AMRITA SCHOOL OF ARTIFICIAL INTELLIGENCE, BENGALURU

B. Tech in Artificial Intelligence and Data Science (AID)

Second Semester, Section F, Academic Year: 2023-24

COMPUTATIONAL MECHANICS II (23PHY114)

END-TERM PROJECT REPORT

on

UNDERWATER TARGET TRACKING CONTROL OF AN UNTETHERED ROBOTIC FISH WITH A CAMERA STABILIZER

Submitted by Group No. 1

AILENI CHANAKYA REDDY(BL.EN.U4AID23001)

SAI ROHITH ELLURU (BL.EN.U4AID23015)

MUDDANNA MANIKANTA (BL.EN.U4AID23031)

MADDI RISHI DHANESHWAR (BL.EN.U4AID23032)

PALLA CHINVITHA(BL.EN.U4AID23041)

Under the guidance of

Abhinav Sir

INTRODUCTION:

The development and deployment of bioinspired aquatic mechatronic systems have undergone significant advancements, leading to the use of robotic fish as a crucial tool for understanding aquatic dynamics. Early research focused on the precise mechanics of these designs, which evolved into the creation of Median-Paired Fin robots. Subsequently, Body-Caudal Fin (BCF) robotic fish gained prominence due to their superior characteristics, despite initial concerns about image stability.

To address these concerns, one approach involved optimizing swimming motion parameters, albeit at the cost of swimming speed. This led to the implementation of a pan-and-tilt camera system. However, deriving control laws in dynamic aquatic environments posed challenges due to parametric uncertainties and external disturbances affecting the accuracy of dynamic models.

Reinforcement Learning (RL), a machine learning methodology, emerged as a viable solution for motion control in robotic fish. RL's ability to adapt strategies to current environments reduces the need for intricate hydrodynamic modelling and leverages self-evolution capabilities. A notable advancement in this field is the proposal of a Deep Deterministic Policy Gradient (DDPG)-based target tracking controller. This innovative controller facilitates continuous tracking of selected targets, marking the first successful wide-range moving target tracking achieved with a self-propelled BCF-type robotic fish.

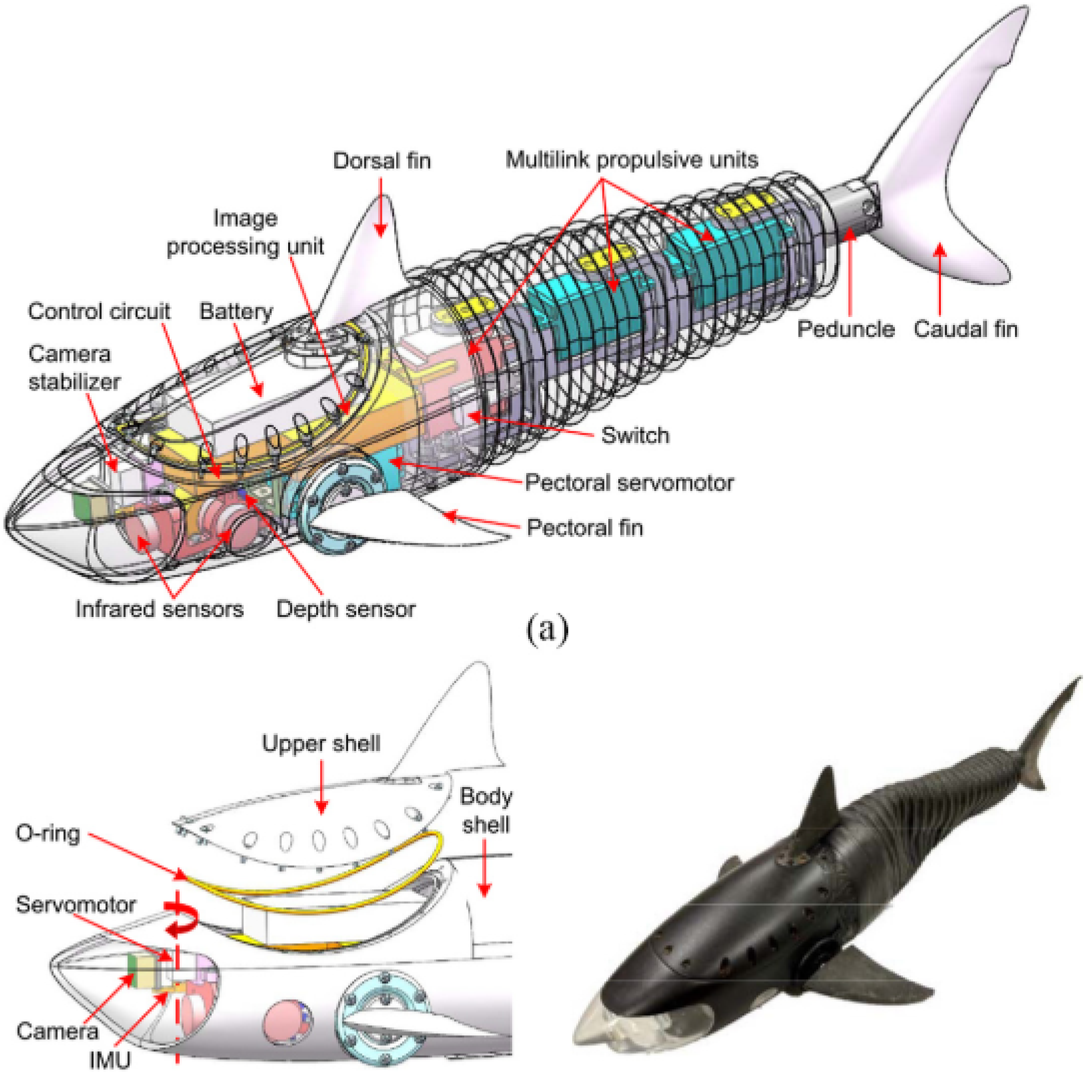
System Design of the Robotic Fish with Camera Stabilizer

Mechatronic Design

Inspired by the natural shark, a fast-swimming ocean predator, a shark-like robotic fish is developed. The robotic fish consists of a rigid anterior body with one pair of pectoral fins, and a multilink posterior body with a heterocercal caudal fin. To enhance underwater perceptual ability, an embedded vision system with a camera stabilizer is built.

For a BCF-type swimmer in motion, the fish head always sways inevitably due to the recoil effect from the left-to-right body undulation, which may lead to severe image blurring. Furthermore, the camera might lose the tracked target during the swaying because of the small field of vision. To reduce image blur and obtain a larger field of vision, we propose a novel mechanism for stabilizing the camera. The camera stabilizer has a motor rotating around the yaw axis in the range of -70° to 70°, responsible for modulating the yaw angle of the camera. Due to strict interior space limitations, a two-DoF camera stabilizer is unavailable, and a one-DoF camera stabilizer is adopted, consisting of an attitude measurement unit and a servomotor rotating around the yaw axis. Two inertial measurement units (IMUs) are mounted on the camera and the fish body to obtain the yaw angles of the camera and the robot body, providing feedback for controlling the camera stabilizer.

The resulting robotic prototype is shown in Figure 1. The detailed technical specifications of the robotic fish are tabulated in Table I. The tracking control implemented is only in the 2-D plane, making the control of pectoral fins for 3-D maneuvers unnecessary in the context of planar tracking control.



(a)

CPG-Based Motion Control

A CPG-based motion control method governs the swimming motions of the robotic fish. The CPG model comprises two chains of amplitude-controlled phase oscillators with controllable nearest neighbor coupling. The dynamics of the ith oscillator are described by the following nonlinear differential equations:

$$\dot{\theta} i = 2\pi\nu_i + \sum j \in T(i)\omega_{ij} \sin(\theta_j - \theta_i - \phi_{ij})$$

$$\ddot{\zeta}_i = a_i \left(a_i^4 (R_i - \zeta_i) - \dot{\zeta}_i \right)$$

$$x_i = \zeta_i \left(1 + \cos\theta_i\right)$$

Where θ_i and ζ_i denote the phase and amplitude of the ith oscillator, respectively; ν_i and R_i represent the intrinsic frequency and amplitude; a_i is a positive constant; and ω_{ij} and ϕ_{ij} determine the weights and phase biases of coupling between the ith and jth oscillators. $T(i)$ is the set of oscillators that the ith oscillator receives inbound couplings from. The output angle χ_i of the ith joint is calculated as:

$$\chi_i = x_i - x_{3+i}$$

Based on the settings in previous studies, the CPG model can yield signals for the set point of the angle of the servomotor of each joint, asymptotically converging to the following equation:

$$\chi_i^{\infty}(t) = (A_{Li} - A_{Ri}) + (A_{Li} + A_{Ri})\cos(2\pi\nu t + i\Delta\phi + \phi_0)$$

Where A_{Li} and A_{Ri} represent the intrinsic amplitudes of the left and right oscillators; $\Delta\phi$ indicates

Oscillator Model

The CPGs are modeled as systems of coupled nonlinear oscillators. The phase (θ_i) and amplitude (r_i) of each oscillator are the primary state variables. The equations governing these variables are:

Phase oscillator:

$$\dot{\theta} i = 2\pi n_i + \sum_j wijr_j \sin(\theta_j - \theta_i - \phi_{ij})$$

Where θ_i is the phase of oscillator i, n_i is its intrinsic frequency, w_{ij} is the coupling weight between oscillators i and j, r_j is the amplitude of oscillator j, and ϕ_{ij} is the phase bias between oscillators i and j.

Amplitude dynamics:

$$\dot{r}_i = a_i(R_i - r_i) - r_i$$

Where a_i is a positive constant that controls the rate of change of the amplitude, R_i is the intrinsic amplitude, and r_i is the current amplitude of oscillator i.

Oscillatory signal:

$$x_i = r_i(1 + \cos(\theta_i))$$

This represents the output signal from the oscillator, which is used to drive motor commands.

Saturation Function

To replicate the behavior observed in biological systems, the model introduces a piecewise linear saturation function for the drive signal d_i, modulating the intrinsic frequency and amplitude n_i and R_i according to a drive signal d:

$$n_i(d) = \begin{cases} 0 & \text{if } d < d_{\text{low}} \\ k_n(d - d_{\text{low}}) & \text{if } d_{\text{low}} \leq d \leq d_{\text{high}} \\ n_{\text{max}} & \text{if } d > d_{\text{high}} \end{cases}$$

$$R_i(d) = \begin{cases} 0 & \text{if } d < d_{\text{low}} \\ k_R(d - d_{\text{low}}) & \text{if } d_{\text{low}} \leq d \leq d_{\text{high}} \\ R_{\text{max}} & \text{if } d > d_{\text{high}} \end{cases}$$

Where k_n and k_R are constants that scale the intrinsic frequency and amplitude based on the drive signal, and n_{max} and R_{max} are the maximum values they can reach.

Coupling Parameters

The coupling parameters w_{ij} and phase biases ϕ_{ij} are set such that:

- The body CPG produces traveling waves when the limb CPG is not active (hypothesis 1).

- Limb oscillators have strong unidirectional couplings to body oscillators to enforce walking patterns (hypothesis 2).

Numerical Implementation

The model equations are implemented numerically to simulate the behavior of the salamander's locomotor system. This involves solving the differential equations for phase and amplitude dynamics using numerical integration methods (e.g., Euler or Runge-Kutta methods).

Active Vision Tracking System

Target tracking is essential for autonomous robots to either maintain focus on a static target or monitor a moving one. The robotic fish employs a small camera to capture underwater images, integrating the Kernelized Correlation Filter (KCF) algorithm, known for its low computational cost and high performance, to track the target effectively. The system combines two control strategies to create a stable active vision tracking system.

Feedback Forward Controller for Camera Stabilization

One of the significant challenges is maintaining image stability despite the oscillatory movements of the fish. To address this, a feedback-feedforward controller is designed for the camera stabilizer, as illustrated in Figure 2(a) of the paper. This controller actively rotates the camera towards the target to counteract the recoil effects on the fish's head. The goal is to keep the camera's attitude angle constant relative to the inertial frame, ensuring the target remains centered in the image regardless of the fish's movements.

The feedback-feedforward controller incorporates a Proportional-Derivative (PD) controller, chosen over a Proportional-Integral-Derivative (PID) controller for its superior dynamic performance. The onboard

Inertial Measurement Unit (IMU) measures the disturbances, assumed to be sinusoidal, and the feedforward control compensates for these disturbances effectively.

Cascade Control System for Time-Delay Compensation

Another critical issue is the time delay in image acquisition and processing caused by hardware and algorithmic constraints. To address this, a cascade control system is implemented. This system uses the feedback-feedforward controller as an inner loop and an active tracking controller as an outer loop to manage image jitter, which occurs at frequencies above 1 Hz and is intolerable without intervention.

The cascade control system mitigates the impact of delays, ensuring the robotic fish can maintain focus on the target even with significant processing delays. The performance of the camera stabilizing system under these conditions is demonstrated through various response tests, showing that the system can achieve acceptable stabilization and target tracking performance.

Performance Evaluation

The effectiveness of the active vision tracking system is evaluated through simulations and experiments. In one test, the system's response to a sinusoidal disturbance with a frequency of 1 Hz and an amplitude of $\pi/6$ is analyzed. The camera stabilizing system reduces the disturbance impact to approximately 6°, or 20% of the disturbance amplitude. The system achieves a target attitude angle with a step response time of about 0.1 seconds and a waveform-following response time of around 0.2 seconds, which is satisfactory for real-world visual tracking applications.

Tracking errors are compared between simulations and experiments. In simulations, the absolute error $|e_{ss}|$ is kept below 200 pixels, with no static error observed. Experimental results show a maximum error of 230 pixels, stabilizing to a sinusoidal wave within 4 seconds, consistent with simulation times. This consistency demonstrates the system's reliability in maintaining the target within the camera's Field of View (FoV).

Underwater Image Comparison

The performance of the active vision tracking system is further validated by comparing two snapshot sequences captured by the robotic fish's underwater camera. In the case without the active vision tracking system, images exhibit significant motion blur, leading to target detection and tracking failures. Conversely, with the active vision tracking system, the target remains centered in the snapshots, and images are stable and clear. This confirms that the active tracking system effectively reduces image instability and maintains target tracking accuracy even with delays in underwater image acquisition and processing.

Questions Asked

Why are the plots distorted?

The output plots look distorted due to a few reasons. One reason is that the input angle or position is not stable or smooth, resulting in a distorted waveform. Other reasons could be the disturbance coming from the water or the viscosity of water that makes the path or flow of the robot rough.

What is the function mentioned in the image-based tracking controlled system?

The function mentioned in the block diagram is discussed in the next section. This is the transfer function used to determine the output path or yaw-angle.

RL-Based Target Tracking Control

Building on the well-developed motion control and active vision tracking systems, it is feasible to achieve target tracking for robotic fish using reinforcement learning (RL). This paper introduces a deep Q-learning algorithm suitable for continuous control, specifically the Deep Deterministic Policy Gradient (DDPG) algorithm, as the tracking control method. DDPG leverages off-policy data and the Bellman equation to learn the Q-function, which in turn is used to derive the policy. The algorithm employs an actor-critic structure, where two neural networks represent the actor and critic functions from deterministic policy gradient (DPG). The actor network takes the current state as input and outputs an action, while the critic network evaluates the action's performance and updates the actor network's weights. The DDPG algorithm is detailed in Algorithm 1.

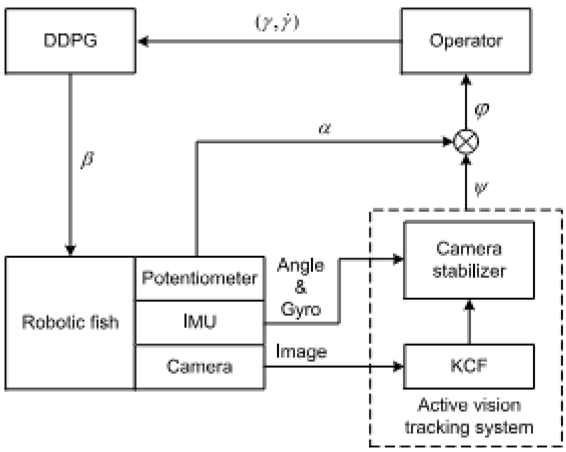


Fig. 7. Overall structure of target tracking control system.

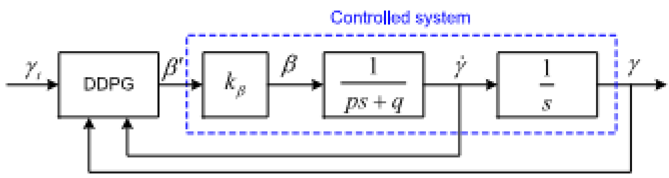


Fig. 8. Block diagram of the DDPG-based learning system.

Algorithm 1 DDPG

Randomly initialize the weights θ^Q and θ^μ of Critic network $Q(s, a|\theta^Q)$ and Actor network $\mu(s|\theta^\mu)$, and initialize weights $\theta^{Q'}$ and $\theta^{\mu'}$ of the target networks: $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$.

Initialize the experience replay pool R.

for episode $= 1$ to M **do**

  Initialize a random process N for exploring the action space.

  Initialize the system and obtain the initial state s_1.

  for $t = 1$ to T **do**

    Select an action $a_t = \mu(s_t|\theta^\mu) + N_t$ based on the current strategy.

    Execute action a_t to receive reward r_t and new state s_{t+1}.

    Store the transition (s_t, a_t, r_t, s_{t+1}) in the replay pool R.

    Sample a batch of transitions (s_i, a_i, r_i, s_{i+1}) from R.

    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$.

    Update the Critic network using the loss function:

$$L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$$

    Update the Actor network using the deterministic policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

    Update target network parameters:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$

Tracking Control Design

The robotic fish's camera, equipped with a stabilizer, can automatically orient toward the target object, enabling the inference of the relative position between the target and the fish body. The angle α represents the angle between the camera's optical axis and the fish body's longitudinal axis, measured via a potentiometer on the camera stabilizer motor. The angle ψ represents the angle between the camera's optical axis and the target, derived from the image using the KCF algorithm, the camera model, and internal parameters. The angle ϕ is the sum of α and ψ, as shown in the relationship $\phi = \alpha + \psi$. The goal of the target tracking control is to minimize ϕ.

The yaw angle γ of the robotic fish can be adjusted by changing the offset variable β in the CPG-based locomotion control. If $\gamma = -\phi$, the fish will direct toward the target. The displacement effect on ϕ can be ignored since the fish is usually far from the target. Therefore, continuously adjusting γ by modifying β enables the fish to swim toward the target.

The RL-based approach is advantageous for underwater robot control due to its adaptability in uncertain environments without requiring precise dynamic models. Consequently, a DDPG-based target tracking control method is adopted. The yaw angle dynamics are modeled as:

$$a\ddot{\gamma} + b\dot{\gamma} = \tau$$

where τ represents the thrust forces and moments from fishlike swimming, linearly related to β:

$$\beta = p\ddot{\gamma} + q\dot{\gamma}$$

Thus, the transfer function from β to γ is:

$$\frac{\Gamma(s)}{B(s)} = \frac{1}{ps^2 + qs}$$

This second-order system can be described by the state equation:

$$\begin{cases} \frac{d\gamma}{dt} = \dot{\gamma} \\ \frac{d\dot{\gamma}}{dt} = \frac{\beta}{p} - \frac{q}{p}\dot{\gamma} \end{cases}$$

The state vector $(\gamma, \dot{\gamma})$ ensures the target tracking control problem meets Markov decision process (MDP) conditions. To prevent non-linearity in β, it is normalized as:

$$\beta = k_\beta \beta'$$

where k_β is a discount factor. The reward function is designed to balance tracking error and angular velocity:

$$r_\gamma = \begin{cases} 2 - |\gamma| & \text{if } |\gamma| \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

To avoid overshoot and oscillation, an angular velocity-based reward function is:

$$r_{\dot{\gamma}} = \begin{cases} 1 - |\dot{\gamma}| & \text{if } |\dot{\gamma}| \leq 1 \text{ and } |\gamma| \leq 0.1 \\ 0 & \text{otherwise} \end{cases}$$

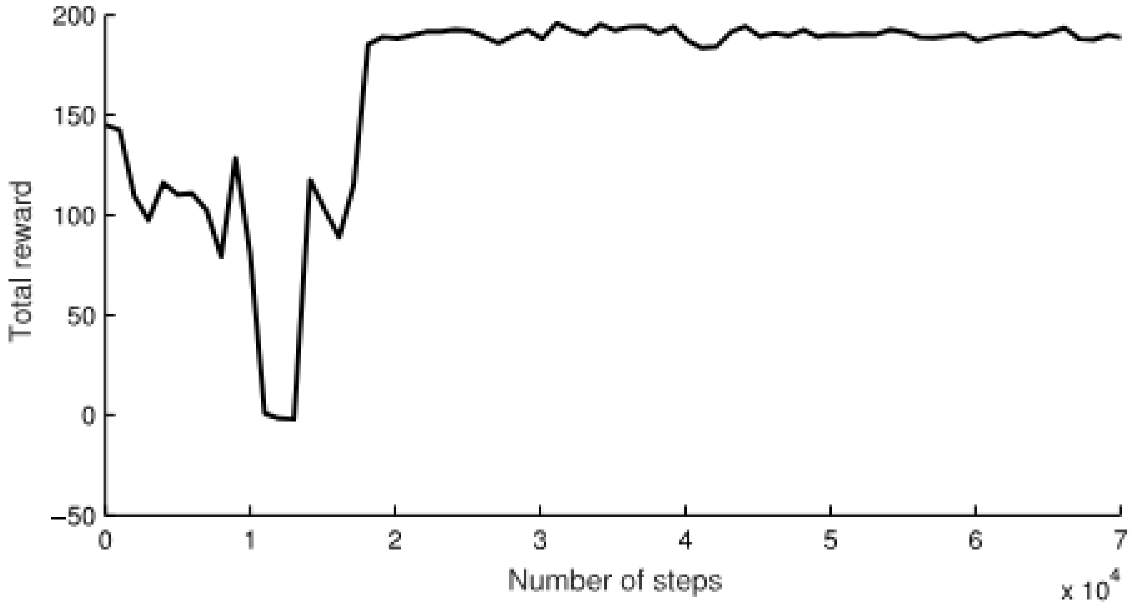
The combined reward function is:

$$r = \frac{w_\gamma r_\gamma + w_{\dot{\gamma}} r_{\dot{\gamma}}}{w_\gamma + w_{\dot{\gamma}}}$$

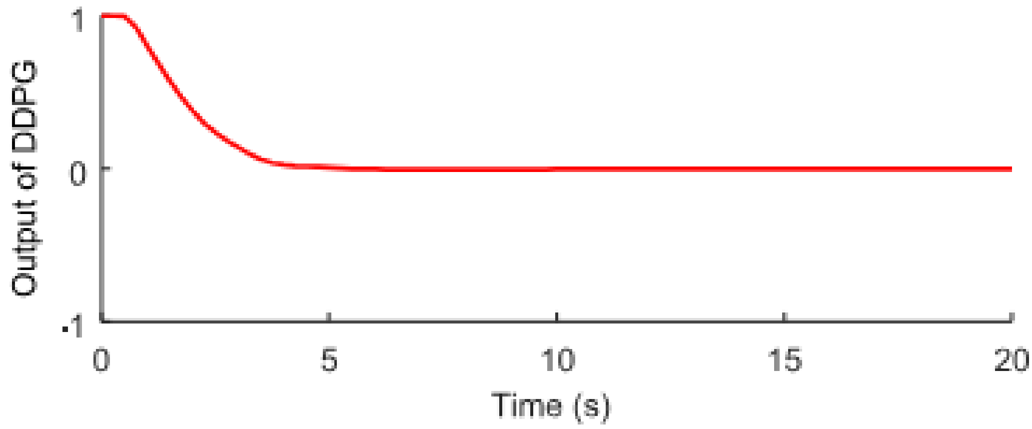
with weights $w_\gamma = 2$ and $w_{\dot{\gamma}} = 1$, maintaining the reward function range as $[0, 1]$. The termination condition is set as $|\gamma| > 2$, with a fixed return value $r = -10$.

The overall structure of the underwater target tracking control system is illustrated. The robotic fish, controlled by the CPG-based controller, sends underwater images to an onshore host computer for target detection using the KCF algorithm. The angle ψ is derived and used to adjust the camera's orientation. The angles α and ψ determine the expected yaw angle, which serves as the DDPG input to decide the robotic fish's turning direction via the CPG model's offset parameter.

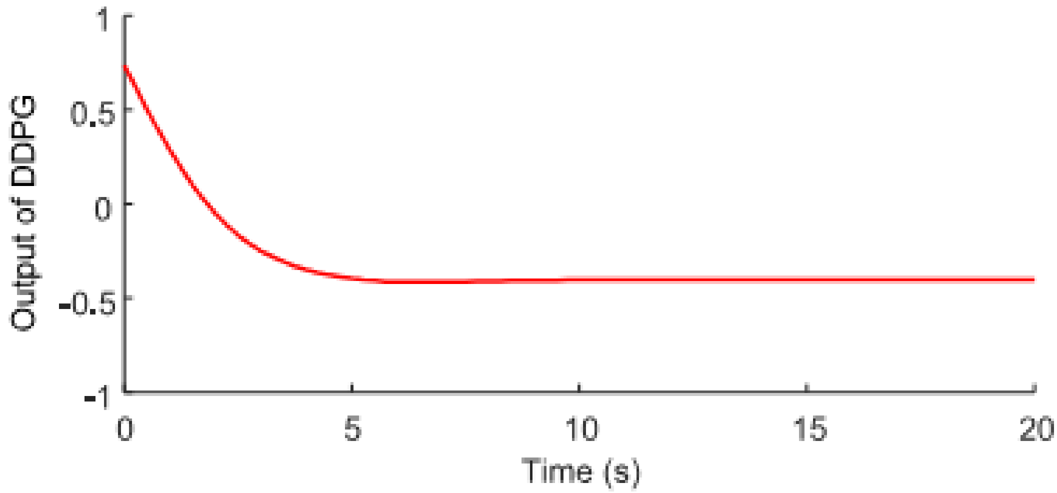
Performance Analysis of DDPG-Based Control System



To evaluate the DDPG algorithm's effectiveness, a simulated learning system was constructed. Using the transfer function $\frac{1}{ps^2+qs}$ and reward function r, the state vector $(\gamma, \dot{\gamma})$ and action vector β' were defined with parameters $p = 1$, $q = 1.25$, and $k_\beta = 0.5$. The simulation ran for a maximum of 200 steps per control period of 0.25 s. The total reward's variation over training steps showed the algorithm converging around 20,000 steps. The step response indicated quick adaptation and no overshoot, with a transient period of approximately 5 s.



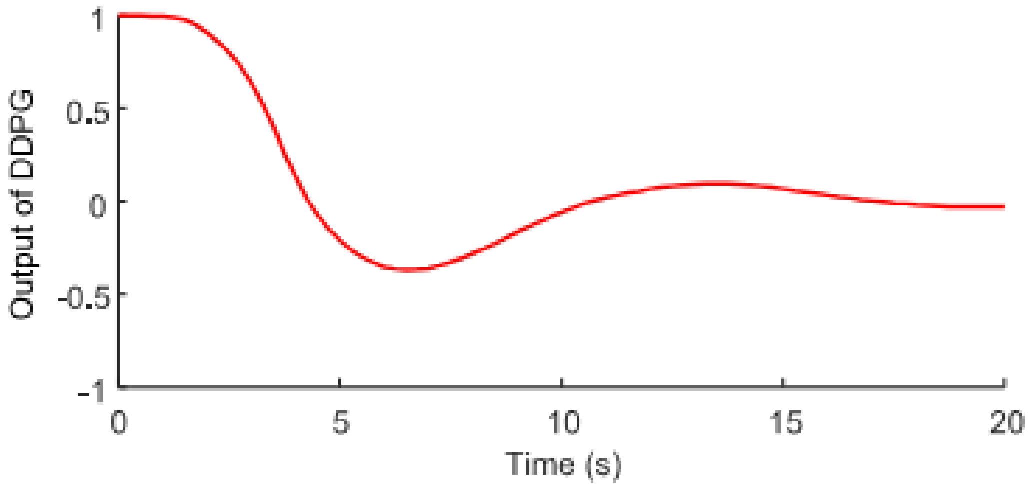
The stability analysis using the open-loop transfer function's Bode plot revealed a gain margin of 51.5 dB and a phase margin of 85.3°, confirming system stability. Adaptability was tested with a static bias of 0.2 in the fish tail's moving joints. The system adapted despite a steady-state error.



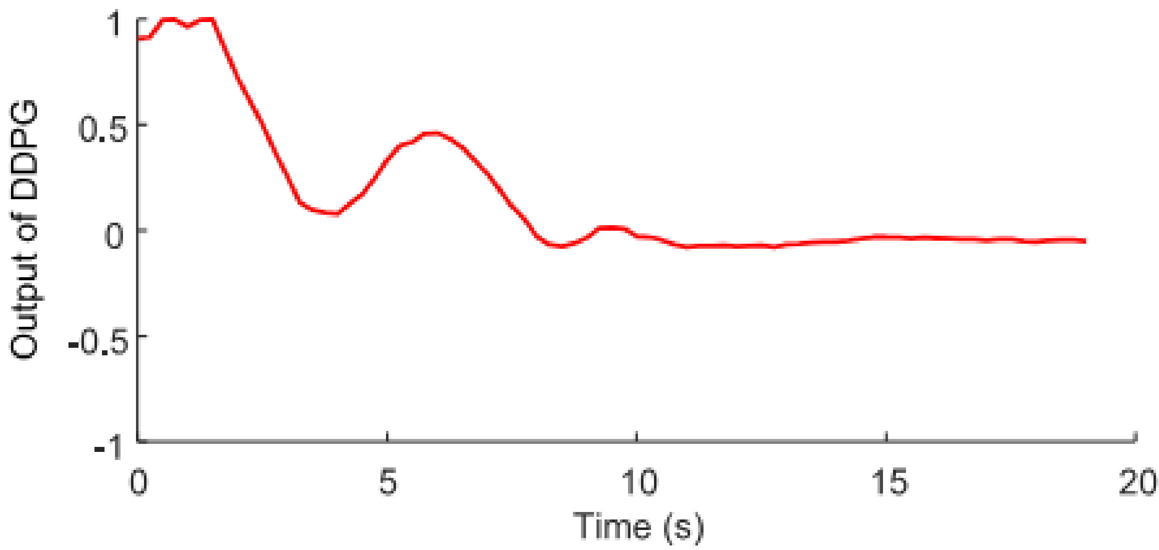
Time-delay effects were also examined, representing filters, image acquisition, and CPG-based control as an equivalent hysteresis element with a transfer function:

$$G_{delay} = e^{-T_{delay}s} \approx \frac{1}{T_{delay}s + 1}$$

Setting $T_{delay} = 2$, the step response shows an adjustment time of 13 s and a steady-state error of 0.15, indicating maintained tracking accuracy.

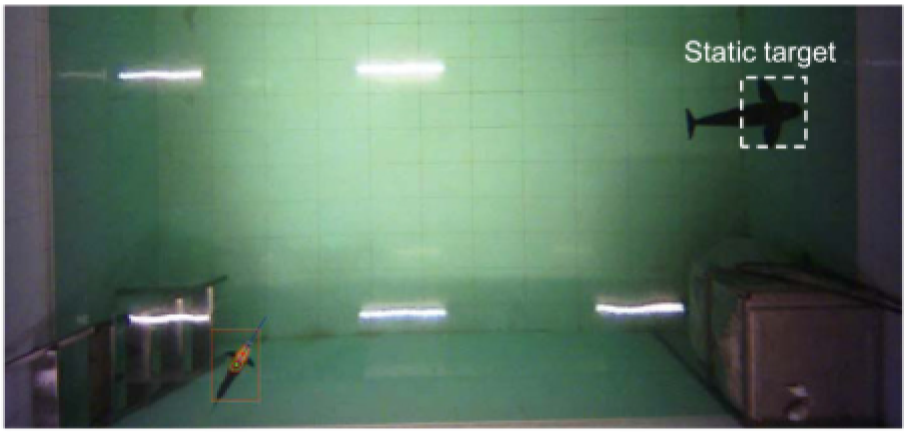


Simulation results confirmed the DDPG-based control system's efficacy, demonstrating both stability and adaptability. An underwater tracking experiment verified the system's robustness and real-time application.

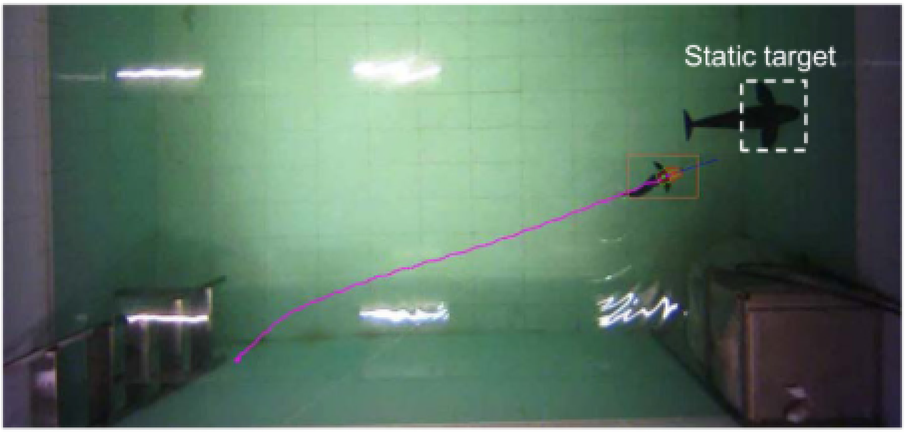


Experiments and Results

To test the new RL-based control system for vision-guided robotic fish, we conducted experiments in a small indoor swimming pool of 5 x 4 x 1.5 meters. We used a motion measurement system with a camera hanging 1.9 meters above the sea level to capture snapshots of the experiment. The role of this camera was not to assist the fish in tracking lines; it was solely to record all conducted trials. To record all the trials, we used software for tracking the target squarely on the machine that transmitted real-time position together with velocity information to the robotic fish.



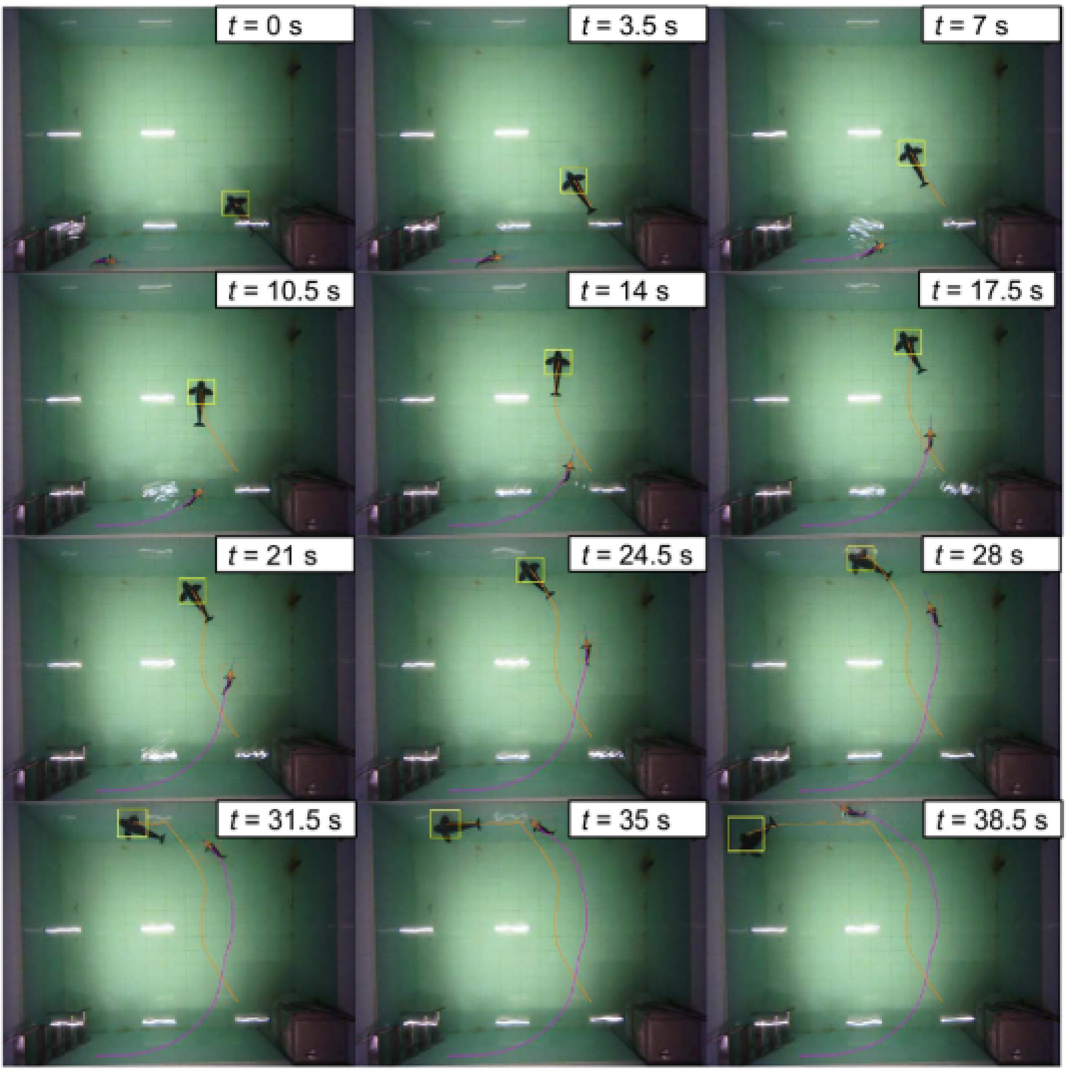
(a)



(b)

In the experiment, we first tested how the system responded to sudden changes. The DDPG-based learning, which was used in simulation, stayed stable even with disturbances. It took about 10 seconds to adjust in testing, compared to 5 seconds for perfect simulation and with 13 seconds time delays. The system was very accurate, with an error of less than 0.1, and followed a smooth path with respect to the simulation.

We also compared the DDPG system with a conventional PID controller. The PID controller reacted faster, reaching the peak value within 1 second, but it had some limitations. It overshot by more than 0.5 radians and had a steady-state error of 0.2 radians. On the other hand, the DDPG system was slower but provided smoother and more stable tracking without steady-state errors.



In another experiment (Fig. 18), the robotic fish tracked a moving target, a robotic fish dolphin, for 40 seconds. During this experiment, the robotic fish followed the dolphin. The delay in response time was observed with the DDPG system itself. The DDPG system kept the angular error around 0.5 radians, proving to be both effective and adaptable.

Questions and Answers

1) In the given research paper, is the targeting object in Fig. 16 fixed or not?

The target object in the figure is fixed. The label ”Static target”in both snapshots (a) and (b) indicates that the target remains in the same position throughout the test. The trajectory of the robotic fish is plotted as it moves toward this fixed target. This was necessary to first check that the robotic fish was taking snapshots or not, so they fixed the targeting object. In Fig. 18, the targeting object is moving by 3.5-3.5 seconds.

2) What is hysteresis? And which standard have you studied it previously?

Hysteresis in robotics refers to the lag between the input signal and the system’s output response, often due to friction, backlash, or other mechanical properties. This can affect the accuracy and precision of robotic movements. We studied it in 8th or 9th class, sir.

Conclusion

Robotic vision systems play a crucial role in target tracking across various environments. Tracking underwater targets represents a problem that requires many unique solutions owing to ever-changing aquatic settings as well as the behavior of the robots involved. In this study, we propose a new camera stabilizing system and a DDPG-based learning system to improve the stability and flexibility of image-based tracking of untethered robotic fish. Key enhancements include a feedback-feedforward controlled camera stabilizer and adaptive configuration.