tasks to train control strategies, ultimately achieving 3-D target point swimming control for the robotic manta.

Author Contributions: Conceptualization, S.S. and J.Z.; methodology, S.S.; software, Y.C.; validation, Y.C., S.S. and K.N.; funding acquisition, S.S.; resource, C.L. and S.S.; writing—original draft preparation, Y.C.; writing—review and editing, C.L. and S.S.; project administration, S.S. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Research Projects on Basic Sciences (Natural Sciences) in Higher Education Institutions of Jiangsu Province of China, grant number 23KJA460005.

Data Availability Statement: The study data can be obtained by email request to the authors.

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

References

1. Katzschmann, R.K.; Delpreto, J.; Maccurdy, R. Exploration of underwater life with an acoustically controlled soft robotic fish. *Sci. Robot.* **2018**, *3*, 12. [CrossRef]
2. Wang, J.; Wu, Z.X.; Dong, H.J. Development and Control of Underwater Gliding Robots: A Review. *IEEE CAA J. Autom. Sin.* **2022**, *9*, 1543–1560. [CrossRef]
3. Wu, Z.X.; Liu, J.C.; Yu, J.Z. Development of a Novel Robotic Dolphin and Its Application to Water Quality Monitoring. *IEEE ASME. Trans. Mechatron.* **2017**, *22*, 2130–2140. [CrossRef]
4. Wu, Y.H.; Duan, Y.H.; Wei, Y.G. Application of intelligent and unmanned equipment in aquaculture: A review. *Comput. Electron. Agric.* **2022**, *199*, 14. [CrossRef]
5. Yu, J.Z.; Wang, C.; Xie, G.M. Coordination of Multiple Robotic Fish With Applications to Underwater Robot Competition. *IEEE Trans. Ind. Electron.* **2016**, *63*, 1280–1288. [CrossRef]
6. Li, T.F.; Li, G.R.; Liang, Y.M. Fast-moving soft electronic fish. *Sci. Adv.* **2017**, *3*, 7. [CrossRef]

7. Thandiackal, R.; White, C.H.; Bart-Smith, H. Tuna robotics: Hydrodynamics of rapid linear accelerations. *Proc. R. Soc. B-Biol. Sci.* **2021**, *288*, 10. [CrossRef] [PubMed]
8. Zhu, Y.; Liu, Y.; Zhang, L.; Wang, Y.; Niu, W.; Huang, C. Dynamic model and motion characteristics of an underwater glider with manta-inspired wings. *J. Bionic Eng.* **2022**, *19*, 1–15. [CrossRef]
9. Cai, Y.; Bi, S.; Li, G.; Hildre, H.P.; Zhang, H. From natural complexity to biomimetic simplification: The realization of bionic fish inspired by the cownose ray. *IEEE Robot. Autom. Mag.* **2019**, *26*, 27–28. [CrossRef]
10. Zhang, Y.; Wang, S.; Wang, X.; Geng, Y. Design and control of bionic manta ray robot with flexible pectoral fin. In Proceedings of the 14th IEEE International Conference on Control and Automation (ICCA), Anchorage, AK, USA, 12–15 June 2018; pp. 1034–1039.
11. He, J.; Cao, Y.; Huang, Q.; Cao, Y.; Tu, C.; Pan, G. A new type of bionic manta ray robot. In Proceedings of the Global Oceans, Biloxi, MS, USA, 5–30 October 2020; pp. 1–6.
12. Meng, Y.; Wu, Z.; Dong, H.; Wang, J.; Yu, J. Toward a novel robotic manta with unique pectoral fins. *IEEE Trans. Syst. Man. Cybern. Syst.* **2020**, *52*, 1663–1673. [CrossRef]
13. Chen, L.; Qiao, T.; Bi, S.; Ren, X.; Cai, Y. Modeling and simulation research on soft pectoral fin of a bionic robot fish inspired by manta ray. *J. Mech. Eng.* **2020**, *56*, 182–190.
14. Yan, Z.P.; Yang, H.Y.; Zhang, W. Bionic Fish Trajectory Tracking Based on a CPG and Model Predictive Control. *J. Intell. Robot. Syst.* **2022**, *105*, 17. [CrossRef]
15. Hao, Y.W.; Cao, Y.; Cao, Y.H.; Huang, Q.G.; Pan, G. Course Control of a Manta Robot Based on Amplitude and Phase Differences. *J. Mar. Sci. Eng.* **2022**, *10*, 16. [CrossRef]
16. Chen, J.Y.; Yin, B.; Wang, C.C. Bioinspired Closed-loop CPG-based Control of a Robot Fish for Obstacle Avoidance and Direction Tracking. *J. Bionic Eng.* **2021**, *18*, 171–183. [CrossRef]
17. Qiu, C.L.; Wu, Z.X.; Wang, J.; Tan, M.; Yu, J.Z. Locomotion Optimization of a Tendon-Driven Robotic Fish With Variable Passive Tail Fin. *IEEE Trans. Ind. Electron.* **2023**, *70*, 4983–4992. [CrossRef]
18. Morgansen, K.A.; Triplett, B.I.; Klein, D.J. Geometric methods for modeling and control of free-swimming ffn-actuated underwater vehicles. *IEEE Trans. Robot.* **2007**, *23*, 1184–1199. [CrossRef]
19. Wang, Y.; Wang, R.; Wang, S.; Tan, M.; Yu, J.Z. Underwater Bioinspired Propulsion: From Inspection to Manipulation. *IEEE Trans. Ind. Electron.* **2020**, *67*, 7629–7638. [CrossRef]
20. Cao, Y.; Lu, Y.; Cai, Y.R. CPG-fuzzy-based control of a cownose-ray-like fish robot. *Ind. Robot.* **2019**, *46*, 779–791. [CrossRef]
21. Yang, Q.M.; Jagannathan, S. Reinforcement Learning Controller Design for Affine Nonlinear Discrete-Time Systems using Online Approximators. *IEEE Trans. Syst. Man. Cybern. Part B-Cybern.* **2012**, *42*, 377–390. [CrossRef]
22. Tong, R.; Feng, Y.K.; Wang, J.; Wu, Z.X.; Tan, M.; Yu, J.Z. A Survey on Reinforcement Learning Methods in Bionic Underwater Robots. *Biomimetics* **2023**, *8*, 29. [CrossRef]
23. Zhang, T.H.; Tian, R.; Wang, C.; Xie, G.M. Path-following Control of Fish-like Robots: A Deep Reinforcement Learning Approach. *IFAC PapersOnLine* **2020**, *53*, 8163–8168. [CrossRef]
24. Woo, J.; Yu, C.; Kim, N. Deep reinforcement learning-based controller for path following of an unmanned surface vehicle. *Ocean Eng.* **2019**, *183*, 155–166. [CrossRef]
25. Zhang, Z.Z.; Chen, J.L.; Chen, Z.B.; Li, W.P. Asynchronous Episodic Deep Deterministic Policy Gradient: Toward Continuous Control in Computationally Complex Environments. *IEEE Trans. Cybern.* **2021**, *51*, 604–613. [CrossRef] [PubMed]
26. Zhang, S.; Qian, Y.; Liao, P. Design and control of an agile robotic fish with integrative biomimetic mechanisms. *IEEE/ASME Trans. Mechatron.* **2016**, *21*, 1846–1857. [CrossRef]
27. Zhang, D.; Pan, G.; Cao, Y. A novel integrated gliding and flapping propulsion biomimetic manta—ray robot. *J. Mar. Sci. Eng.* **2022**, *10*, 924. [CrossRef]
28. He, Y.; Xie, Y.; Pan, G. Depth and Heading Control of a Manta Robot Based on S–Plane Control. *J. Mar. Sci. Eng.* **2022**, *10*, 1698. [CrossRef]
29. Meng, Y.; Wu, Z.; Chen, D. Development and 3–D Path—Following Control of an Agile Robotic Manta With Flexible Pectoral Fins. *IEEE Trans. Cybern.* **2024**, *54*, 3227–3738. [CrossRef] [PubMed]
30. Korkmaz, N.; Öztürk, İ.; Kiliç, R. Modeling, simulation, and implementation issues of CPGs for neuromorphic engineering applications. *Comput. Appl. Eng. Educ.* **2018**, *26*, 782–803. [CrossRef]
31. Yu, J.; Tan, M.; Chen, J.; Zhang, J. A Survey on CPG-Inspired Control Models and System Implementation. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 441–456. [CrossRef]
32. Cao, Y.; Bi, S.; Cai, Y.; Wang, Y. Applying central pattern generators to control the robofish with oscillating pectoral fins. *Ind. Robot. Int. J.* **2015**, *42*, 392–405. [CrossRef]
33. Lu, Q.; Zhang, Z.C.; Yue, C. The programmable CPG model based on Matsuoka oscillator and its application to robot locomotion. *Int. J. Model. Simul. Sci.* **2020**, *11*, 2050018. [CrossRef]
34. Feng, Z.J.; Hou, Q.; Zheng, Y.L. Method of artificial intelligence algorithm to improve the automation level of Rietveld refinement. *Comput. Mater. Sci.* **2019**, *156*, 310–314. [CrossRef]

35. Chen, Y.W.; Mehdi, J.; Luo, H. Model-free Reinforcement Learning in Infinite-horizon Average-reward Markov Decision Processes. *CoRR* **2019**, *119*, 10170–10180.
36. He, J.; Su, S.; Wang, H.; Chen, F.; Yin, B. Online PID Tuning Strategy for Hydraulic Servo Control Systems via SAC-Based Deep Reinforcement Learning. *Machines* **2023**, *11*, 593. [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.