Emulating Underwater Locomotion: Design and Development of CPG-Controlled Biomimetic Robotic Fish

Sourish Varanasi[1], Aditya Bisla[2], Jyotindra Narayan[3], Bhavik Patel[4], and Santosha K. Dwivedy[5]

***Abstract*—This paper presents the design, modeling, control, and experimental validation of a biomimetic robotic fish that emulates thunniform locomotion. Motivated by the challenges of traditional underwater vehicles in terms of maneuverability and adaptability, the study aims to leverage bio-inspired propulsion strategies to enhance aquatic navigation in constrained environments. The fish robot is designed and developed with a hydrodynamic structure, incorporating active and passive fins and flexible joints to mimic natural fish movements. The dynamic and kinematic models are derived using Lagrangian mechanics, and a Central Pattern Generator (CPG)-based control scheme is implemented to generate rhythmic joint actuation without requiring precise trajectory planning. Simulation studies conducted in MATLAB/Simulink demonstrate smooth transitions from static to straight-line and turning maneuvers with a maximum lateral deviation of under 2 cm and a root mean square trajectory error of 0.0143 m. Experimental validations, both mid-air and in a water tank, confirm the sinusoidal motion patterns and verify the effectiveness of the control strategy. The results showcase the robot's capability for stable and lifelike planar swimming, offering a promising platform for further developments in autonomous underwater systems.**

***Index Terms*—robotic fish, thunniform locomotion, CPG, underwater navigation, bioinspired propulsion**

I. Introduction

In recent years, there has been a significant increase in the research and development of bio-mimetic robots, with a wide range of applications, including underwater exploration, water quality monitoring, surveillance, and navigation in hazardous terrains and environments [1]. Traditional underwater vehicles, such as remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs), rely on propeller-based propulsion systems, which are often limited in terms of maneuverability, energy efficiency, and adaptability in complex environments [2]. In contrast, bio-mimetic robots leverage nature's evolutionary designs to achieve superior performance [3]. Due to their slender structure and reduced hydrodynamic disturbance, these robots can cover long distances with minimal impact on the surrounding flora and fauna. Robots like Open Fish [4] utilizes a wire-driven compliant fin to mimic, Boxybot [5] uses a 3 fin model for swimming and crawling, Robopike [6] uses a constraint mechanism for planar bending, and SoFi [7] utilizes fluid elastomers for controlled directional bending, are able to generate a sinusoidal waveform that propels them forward, providing better maneuverability than traditional AUVs. These features make bio-mimetic fish robots ideal for underwater surveillance in hard-to-reach locations.

In order to fulfill bio-mimetic robots' requirement to be aerodynamically efficient and flexible, the design of the robot has to be hydrodynamic, optimizing speed and efficiency. Although motors are the most common method of actuation, these robots could be maneuvered using pneumatic actuators, smart material actuators or compliant mechanisms as well. These robots involve caudal fins and pectoral fins to increase efficiency and improve maneuverability. To achieve various types of underwater movement, a range of actuation and control strategies have been employed. These include simple DC motors [4] and servo motors [5], which utilize motor motion to actuate fish robots; hydraulic and pneumatic actuation [8], which employ small actuators to manipulate the fish robot's spine in the desired direction; Ionic Polymer Metal Composites (IPMC) [9], which consist of metal materials that can be easily deformed using chemical or electrical stimulants; and piezoelectric materials [10], which induce strain in metal when an electrical field is applied, facilitating the bending of the fish robot in the desired manner, among others.

Several Traditional and Non-Traditional control strategies have been successfully implemented in these robots. Yu et al. [11] utilizes a simple PID controller for desired speed and a fuzzy-logic orientation controller. Wang et al. utilizes Particle Swarm Optimization to optimize a LQR controller for trajectory tracking [12]. Dong et al. [13] employs a back-stepping and sliding mode control methodology for robustness. Castaño et al. [14] uses a Non-linear Model Predictive Control method using a simplified averaged robot model for trajectory control. Chen et al. implemented a stochastic optimization control algorithm which uses an artificial neural network model of the robot to generate the required control signal

[1]Sourish Varanasi is with the Department of Mechanical Engineering, Indian Institute of Technology Guwahati, Assam, India. sourishkumar9@gmail.com

[2]Aditya Bisla is with the Department of Mechanical Engineering, Indian Institute of Technology Guwahati, Assam, India. aditya.bisla2002@gmail.com

[3]Jyotindra Narayan is with the Department of Mechanical Engineering, Indian Institute of Technology Patna, 801106, Bihar, India. He is also with the Department of Computing, Imperial College London, UK. n.jyotindra@gmail.com; jnarayan@iitp.ac.in; jnarayan@ic.ac.uk

[4]Bhavik Patel is with the Department of Mechanical Engineering, Indian Institute of Technology Guwahati, 781039, Assam, India. bpatel2195@gmail.com

[5]Santosha K. Dwivedy is with the Department of Mechanical Engineering, Indian Institute of Technology Guwahati, Assam, India. He is also with the Center for Intelligent Cyber Physical Systems, Indian Institute of Technology Guwahati, 781039, Assam, India. dwivedy@iitg.ac.in

[15]. One such model-free method is based on Central Pattern Generators (CPG) [16]. Central Pattern Generators (CPG) are neural circuits responsible for rhythmic outputs even in the absence of sensory feedback from peripheral nervous system. These cells are responsible for basic rhythmic activities such as breathing, chewing, walking, and fin propulsion in the case of fishes. CPG's eliminate the need for trajectory planning and accurate knowledge of system dynamics which are difficult to solve accurately. Bio-inspired CPG control has demonstrated remarkable success across a wide range of robotic applications. These include legged robots, such as bipedal and humanoid systems, quadrupeds, hexapods, and even reconfigurable robotic platforms. Additionally, CPG controllers are model-free and have been effectively applied to crawling robots, enabling adaptive and efficient locomotion in various environments.

Adding the aforementioned challenges, this paper presents the comprehensive development of a bio-mimetic fish robot, encompassing its design, kinematic and dynamic modeling, low-level CPG controller implementation, and experimental testing. The robot's design, which has been created from scratch, emphasizes a streamlined structure for efficient and lifelike movement in aquatic environments and detailed kinematic and dynamic models that simulate its motion and interactions with water. A Central Pattern Generator (CPG) controller is developed to enable smooth transitions between different forms of swimming. Lastly, the experiments conducted with the fish robot are presented.

II. Design and Prototype Development

A. Robotic components

While designing the outer body of the fish robot, inspirations were taken from various other existing fish models, and created the design by taking inspiration from those models. The head of the fish robot had to be designed to be aerodynamically proficient to minimize drag. A smooth contour was chosen for the head shape. The robotic fish contains two robotic links that actuate the robot in water. Both links encase servo motors with 1.2 Nm stall torque and 62.5 RPM at 6.6V, which create the robot's actuation. These robotic links are connected using connectors. This robot used two fins: a pair of caudal fins and a pectoral fin. NACA 0012 airfoil was used as a cross-section for the fins. Figs. 1(a)-(d) presents the images of parts of the fish robot. The list of fish robot components is given in Table I.

B. Buoyancy analysis

Buoyancy analysis was conducted for the fish robot to ensure proper actuation in water bodies. The total weight of the robot came out to be 4.68 N. The net water replaced by the model came out to be 565.08 cc. Consequently, the weight of the displaced water comes out to be 5.563 N. The robot has a center of mass at (42.71, 61.65, 37.39) mm from the origin and the center of buoyancy for the robot comes out to be around (7.77, 61.62, 37.14) mm from the origin. The pictorial representation of the coordinates can be seen in Fig. 2. There would be a significant torque about the z-axis of about 0.155 N-m. Additional adhesive weights are applied to the head of the robot to stop the robot from toppling.

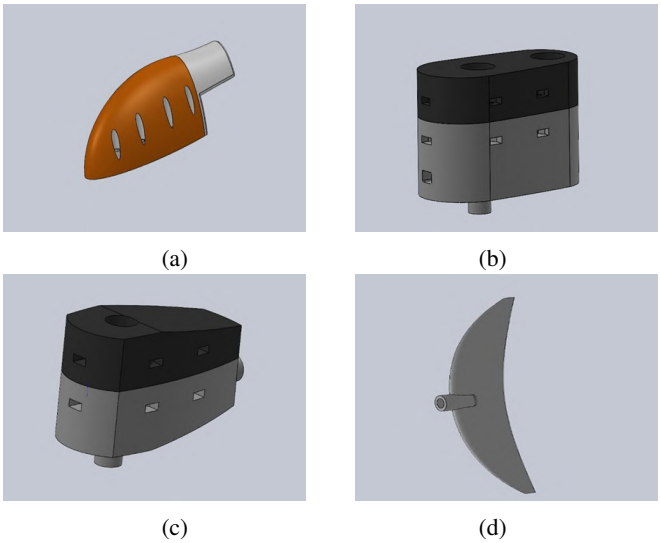


Fig. 1: Components of the fish robot (a) head, (b) first motor link, (c) second motor link, and (d) Caudal fin

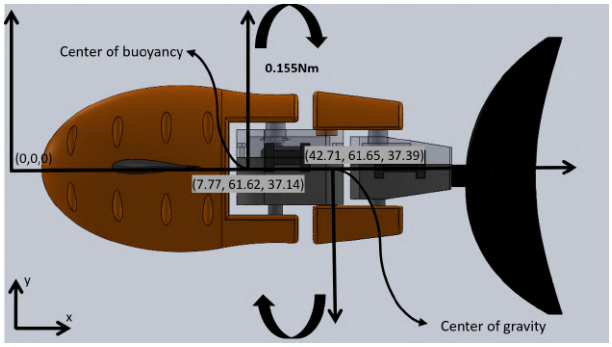
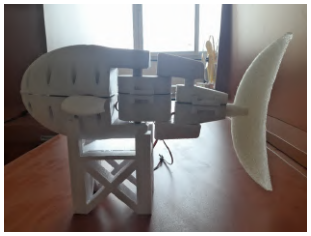


Fig. 2: Torque action representation

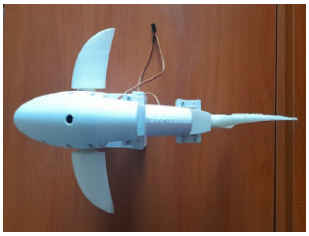
C. Manufacturing and assembling the model

The designed parts are printed using PLA (Polylactic Acid). The fins (caudal and pectoral) were printed using TPU (Thermoplastic Polyurethane). TPU has rubber-like properties, making it flexible and suitable for creating propulsion similar to a fish's fin. Sealing and reinforcements were properly applied to robot parts before testing them in water. External components like motors, bearings, etc., were also used in the robot. Figs. 3(a) and 3(b) contain the images of the manufactured robot.

Reinforcements were put on the robot to make it waterproof and buoyant. Silicon-based water sealant was applied to the openings and pores of the fish model to seal any potential leaks or gaps effectively. The robot was also attached with some extra weights to balance the toppling torque. Foam material was applied to the fins to help the fish float properly. The foam also acts as an insulant to water, further protecting the robot from water infiltration.



(a)



(b)

Fig. 3: (a) Side and (b) top view of the fish robot

TABLE I: Components of fish robot

Component	Quantity
Head Links	2
1st Link (Upper)	1
1st Link (Lower)	1
2nd Link (Upper)	1
2nd Link (Lower)	1
1st and 2nd Link connector	2
Caudal Fin	1
Pectoral Fin	2
Motor for Link 1	1
Motor for Link 2	1
Microcontroller	1
Battery	1
Bearings	2

III. Modeling of the Fish Robot

The robotic fish consists of 3 links, including a rigid head and flexible tail links. The fish head dimension is 165 mm, the link length is 74 mm, the caudal fin is 70 mm, and the total fish robot length is 344.5 mm. The optimal link lengths were determined using Particle Swarm Optimization. The objective function (3) aimed to minimize the effective envelope area between the fish's traveling wave motion (1) and a straight reference line (2). Equation (3) defines the objective function used for optimization, y_{body} corresponds to the traveling wave described by equation(1b) and y_{link} represents the straight-line equation of link i, $x_{d_{i-1}}$ and x_{d_i} denote the x-coordinates of the endpoints of link i and N is the total number of links. Let θ_i denote the absolute angles with respect to the World Coordinate System, and ϕ_j the relative joint angles between the links. To obtain the joint angles for swimming, the fish link equations are curve fitted with the actual equation for straight line swimming.

$$y_{body}(x,t) = (c_1 x + c_2 x^2) sin(kx + \omega t) \tag{1a}$$

$$y_{body}(x,t) = (c_1 x + c_2 x^2) sin(kx + \frac{2\pi}{M} m) \tag{1b}$$

$$(x_{ij} - x_{ij-1})^2 + (y_{ij} - y_{ij-1})^2 = d_j^2 \tag{2}$$

$$E = \sum_{i=1}^{N} e_i(x) = \sum_{i=1}^{N} \int_{x_{d_{i-1}}}^{x_{d_i}} |y_{\text{body}}(x) - y_{\text{link}}(x)|^2 \, dx \tag{3}$$

In (1), c_1 and c_2 are the weights of the quadratic amplitude envelope, while k and ω represent the wavenumber and angular frequency, respectively. The term $\frac{2\pi}{M} m$ discretizes the temporal evolution of the waveform over one full cycle, where $m = 0, 1, 2, \ldots, M-1$. Assuming the origin (x_0,y_0)=(0,0), (1b) and (2) are solved at each discrete time step m, providing the link equations from which the relative angles between the links, ϕ_i, are determined.

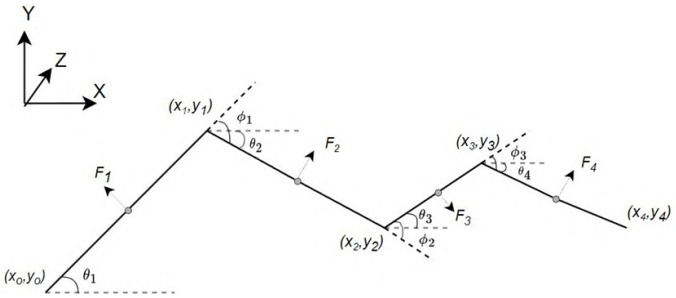


Fig. 4: Simplified robotic fish and definition of parameters, θ_1 is the angle made by link 1 in world-frame whereas ϕ_1 is the relative angle between link1 and link2. F_1 shows the force acting on the surface of link1

A. Dynamic Model

The dynamic model of the robotic is constructed using the Lagrangian function. The generalized coordinates $(q_1, q_2, q_3) = (X, Y, \theta)$ are taken to be $X = x_1, Y = y_1, \theta = \theta_2$ representing the coordinates of the first actuator in world frame. Fig. 4 represents the free body diagram of the fish robot. The other parameters with respect to the generalized coordinated are shown below:

$$\begin{aligned} L(q,\dot{q}) &= K(q,\dot{q}) - E(q) \\ \frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_i}) - \frac{\partial L}{\partial q_i} &= \tau_i \end{aligned} \tag{4}$$

$$\begin{aligned} L &= \sum_{i=1}^{N} \frac{1}{2} m_i v_i^2 + \sum_{i=1}^{N} \frac{1}{2} I_i \omega_i^2 - E_p \\ &= \sum_{i=1}^{N} \frac{1}{2} m_i (\dot{x_i}^2 + \dot{y_i}^2) + \sum_{i=1}^{N} \frac{1}{2} I_i \dot{\theta_i}^2 - E_p \end{aligned} \tag{5}$$

In (3) and (4), N represents the total number of links and q_i is the generalized coordinate. m_i and I_i define the mass and moment of inertia of the links based on the generalized coordinates. The hydrodynamic forces acting on the robot can be written as the sum of the force acting perpendicular to the link $F_i^{\perp}$ and the force acting parallel $F_i^{\parallel}$ as shown in (6a). (6b) and (6c) represent the perpendicular and parallel forces acting on each link i, where $\mu_i^{\perp}$, $\mu_i^{\parallel}$ are perpendicular and parallel drag coefficients and $v_i^{\parallel}$, $v_i^{\perp}$ are the links velocities.

$$F_i = F_i^{\perp} + F_i^{\parallel} \tag{6a}$$

$$F_i^{\perp} = \mu^{\perp} sgn(v^{\perp})(v^{\perp})^2 \tag{6b}$$

$$F_i^{\parallel} = -\mu^{\parallel} v^{\parallel} |v^{\parallel}| \tag{6c}$$

$$F_x = \sum_{i=1}^{N} F_{x_i} = \sum_{i=1}^{N} F_i^{\perp} \sin\theta_i + F_i^{\parallel} \cos\theta_i \tag{6d}$$

$$F_y = \sum_{i=1}^{N} F_{y_i} = \sum_{i=1}^{N} F_i^{\perp} \cos\theta_i + F_i^{\parallel} \sin\theta_i \tag{6e}$$

$$M_\theta = \frac{d}{dt}\frac{\partial L}{\partial \theta} - \frac{\partial L}{\partial \theta} = \sum_{i=1}^{N}[-F_{x_i}(y_i^f - Y)] + \sum_{i=1}^{N} F_{y_i}(x_i^f - X) \tag{6f}$$

Equations (6d) and (6e) represent the total force acting on the robot along the X and Y axes, respectively. In the moment equation (6f), x_i^f and y_i^f denote the center coordinates of link i. This mathematical model was used to simulate the fish robot's dynamics in MATLAB Simulink.

IV. Controller Design

The current model of CPG used in the robotic fish is inspired from the one used by Crespi et al. [17]. The motion of each joint can be represented as a harmonic oscillator. CPG control algorithms were applied to the robot to achieve sinusoidal motion. Fig. 5 provides a schematic of the controller.

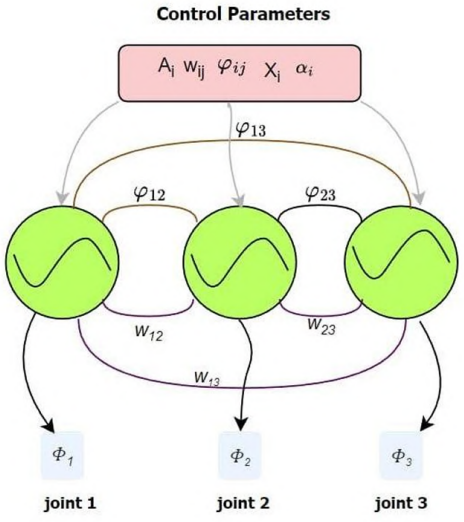


Fig. 5: Circuit schematic of CPG controller

At any instant, let x_i, a_i, and ϕ_i represent the offset, amplitude, and phase state, respectively. Second-order critically damped systems are used for the offset (step 7 in Algorithm 1) and amplitude (step 6 in Algorithm 1), with β_r, β_x being the equation's weights. Phase update equations are based on the Kuramoto model of synchronized oscillators, a second-order coupled non-linear differential equation. φ_{ij} represents the phase difference between i^{th} and j^{th} joint respectively. Weights are based on standard values, and the phase difference between the joints is obtained during the kinematic model of

Algorithm 1 CPG Control Parameter Update Algorithm

1: Initialize α_i, β_r, β_x, and ξ_{ij}, W_{ij} matrices
2: **repeat**
3: **if** locomotion pattern transition **then**
4: Update CPG control parameters (x_{d_i}, a_{d_i}, ϕ_i) desired values
5: **end if**
6: Update amplitude a_i and its derivative $\dot{a}_i$

$$\ddot{a}_i = \beta_r \left(\frac{\beta_r}{4}(a_{d_i} - a_i) - \dot{a}_i \right)$$

7: Update offset x_i and its derivative $\dot{x}_i$

$$\ddot{x}_i = \beta_x \left(\frac{\beta_x}{4}(x_{d_i} - x_i) - \dot{x}_i \right)$$

8: Update phase ϕ_i and its derivative $\dot{\phi}_i$

$$\dot{\phi}_i = \omega + \sum_{j \neq i}^{3} \alpha_i w_{ij} \sin(\phi_i - \phi_j - \varphi_{ij})$$

9: Update output angle θ_i

$$\theta_i(t) = x_i + a_i \sin(\omega t + \phi_i)$$

10: **if** $\min(\phi_i) \geq 2\pi$ **then**
11: $\phi_i' \leftarrow \phi_i + 2\pi$
12: **end if**
13: **until** termination condition is met

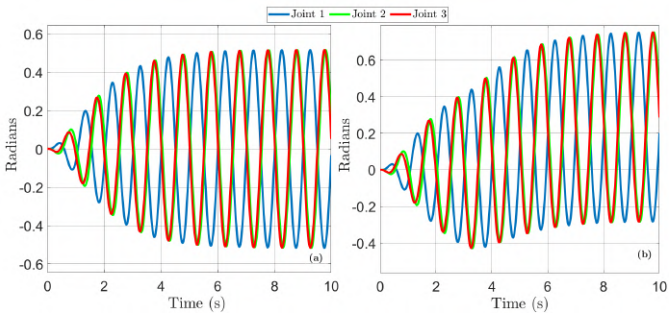


Fig. 6: Joint trajectories for (a) straight-line swimming (b) straight-line pattern to a turning motion

the robot. An offset ΔX_i is added to each joint to obtain turning motion. The control update process is provided in Algorithm 1. Given the desired offset x_d, amplitude a_d, and phase ϕ_d, steps 6, 7, and 8 in Algorithm 1 are solved to compute the instantaneous joint angle using step 9 in Algorithm 1. Fig. 6(a) illustrates the transition from a static state to straight-line motion, while Fig. 6(b) demonstrates the further transition from straight-line motion to turning, which is achieved solely by adjusting the desired offset.

V. Results and Discussion

Experiments were conducted to validate the control algorithms developed. The simulated trajectory of the robot was developed which was implemented in still water body.

A. Simulation results

Multiple simulations were conducted to estimate parameters such as force, moment, and the trajectory of the fish robot. These simulations provided an approximation of the robot's actual motion in still water. Fig. 7(a) illustrates the trajectory of the first joint in the x-y plane over a 20s simulation. It clearly demonstrates a smooth transition from a stationary position to straight-line swimming, with a maximum lateral deviation of less than 2 cm. Given the desired trajectory of a straight line along the x-axis, the observed mean absolute error and root mean square error are 0.0125 m and 0.0143 m, respectively. Fig. 7(b) illustrates the transition from a static position to straight-line swimming and then to a turning motion over a 20s simulation, with the shift to the turning pattern occurring at the 10s mark by introducing a 10-degree offset. The joint angle trajectories are similar to those shown in Fig. 6.

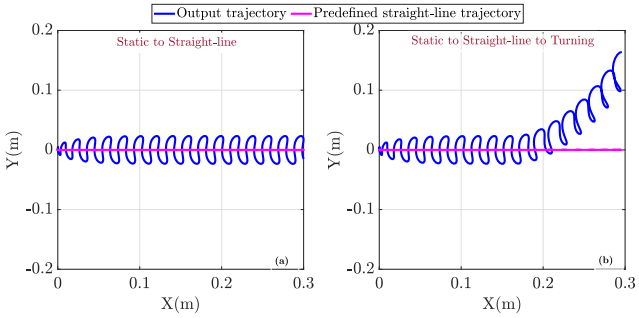


Fig. 7: Transition from (a) static to straight-line motion (b) static to straight-line to turning where a non-zero positive offset is introduced at the 10s mark, causing the robot to deviate and move in the positive xy direction.

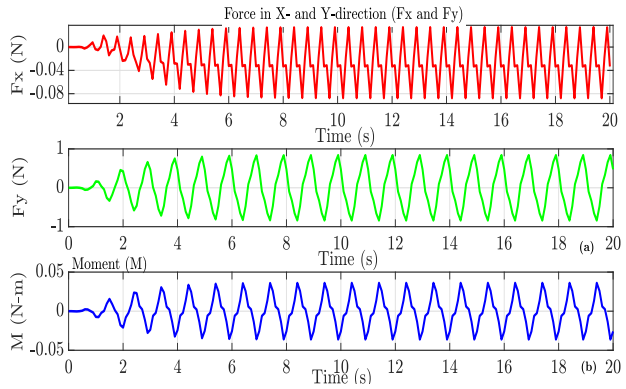


Fig. 8: Forces in (a) Longitudinal and lateral directions (x- and y-directions) and (b) moment acting on the first link.

Fig. 8 illustrates the oscillatory behavior of forces and moments acting on a robotic link due to the harmonic motion of its joints. In Fig. 8(a), the force in the X -direction (F_x) varies between approximately -0.08 N and -0.02 N , indicating the periodic forward and backward thrust generated during the link's motion. The lateral force in the Y -direction (F_y) exhibits larger oscillations, ranging from -1 N to +1 N , and reaches its maximum magnitude just under 1 N , which reflects the significant effort required to propel the link laterally against the resistance of the surrounding water. In Fig. 8(b), the moment (M) acting on the link also shows a distinct oscillatory pattern, fluctuating between -0.05 Nm and +0.05 Nm, corresponding to the cyclic torques generated by the joint's angular movements. These periodic force and moment variations are characteristic of the robot's harmonic joint activity and highlight the dynamic mechanical interaction between the link and the fluid environment.

B. Physical experimentation

After generating plots and a trajectory for the motion of the fish robot, the algorithm was implemented on the fish robot to validate the simulations. To observe the wave pattern generated by the fish robot, it was actuated mid-air on a stand. The sinusoidal motion created by the robot was observed. Images from that experiment can be seen in Fig. 9(a)-(d).

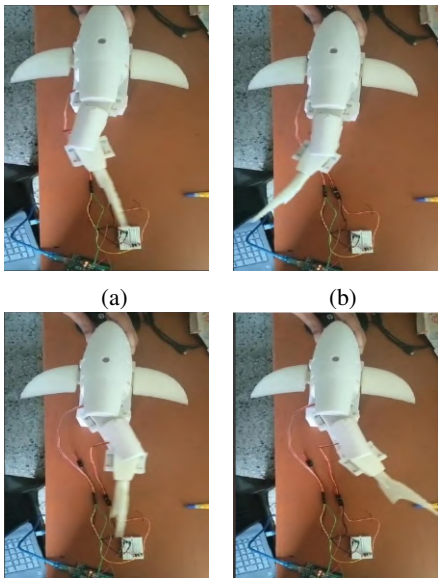


Fig. 9: Sinusoidal motion of the fish robot with (a) starting point, (b) left-ward tail tilt, (c) return to center, and (d) right-ward tail tilt

After actuating the robot mid-air, it was actuated in a water tank to observe the trajectory of the robot in a still water body. The robot was insulated against water using a silicon-based adhesive. A 1x3m dimension tank was taken for the experiment and filled 3/4th. Despite the reinforcements, some rotating parts had crevices for water entry, so it was made sure that the fish robot stayed on top of the water during testing. Additional weights were added to the front of the robot to balance the torque on the robot. After the stationary robot was floating stably in the water tank, it was actuated using the controller. The motion and trajectory generated were observed and changes in the motion of the robot in a water body compared to mid-air actuation were observed. The images from this experiment can be observed in Fig. 10(a)-(f).

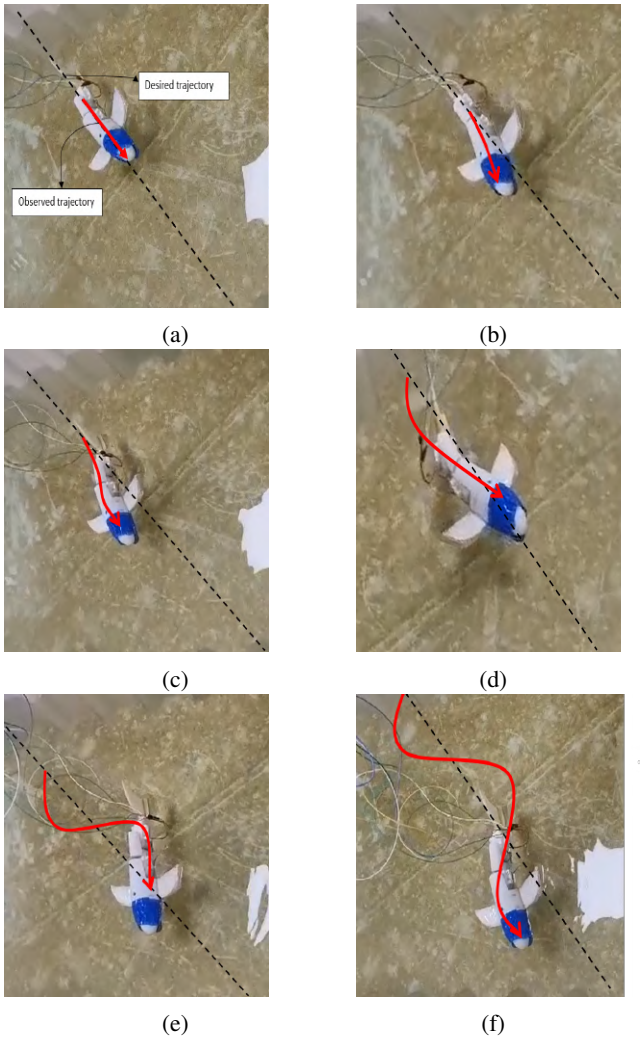


Fig. 10: Desired trajectory v/s observed robot motion with (a) starting point, (b) left turn, (c) right turn, (d) return to center, (e) second left turn, and (f) second right turn.

The simulation and experimental results of the proposed fish robot have shown strong qualitative agreement, demonstrating smooth and coordinated swimming patterns. The robot successfully replicates lifelike sinusoidal motion in physical tests, closely matching the behaviors observed in the simulation. While existing robots like OpenFish [4] and BoxyBot [5] focus on multifunctionality and high-speed swimming, respectively, the current design emphasizes stable motion and precise control. This highlights the effectiveness of the CPG-based control strategy in achieving more coordinated transitions and reliable navigation in constrained aquatic environments.

VI. Conclusion

This work presented the design and development of a biomimetic fish robot capable of demonstrating planar swimming in a still water environment. The robot's design incorporated a hydrodynamic structure with active and passive fins and flexible joints, enabling it to mimic the thunniform locomotion of natural fish. Detailed kinematic and dynamic analyses of the robotic system were conducted, utilizing Lagrangian mechanics to model its motion and interaction with the surrounding water. Furthermore, a CPG has been effectively implemented to govern the rhythmic joint actuation of the robot, allowing for smooth transitions between different swimming behaviors without the need for complex trajectory planning. This CPG-based control has offered a bio-inspired approach to generating the sinusoidal motion observed in fish. Future work would involve adding another degree of freedom and a high-level controller to allow the robot to traverse along the z-axis.

References

[1] R. Wang, S. Wang, Y. Wang, L. Cheng, and M. Tan, "Development and motion control of biomimetic underwater robots: A survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 833–844, 2020.

[2] A. Sahoo, S. K. Dwivedy, and P. Robi, "Advancements in the field of autonomous underwater vehicle," *Ocean Engineering*, vol. 181, pp. 145–160, 2019.

[3] B. M. Patel, J. Narayan, and S. K. Dwivedy, "Manoeuvring planar snake robot in uncertain underwater condition using adaptive neural network sliding mode control," *International Journal of Dynamics and Control*, vol. 12, no. 11, pp. 4138–4156, 2024.

[4] S. C. Van Den Berg, R. B. Scharff, Z. Rusák, and J. Wu, "Openfish: Biomimetic design of a soft robotic fish for high speed locomotion," *HardwareX*, vol. 12, p. e00320, 2022.

[5] D. Lachat, A. Crespi, and A. J. Ijspeert, "Boxybot: a swimming and crawling fish robot controlled by a central pattern generator," IEEE, pp. 643–648, 2006.

[6] J. M. Kumph, "Maneuvering of a robotic pike," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[7] M. Sachinis, "The design and testing of a biologically inspired underwater robotic mechanism," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[8] Y.-H. Lin, R. Siddall, F. Schwab, T. Fukushima, H. Banerjee, Y. Baek, D. Vogt, Y.-L. Park, and A. Jusufi, "Modeling and control of a soft robotic fish with integrated soft sensing," *Advanced intelligent systems*, vol. 5, no. 4, p. 2000244, 2023.

[9] X. Yi, A. Chakarvarthy, and Z. Chen, "Cooperative collision avoidance control of servo/ipmc driven robotic fish with back-relaxation effect," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1816–1823, 2021.

[10] L. Cen and A. Erturk, "Bio-inspired aquatic robotics by untethered piezohydroelastic actuation," *Bioinspiration & biomimetics*, vol. 8, no. 1, p. 016006, 2013.

[11] J. Yu, M. Tan, S. Wang, and E. Chen, "Development of a biomimetic robotic fish and its control algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 4, pp. 1798–1810, 2004.

[12] M. Wang, K. Wang, Q. Zhao, X. Zheng, H. Gao, and J. Yu, "Lqr control and optimization for trajectory tracking of biomimetic robotic fish based on unreal engine," *Biomimetics*, vol. 8, no. 2, p. 236, 2023.

[13] H. Dong, Z. Wu, D. Chen, M. Tan, and J. Yu, "Development of a whale-shark-inspired gliding robotic fish with high maneuverability," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 6, pp. 2824–2834, 2020.

[14] M. L. Castano and X. Tan, "Model predictive control-based path-following for tail-actuated robotic fish," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 7, p. 071012, 2019.

[15] G. Chen, X. Yang, Y. Xu, Y. Lu, and H. Hu, "Neural network-based motion modeling and control of water-actuated soft robotic fish," *Smart Materials and Structures*, vol. 32, no. 1, p. 015004, 2022.

[16] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural networks*, vol. 21, no. 4, pp. 642–653, 2008.

[17] A. Crespi, D. Lachat, A. Pasquier, and A. J. Ijspeert, "Controlling swimming and crawling in a fish robot using a central pattern generator," *Autonomous Robots*, vol. 25, pp. 3–13, 2008.