ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Design Document

Team #39

KE, XUANYU (xuanyuk2)
WANG, LIBIN (libin2)
ZHANG, BOWEN (bz20)
ZHENG, KAIJUN (kaijunz2)

TA: Shao, Xihe

April 1, 2026

1 Introduction

1.1 Problem Statement

Most existing underwater robotic fish rely on a single propulsion method. Bio-inspired tail propulsion is generally smooth, quiet, and energy-efficient, making it well suited for steady swimming and maneuvering. However, it usually provides limited peak thrust and acceleration. In contrast, propeller-based propulsion can generate higher speed and stronger thrust, but it is less effective at reproducing the natural and flexible motion of real fish. Because these two propulsion strategies offer complementary advantages, a robotic platform capable of using both would be more versatile than one relying on only a single mode.

The main challenge is that integrating both propulsion methods into one compact robotic system is difficult. A bio-inspired tail mechanism requires oscillatory motion, flexible structural support, and a control strategy focused on periodic fin movement, while a propeller system requires a rotational drive mechanism, different structural constraints, and a separate mode of actuation. These different requirements make it difficult to combine both functions in a single robotic fish without increasing size or complexity. Therefore, there is a need for a transformable underwater robotic fish that can switch between bionic tail propulsion and propeller propulsion to better adapt to different tasks and environments.

1.2 Solution Overview & Visual Aid

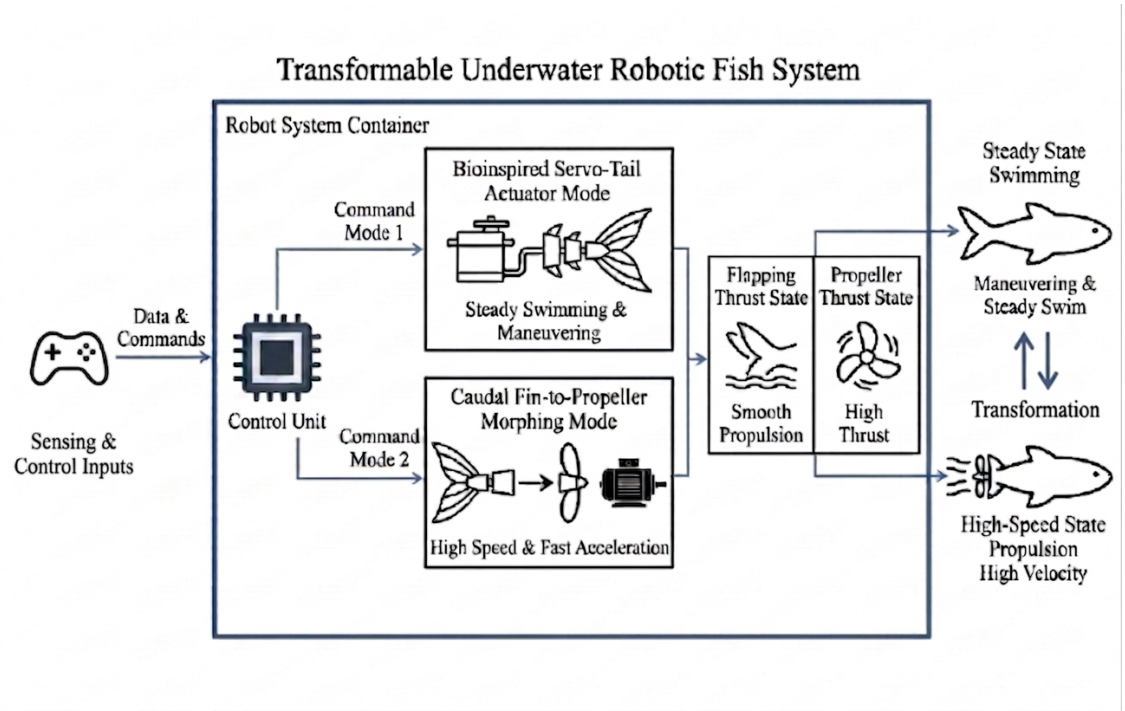

Figure 1: Visual Aid

To address this problem, our project proposes a transformable underwater robotic fish capable of multi-mode propulsion, including a caudal-fin-to-propeller morphing system.

- In bionic mode, a servo-driven tail generates periodic left-right oscillatory motion to produce fish-like swimming behavior. This mode is intended to provide more natural underwater movement.
- In propeller mode, the rear structure transforms the caudal fin into a propeller-based configuration, which is then driven by a brushless motor to produce higher thrust for faster movement.

A manual control system is used to command forward motion, steering, and propulsion mode switching during underwater operation. By combining these two propulsion strategies in a single transformable platform, the robotic fish can serve as a testbed for studying adaptive underwater locomotion and for comparing the performance of different propulsion modes.

Below is an example of a use case for our robotic fish, generated by the Nano Banana Pro model.

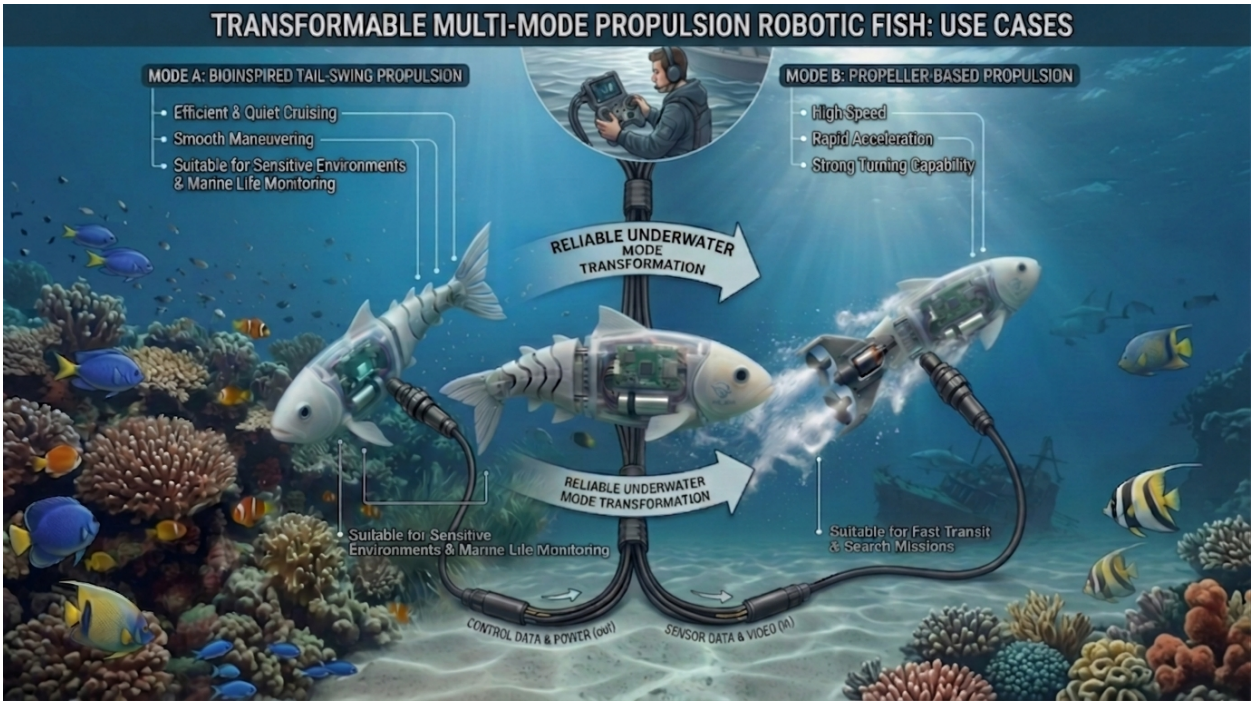

Figure 2: Use Example

1.3 High-Level Requirements List

The transformable underwater robotic fish shall meet the following high-level requirements:

1. The bio-inspired fish must be capable of stable underwater movement in an underwater environment, sustaining controlled forward swimming and directional turns for no less than 30 seconds. Additionally, the bio-inspired fish must be capable of both bio-inspired tail propulsion and propeller propulsion modes, capable of completing a mode switch underwater within 20 seconds while maintaining stability.
2. In bio-inspired mode, the bio-inspired fish should exhibit greater maneuverability. Under identical test conditions, its turning radius should be at least 15% smaller than that of the propeller model.
3. In propeller mode, this bio-inspired fish should demonstrate enhanced high-speed propulsion capabilities. Under identical test conditions, its maximum forward swimming speed should be at least 15% higher than in bio-inspired mode.

2 Design

2.1 Block Diagram

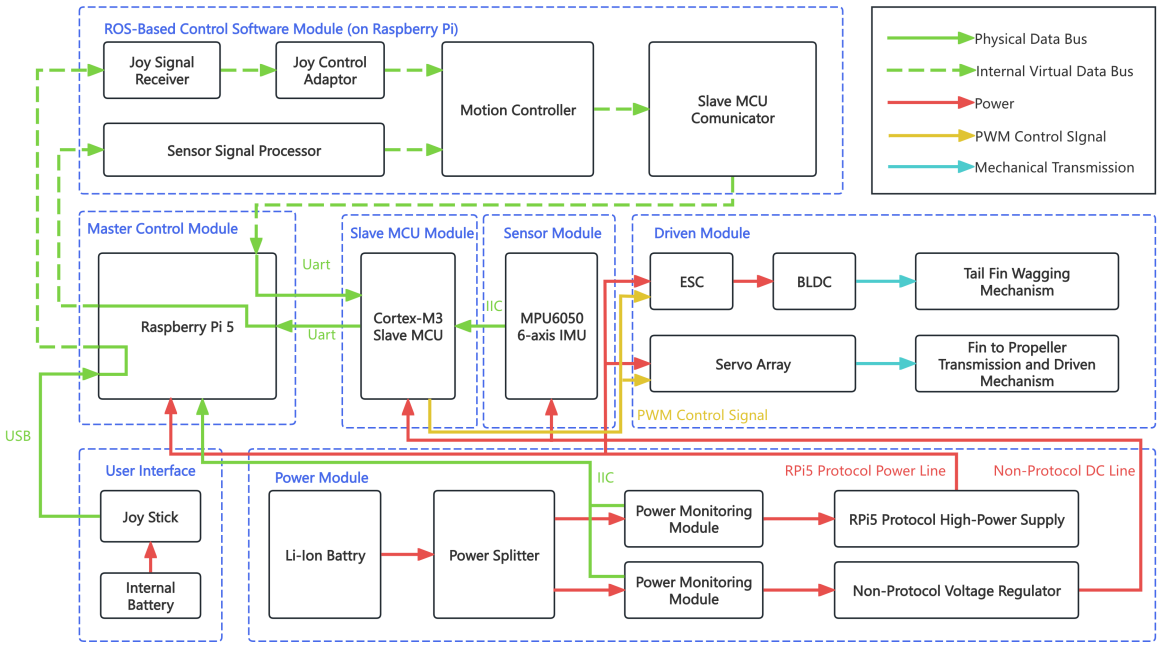

Figure 3: Block Diagram

2.2 ROS-Based Control Software Module

This module serves as the high-level decision-making center, hosted on the Raspberry Pi 5. It integrates user inputs and sensor feedback to generate bionic motion commands. The module comprises four functional units: a Joy Signal receiver and adapter for parsing joy stick controller inputs; a Sensor Signal Processor that fuses IMU data for state estimation; a Motion Controller implementing the Central Pattern Generator (CPG) algorithm for gait synthesis; and a Slave MCU Communicator that encodes and transmits commands via UART.

Table 1: RV Table of ROS-Based Control Software Module

Requirements	**Verification**
1. The module shall parse joystick input and generate motion commands at a minimum update rate of 20 Hz.	Connect a standard gamepad and monitor the ”/joy” topic; verify command frequency using rostopic hz.
2. The module shall switch between bionic and propeller modes within 20 seconds of user command.	Press the mode-switch button; measure time from input reception to UART command transmission via logged timestamps.
3. The UART communication with the slave MCU shall operate at 11,5200 baud with a defined packet format including checksum.	Capture UART packets using a logic analyzer; verify correct framing and checksum handling for valid and invalid packets.

2.2.1 Supporting Material

Figure 4: ROS Node Graph

```
root@eab87308bbd0:~/ros2_ws# ros2 launch pwm_servo_driver sim_demo.launch.py
[INFO] [launch]: All log files can be found below /root/.ros/log/2026-03-29-07-17-57-402793-eab87308bbd0-45
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [pwm_servo_driver_node-1]: process started with pid [50]
[INFO] [servo_test_publisher-2]: process started with pid [51]
[pwm_servo_driver_node-1] [INFO] [1774768677.630399257] [pwm_servo_driver]: === SIMULATION MODE === (no serial hardware)
[pwm_servo_driver_node-1] [INFO] [1774768677.632757799] [pwm_servo_driver]: PWM servo driver node started
[servo_test_publisher-2] [INFO] [1774768677.637093965] [servo_test_publisher]: Test publisher: servos=[1], period=2.0s
[servo_test_publisher-2] [INFO] [1774768677.637244340] [servo_test_publisher]: Will toggle between pulse 1100 and 1900
[servo_test_publisher-2] [INFO] [1774768679.637192300] [servo_test_publisher]: Publishing: pulse_width=1900
[pwm_servo_driver_node-1] [INFO] [1774768679.639253508] [pwm_servo_driver]: -> Set position [servo1:1900us] duration=1.6
0s
[pwm_servo_driver_node-1] [INFO] [1774768679.639658216] [pwm_servo_driver]: [SIM] Would send 12 bytes: AA 55 04 07 01 40
 06 01 01 6C 07 16
[servo_test_publisher-2] [INFO] [1774768681.639772009] [servo_test_publisher]: Publishing: pulse_width=1100
[pwm_servo_driver_node-1] [INFO] [1774768681.641974301] [pwm_servo_driver]: -> Set position [servo1:1100us] duration=1.6
0s
[pwm_servo_driver_node-1] [INFO] [1774768681.642414009] [pwm_servo_driver]: [SIM] Would send 12 bytes: AA 55 04 07 01 40
 06 01 01 4C 04 35
[servo_test_publisher-2] [INFO] [1774768683.637279802] [servo_test_publisher]: Publishing: pulse_width=1900
[pwm_servo_driver_node-1] [INFO] [1774768683.638986093] [pwm_servo_driver]: -> Set position [servo1:1900us] duration=1.6
0s
[pwm_servo_driver_node-1] [INFO] [1774768683.639427010] [pwm_servo_driver]: [SIM] Would send 12 bytes: AA 55 04 07 01 40
 06 01 01 6C 07 16
[servo_test_publisher-2] [INFO] [1774768685.639381053] [servo_test_publisher]: Publishing: pulse_width=1100
[pwm_servo_driver_node-1] [INFO] [1774768685.641141886] [pwm_servo_driver]: -> Set position [servo1:1100us] duration=1.6
0s
[pwm_servo_driver_node-1] [INFO] [1774768685.641496761] [pwm_servo_driver]: [SIM] Would send 12 bytes: AA 55 04 07 01 40
 06 01 01 4C 04 35
```

Figure 5: Terminal Screenshot of ROS Software Controlling a Servo

2.3 Master Control Module

Designed around the Raspberry Pi 5, this module provides the hardware platform for ROS execution and system coordination. It manages high-level data processing and inter-module communication. The module interfaces with the Slave MCU via UART for real-time command dispatch and receives regulated power from the Power Module through a dedicated USB Type-C interface. The module receives power monitoring information from the Power Module via IIC for energy efficiency comparison between two propulsion mode.

Table 2: RV Table of Master Control Module

Requirements	**Verification**
1. The Raspberry Pi 5 shall successfully boot the ROS environment and initialize all required nodes within 30 seconds after power-on.	Power on the system and record boot time; verify all ROS nodes are active using rosnode list.
2. The module shall exchange commands with the slave MCU via UART at 11,5200 baud with a packet loss rate below 1%.	Transmit 1000 test packets; calculate packet loss rate using a serial monitor.

2.3.1 Supporting Material

Please refer to Fig. 6 for information about our Master Control Module Device and ROS Environment. We set up the Ubuntu 24.04 Linux distribution and installed the Jazzy version of the ROS 2 environment.

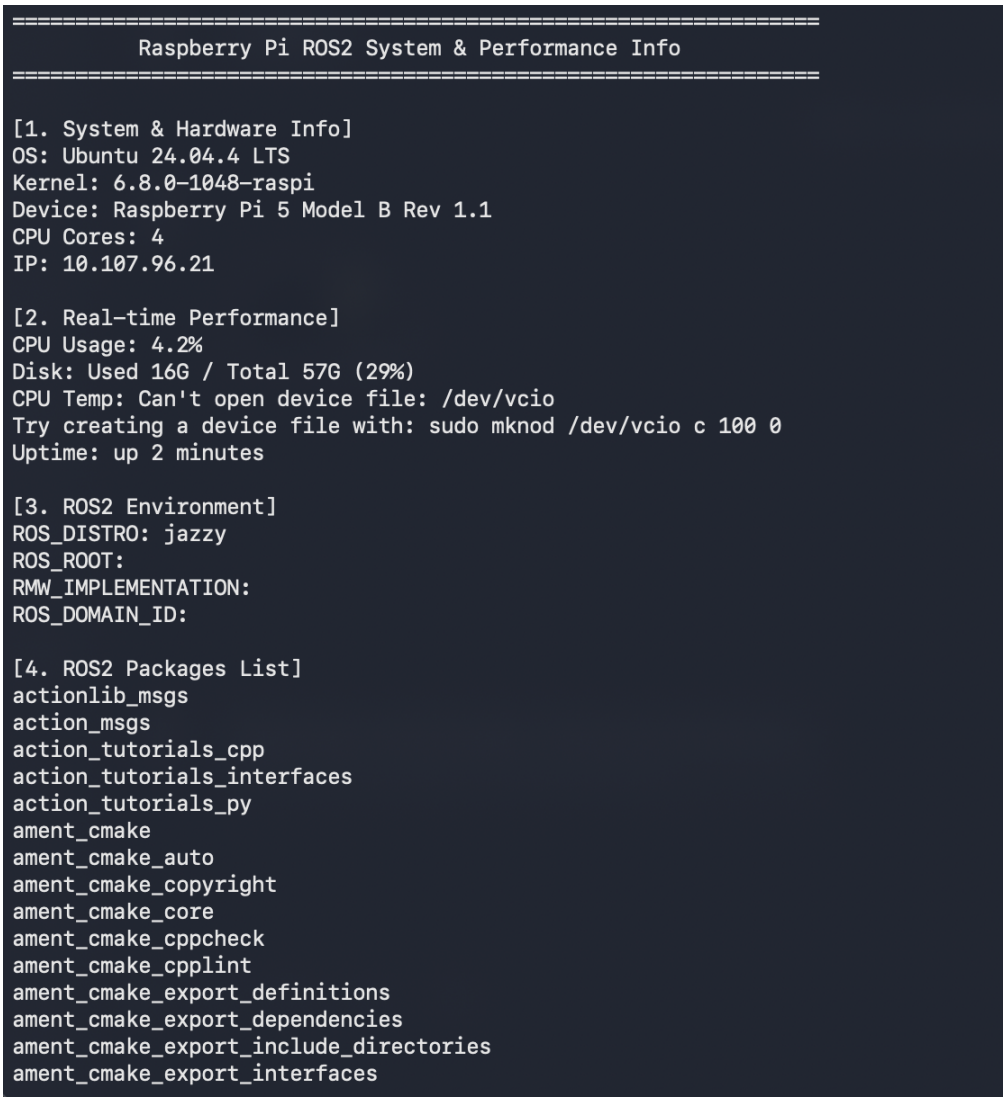

Figure 6: Master Control Module Device and ROS Environment Info

2.4 Slave MCU Module

This module employs a Cortex-M3 microcontroller as a real-time executive controller. Its primary functions are:

1. Decoding high-level commands from the Master Control Module;
2. Generating precision PWM signals for actuation;
3. Acquiring raw sensor data from the MPU6050 via I^2C.

To support these functions, the Cortex-M3 maintains bidirectional UART communication with the Raspberry Pi 5 for command reception and IMU data telemetry, while relying on the Power Module's Non-Protocol supply for voltage input.

Table 3: RV Table of Slave MCU Module

Requirements	**Verification**
1. The slave MCU shall decode UART commands from the master module and generate corresponding PWM signals within 10 ms of command reception.	Send test commands via UART; measure time from command arrival to PWM output change using a logic analyzer.
2. The module shall acquire IMU data (acceleration and angular velocity) from the MPU6050 via I^2C at a minimum sampling rate of 20 Hz.	Log I^2C readouts; verify sampling frequency using timestamps; check that raw data values respond correctly to sensor motion.

2.4.1 Supporting Material

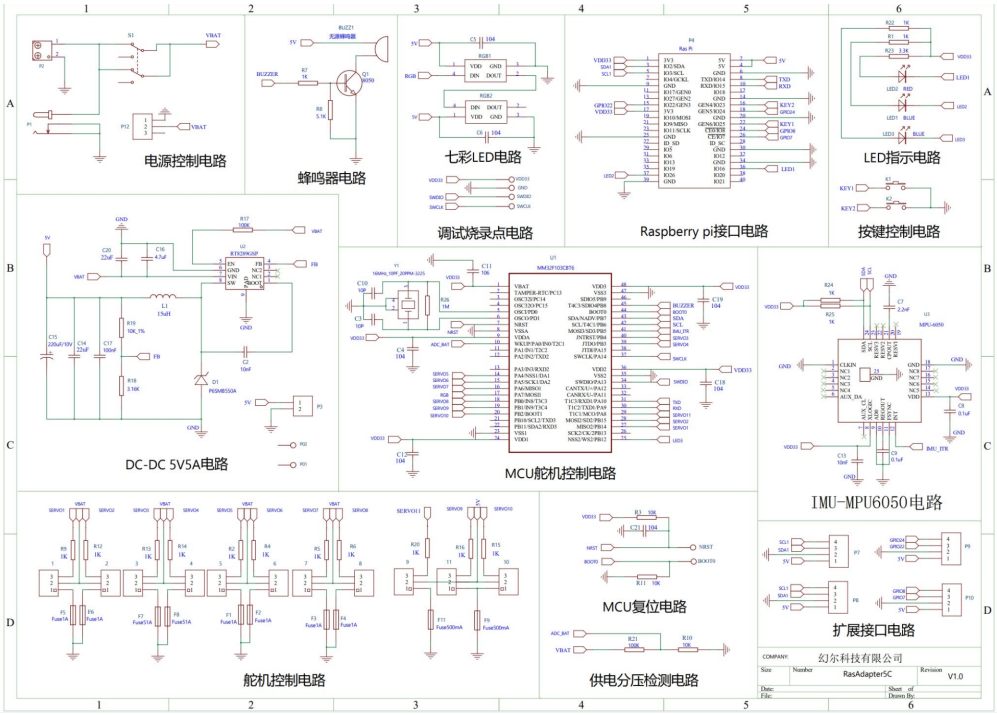

Figure 7: Schematics Diagram of Slave MCU Module

2.5 Sensor Module

This module incorporates an MPU6050 6-axis IMU to provide inertial measurement for closed-loop control. It delivers 3-axis acceleration and 3-axis angular velocity data to the Slave MCU via the I^2C bus, enabling real-time attitude estimation and motion feedback critical for underwater stability, while powered by the Non-Protocol supply from the Power Module.

Table 4: RV Table of Sensor Module

Requirements	**Verification**
1. The MPU6050 shall provide 3-axis acceleration and 3-axis angular velocity data to the slave MCU via I^2C at a minimum sampling rate of 20 Hz.	Log I^2C readouts; verify data rate using timestamps; confirm all six axes produce non-zero responses when the sensor is moved.
2. The module shall operate correctly after integration into the sealed waterproof core compartment, with no data corruption caused by other onboard electronics.	Run the sensor continuously for 5 minutes inside the sealed compartment with all other modules powered; verify no unexpected data spikes or I^2C communication errors.

2.5.1 Supporting Material

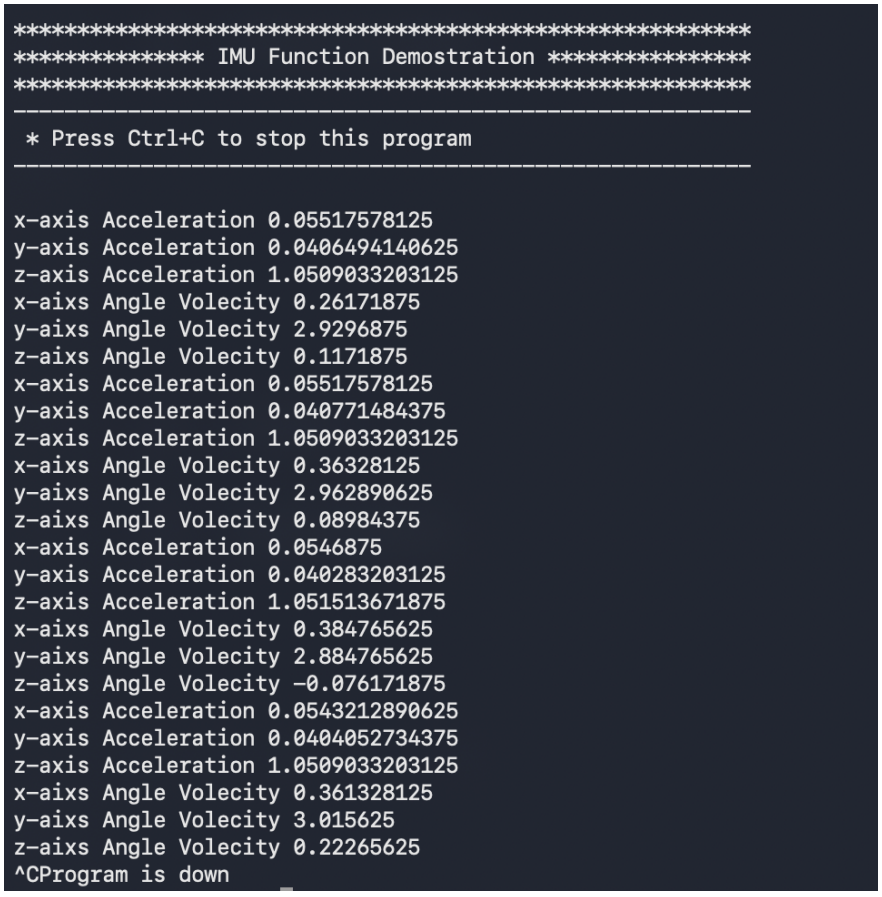

Figure 8: Terminal Screenshot of IMU Demonstration

2.6 Driven Module

This module constitutes the electromechanical actuation layer, comprising two distinct subsystems:

1. A BLDC motor(Brushless Direct Current Motor) paired with an ESC(Electronic Speed Controller) for propeller-mode rotation;
2. A Servo Array for tail oscillation.

Both actuators receive synchronized PWM control signals from the Slave MCU Module and receive the Non-Protocol supply from the Power Module. They generate mechanical driving force and torque to enable the robotic fish to oscillate its fins and rotate the propeller via a specially designed mechanical structure.

Table 5: RV Table of Driven Module

Requirements	**Verification**
1. The BLDC motor and ESC shall respond to PWM control signals and generate sufficient thrust to achieve at least 15% higher forward speed in propeller mode compared to bionic mode.	Run the robotic fish in a test tank; measure maximum forward speed using video tracking or an optical flow sensor under identical test conditions.
2. The servo array shall produce periodic tail oscillations in bionic mode, enabling a turning radius at least 15% smaller than in propeller mode.	Perform turning radius tests in a controlled pool; record trajectory and compare minimum turning radii between the two modes.
3. Both actuators shall operate reliably during underwater mode switching, with no mechanical stalling or signal loss within the 20-second transition window.	Trigger mode switching underwater; visually inspect tail-to-propeller transformation; monitor PWM signals with a logic analyzer throughout the switching period.

2.6.1 Supporting Material

Please refer to Fig. 12 and Fig. 13 in Appendix A. They are the modeling diagrams of our two types of propulsion modes. By altering the structure of the tail fin, the two propulsion modes can be switched between.

2.7 Power Module

This module manages energy storage, distribution, and monitoring. A Li-Ion Battery serves as the primary source, with power conditioned through a Power Splitter and dedicated Voltage Regulators. The module provides high-current RPi5 Protocol Power for the

Master Control Module and low-noise regulated voltage supplies for other devices, with power monitoring telemetry relayed to the Master Control Module via I^2C.

Table 6: RV Table of Power Module

Requirements	**Verification**
1. The module shall supply stable power to all subsystems for continuous underwater operation of at least 30 minutes.	Run the robotic fish in a test tank for 30 minutes; monitor voltage levels at each subsystem input; verify no unexpected power reset occurs.
2. The module shall provide RPi5 Protocol Power to the Master Control Module and low-noise regulated voltages to sensors and actuators, with ripple below 50 mV peak-to-peak for analog-sensitive components.	Measure output rails using an oscilloscope; verify ripple specifications under both idle and full-load conditions.
3. The module shall report voltage and current consumption to the Master Control Module via I^2C, enabling energy efficiency comparison between bionic and propeller modes.	Log I^2C telemetry during separate bionic and propeller mode runs; compare power draw and calculate energy consumption per operation cycle.

2.7.1 Supporting Material

Fig. 9 below illustrates the architecture of our power module. The system is built around a Raspberry Pi 5 board, combined with an expansion board and an RP5 protocol high-power energy supply board. The power is delivered through a USB Type-C interface. This stacked and modular design provides stable power support, improves integration efficiency, and offers a compact hardware solution for the overall robotic fish.

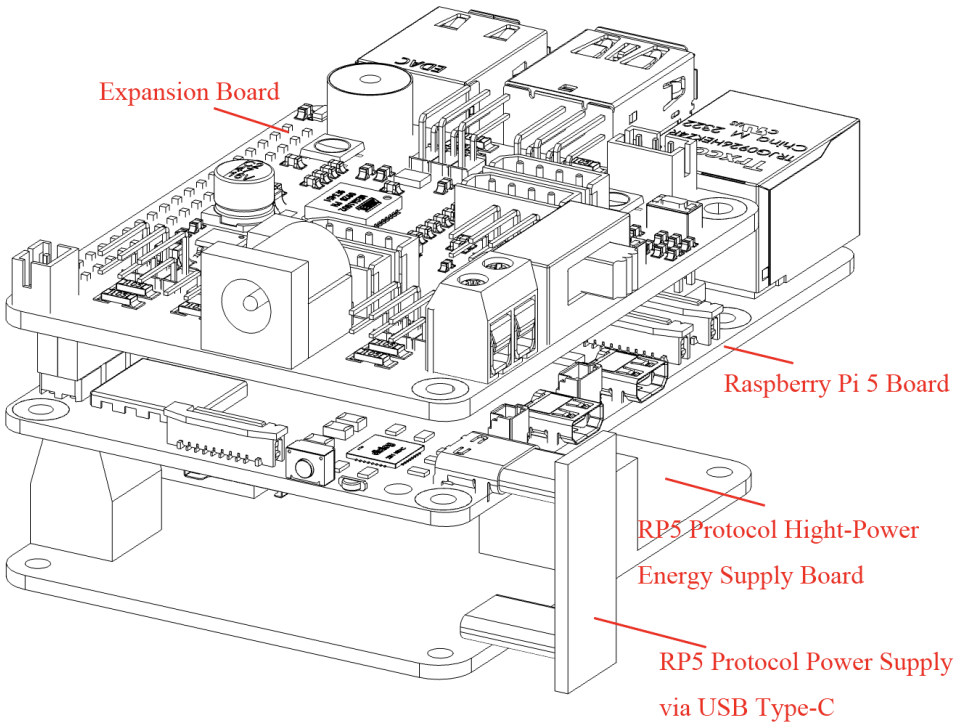

Figure 9: Raspberry Pi Protocol Power Module

2.8 User Interface Module

This module enables human-in-the-loop control through a Joy Stick with an Internal Battery. It transmits raw control inputs to the Master Control Module via USB, facilitating real-time wired teleoperation.

Table 7: RV Table of User Interface Module

Requirements	**Verification**
The joystick shall transmit raw control inputs to the Master Control Module via USB with a latency of less than 50 ms.	Connect the joystick and run a loopback test; measure time between physical input actuation and command reception using logged timestamps.

2.8.1 Supporting Material

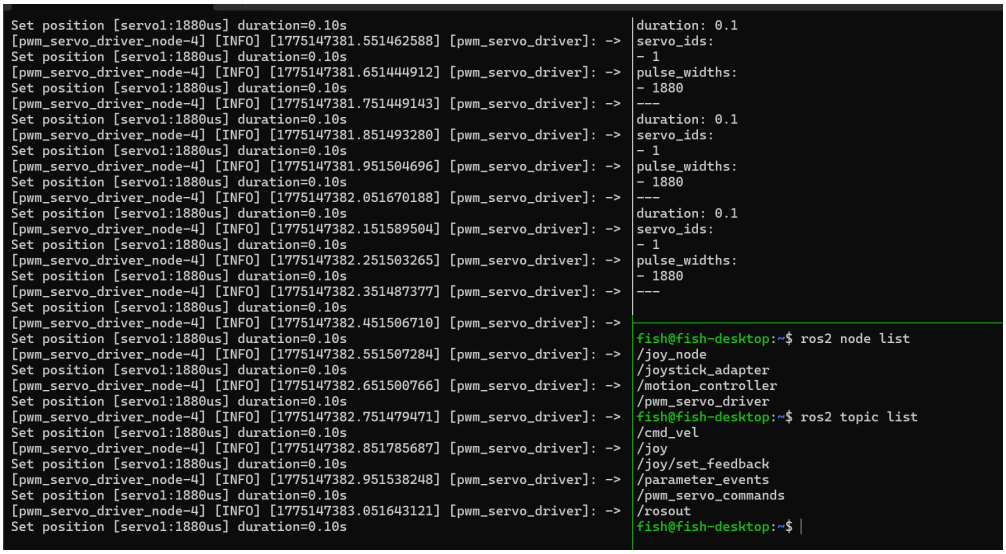

Figure 10: Terminal Screenshot of Joy Stick Control

2.9 Tolerance Analysis

A critical risk in the proposed multimodal robotic fish is whether the tail fin can be reliably morphed into a propeller underwater using the selected brushless motor and custom bevel-gear drivetrain. During this transition, the motor must overcome the hydrodynamic drag acting on the fin panels while also accelerating the fin structure through the gear transmission. The purpose of this section is therefore to verify that the fin-to-propeller transformation is mechanically feasible, that the available motor torque is sufficient to initiate the motion in water, and that the opening motion remains fast, monotonic, and bounded throughout deployment.

2.9.1 Modeling Assumptions

To obtain a tractable but conservative first-order model, the following assumptions are adopted.

First, the surrounding fluid is modeled as quiescent water in an effectively unbounded domain. This removes wall and free-surface effects and isolates the hydrodynamic resistance associated with the morphing motion itself.

Second, each fin panel is modeled as a zero-thickness plate during the unfolding process. Although the real tail fin has finite thickness and geometric detail, the dominant resistance during opening is the pressure drag induced by motion normal to the fin surface.

Third, the true fin outline is conservatively bounded by a rectangular plate of width b and radial span from r_1 to r_2. The rectangle is selected such that it fully encloses the projected fin shape during the morphing motion. As a result, the predicted hydrodynamic torque

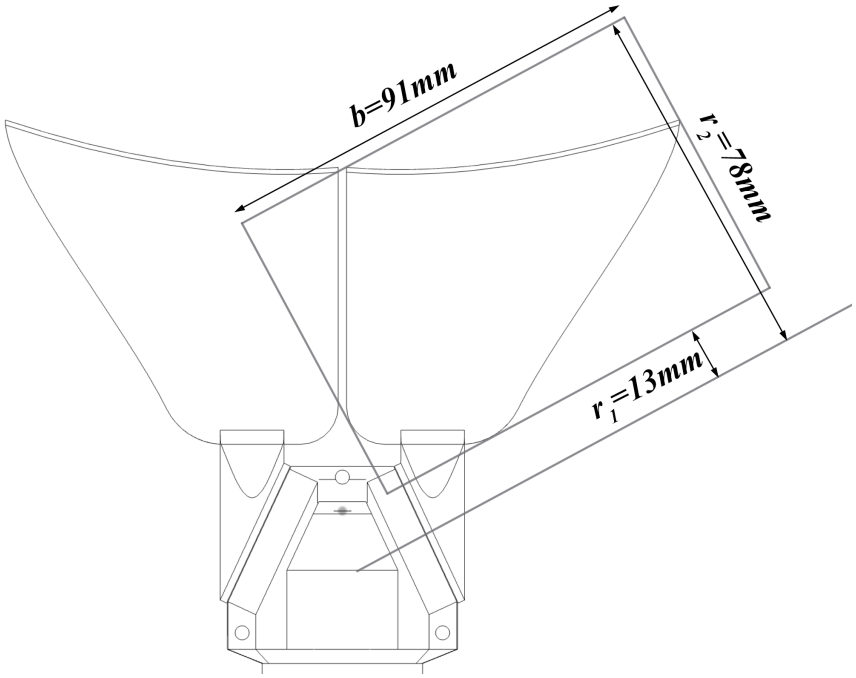

Figure 11: Schematic of the actual tail-fin outline and the circumscribed rectangular plate used in the tolerance analysis. The rectangle fully encloses the projected fin shape during morphing, yielding a conservative estimate of hydrodynamic drag torque. The geometric parameters b, r_1, and r_2 are defined in the figure.

is intentionally biased high, providing an upper-bound estimate of the required actuation torque.

Fourth, the rotation axis is assumed to lie in the plane of each fin panel. Therefore, the local relative flow is taken to be normal to the panel surface, and the local speed depends only on the distance from the rotation axis.

Fifth, the two fin panels are assumed to be geometrically symmetric and to open synchronously under the bevel-gear transmission. The morphing mechanism can therefore be described by a single generalized coordinate θ, which denotes the common opening angle of the two panels.

Sixth, the inertia of compact and irregular components located very close to the output rotation axis, including the hub and small output-side hardware, is neglected in the first-order model. This approximation is justified because their characteristic radius is much smaller than that of the fin panels, so their contribution to the total reflected inertia is expected to be small relative to the fin inertia and the reflected motor inertia.

Finally, drivetrain and parasitic losses are lumped into a conservative torque-delivery coefficient $\eta_d = 0.9$. This coefficient is used to account for bevel-gear meshing loss, bearing friction, alignment error, and other parasitic mechanical losses in a single conservative parameter. Since the bevel-gear transmission is custom-designed, the kinematic gear ra-

tio is obtained directly from the tooth counts, while the choice $\eta_d = 0.9$ intentionally underestimates the actually delivered output torque to preserve design margin.

2.9.2 Hydrodynamic Torque on a Single Fin Panel

Let $\omega = \dot{\theta}$ denote the angular speed of one fin panel during opening. At radius r, the local speed is

$$U(r) = \omega r. \tag{1}$$

For a differential strip of the bounding rectangle shown in Fig. 11, the differential area is

$$dA = b\, dr. \tag{2}$$

The local Reynolds number is written as

$$Re(r) = \frac{U(r)b}{\nu} = \frac{\omega r b}{\nu}, \tag{3}$$

where ν is the kinematic viscosity of water. In the most general form, the drag coefficient may therefore be written as

$$C_D = C_D(Re(r))\,. \tag{4}$$

The differential drag force acting on the strip is

$$dF_D = \frac{1}{2}\rho C_D(Re(r))\, U(r)^2\, dA = \frac{1}{2}\rho C_D(Re(r))\, (\omega r)^2 b\, dr, \tag{5}$$

where ρ is the water density. The corresponding differential torque about the panel rotation axis is

$$d\tau_{h,1} = r\, dF_D = \frac{1}{2}\rho b\, \omega^2 C_D(Re(r))\, r^3\, dr. \tag{6}$$

Integrating over the full panel gives the hydrodynamic resisting torque on a single fin panel:

$$\tau_{h,1}(\omega) = \frac{1}{2}\rho b\, \omega^2 \int_{r_1}^{r_2} C_D\left(\frac{\omega r b}{\nu}\right) r^3\, dr. \tag{7}$$

Equation (7) is the most general form used in this analysis. However, for a thin flat plate normal to the flow, standard references indicate that the drag coefficient can be approximated as nearly constant over a broad range of separated-flow conditions, and a representative value of $C_D = 1.28$ is commonly used for a flat plate with frontal area as the reference area **princeton˙flatplate**, [1]. Therefore, the first-pass tolerance analysis adopts

$$C_D \approx 1.28. \tag{8}$$

Substituting this approximation into Eq. (7) gives

$$\tau_{h,1}(\omega) = \frac{1}{2}\rho C_D b\, \omega^2 \int_{r_1}^{r_2} r^3\, dr = \frac{\rho C_D b}{8}\left(r_2^4 - r_1^4\right)\omega^2. \tag{9}$$

Defining

$$k_q = \frac{\rho C_D b}{8}\left(r_2^4 - r_1^4\right), \tag{10}$$

the hydrodynamic torque on a single fin panel can be written compactly as

$$\tau_{h,1}(\omega) = k_q \omega^2. \tag{11}$$

Using

$$\rho = 1.0\,\mathrm{g/cm^3} = 1000\,\mathrm{kg/m^3}, \quad C_D = 1.28, \quad b = 91\,\mathrm{mm} = 0.091\,\mathrm{m}, \quad r_1 = 13\,\mathrm{mm} = 0.013\,\mathrm{m}, \quad r_2 = 78\,\mathrm{mm} \tag{12}$$

Eq. (10) becomes

$$k_q = \frac{(1000)(1.28)(0.091)}{8}\left(0.078^4 - 0.013^4\right) = 5.39 \times 10^{-4}\,\mathrm{N\,m\,s^2}. \tag{13}$$

2.9.3 Total System Model for Two Synchronous Fin Panels

Because the two fin panels are assumed to open synchronously and symmetrically, the total hydrodynamic torque acting on the mechanism is the sum of the torques on the two panels:

$$\tau_{h,\mathrm{tot}}(\omega) = 2\tau_{h,1}(\omega) = 2k_q\omega^2. \tag{14}$$

For compactness, define

$$k_{q,\mathrm{tot}} = 2k_q, \tag{15}$$

so that

$$\tau_{h,\mathrm{tot}}(\omega) = k_{q,\mathrm{tot}}\omega^2. \tag{16}$$

Using Eq. (13),

$$k_{q,\mathrm{tot}} = 2k_q = 1.08 \times 10^{-3}\,\mathrm{N\,m\,s^2}. \tag{17}$$

Let τ_m denote the motor-side torque. The bevel-gear ratio is defined as

$$i = \frac{\omega_m}{\omega_o} = \frac{N_o}{N_m} = \frac{7}{8}, \tag{18}$$

where ω_m is the motor speed, ω_o is the output-panel speed, and N_m and N_o are the tooth counts of the motor-side and output-side bevel gears, respectively.

The available output torque is therefore modeled as

$$\tau_{\mathrm{drive}} = \eta_d\, i\, \tau_m. \tag{19}$$

The full equivalent inertia about the common output coordinate θ may be written as

$$J_{\mathrm{eq}} = 2J_{\mathrm{fin}} + 2J_{\mathrm{hub}} + J_{\mathrm{gear,out}} + i^2 J_m + J_{\mathrm{misc}}, \tag{20}$$

where J_{fin} is the inertia of one fin panel, J_{hub} is the inertia of the hub or nearby output-side hardware associated with one panel, $J_{\text{gear,out}}$ is the inertia of the output-side transmission elements, J_m is the motor rotor inertia, and J_{misc} denotes any additional reflected inertias.

Under the sixth modeling assumption, the near-axis terms are neglected in the first-order model, so Eq. (20) reduces to

$$J_{\text{eq}} \approx 2J_{\text{fin}} + i^2 J_m. \tag{21}$$

Using

$$J_{\text{fin}} = 3.0 \times 10^{-5}\,\text{kg m}^2, \qquad J_m = 1.5 \times 10^{-6}\,\text{kg m}^2, \qquad i = \frac{7}{8}, \tag{22}$$

Eq. (21) gives

$$J_{\text{eq}} \approx 2(3.0 \times 10^{-5}) + \left(\frac{7}{8}\right)^2 (1.5 \times 10^{-6}) = 6.11 \times 10^{-5}\,\text{kg m}^2. \tag{23}$$

Since all drivetrain and parasitic losses have already been absorbed into $\eta_d = 0.9$, no separate τ_{other} term is retained in the first-order model. The rotational dynamics of the synchronized two-panel morphing mechanism are therefore written as

$$J_{\text{eq}}\ddot{\theta} = \eta_d i \tau_m - k_{q,\text{tot}}\dot{\theta}^2. \tag{24}$$

2.9.4 Startup Feasibility

At the instant of startup, $\dot{\theta} = 0$, so the hydrodynamic term vanishes. Equation (24) then gives the initial angular acceleration as

$$\alpha_0 = \ddot{\theta}\Big|_{\dot{\theta}=0} = \frac{\eta_d i \tau_{m,\max}}{J_{\text{eq}}}. \tag{25}$$

Using

$$\eta_d = 0.9, \qquad i = \frac{7}{8}, \qquad \tau_{m,\max} = 0.264\,\text{N m}, \qquad J_{\text{eq}} = 6.11 \times 10^{-5}\,\text{kg m}^2, \tag{26}$$

the available output torque at startup is

$$\tau_{\text{drive,max}} = \eta_d i \tau_{m,\max} = 0.9\left(\frac{7}{8}\right)(0.264) = 0.2079\,\text{N m}. \tag{27}$$

Substituting Eq. (27) into Eq. (25) yields

$$\alpha_0 = \frac{0.2079}{6.11 \times 10^{-5}} = 3.40 \times 10^3\,\text{rad/s}^2. \tag{28}$$

Because $\alpha_0 > 0$ and is large in magnitude, the selected motor is able to initiate the underwater fin-to-propeller morphing motion from rest. In other words, the drivetrain still provides a strong positive startup torque even after the conservative loss factor $\eta_d = 0.9$ is applied.

2.9.5 Opening-Speed Characteristics

For a first-pass estimate of the transient response over the short deployment stroke, the available motor torque is approximated as constant and equal to the same maximum available value used above, i.e.,

$$\tau_m \approx \tau_{m,\max} = 0.264\,\mathrm{N\,m}. \tag{29}$$

Under this approximation, define the net constant driving torque

$$A = \eta_d i \tau_m. \tag{30}$$

Using Eq. (27),

$$A = 0.2079\,\mathrm{N\,m}. \tag{31}$$

Equation (24) then becomes

$$J_{\mathrm{eq}}\ddot{\theta} = A - k_{q,\mathrm{tot}}\dot{\theta}^2. \tag{32}$$

The corresponding equilibrium angular speed, obtained by setting $\ddot{\theta} = 0$, is

$$\omega_\infty = \sqrt{\frac{A}{k_{q,\mathrm{tot}}}}. \tag{33}$$

Substituting Eqs. (31) and (17) gives

$$\omega_\infty = \sqrt{\frac{0.2079}{1.08 \times 10^{-3}}} = 13.9\,\mathrm{rad/s} \approx 133\,\mathrm{rpm}. \tag{34}$$

This means that, under the constant-torque approximation, the opening speed rises from zero and remains bounded above by approximately $13.9\,\mathrm{rad/s}$.

Equation (32) may also be written directly in terms of angular speed:

$$J_{\mathrm{eq}}\dot{\omega} = A - k_{q,\mathrm{tot}}\omega^2. \tag{35}$$

This is a first-order nonlinear ordinary differential equation in $\omega(t)$, and its closed-form solution is

$$\omega(t) = \omega_\infty \tanh\left(\frac{\sqrt{A k_{q,\mathrm{tot}}}}{J_{\mathrm{eq}}}\, t\right). \tag{36}$$

Using the numerical values above,

$$\frac{\sqrt{A k_{q,\mathrm{tot}}}}{J_{\mathrm{eq}}} = \frac{\sqrt{(0.2079)(1.08 \times 10^{-3})}}{6.11 \times 10^{-5}} = 244.7\,\mathrm{s}^{-1}, \tag{37}$$

so Eq. (36) becomes

$$\omega(t) = 13.9\,\tanh(244.7\,t) \quad \mathrm{rad/s}. \tag{38}$$

Equation (38) shows that the opening speed increases monotonically from zero and rapidly approaches its finite upper bound. Thus, the deployment response is well behaved and does not exhibit unbounded acceleration or velocity growth.

A corresponding angle-domain expression may be obtained by rewriting Eq. (32) with

$$\ddot{\theta} = \omega \frac{d\omega}{d\theta} = \frac{1}{2} \frac{d(\omega^2)}{d\theta}. \tag{39}$$

This gives

$$\frac{J_{\text{eq}}}{2} \frac{d(\omega^2)}{d\theta} + k_{q,\text{tot}} \omega^2 = A, \tag{40}$$

which is a first-order linear ordinary differential equation in $y(\theta) = \omega^2(\theta)$. Therefore, its solution takes an exponential form:

$$\omega^2(\theta) = \frac{A}{k_{q,\text{tot}}} \left[1 - \exp\left(-\frac{2k_{q,\text{tot}}}{J_{\text{eq}}} (\theta - \theta_0) \right) \right]. \tag{41}$$

Using the numerical values already obtained,

$$\frac{A}{k_{q,\text{tot}}} = 193.0 \quad (\text{rad/s})^2, \qquad \frac{2k_{q,\text{tot}}}{J_{\text{eq}}} = 35.23\,\text{rad}^{-1}, \tag{42}$$

so Eq. (41) becomes

$$\omega^2(\theta) = 193.0 \left[1 - \exp(-35.23(\theta - \theta_0)) \right], \tag{43}$$

with angles expressed in radians.

Integrating Eq. (36) gives the opening angle as a function of time:

$$\theta(t) - \theta_0 = \frac{J_{\text{eq}}}{k_{q,\text{tot}}} \ln \left[\cosh\left(\frac{\sqrt{A k_{q,\text{tot}}}}{J_{\text{eq}}}\, t \right) \right]. \tag{44}$$

Using the numerical values above,

$$\frac{J_{\text{eq}}}{k_{q,\text{tot}}} = \frac{6.11 \times 10^{-5}}{1.08 \times 10^{-3}} = 5.68 \times 10^{-2}, \tag{45}$$

so Eq. (44) becomes

$$\theta(t) - \theta_0 = 5.68 \times 10^{-2} \ln[\cosh(244.7\, t)]. \tag{46}$$

For the fixed deployment stroke prescribed by the mechanism geometry,

$$\Delta\theta = \theta_f - \theta_0, \tag{47}$$

the corresponding opening time is obtained by inverting Eq. (46):

$$t_f = \frac{J_{\text{eq}}}{\sqrt{A k_{q,\text{tot}}}} \operatorname{arcosh}\left[\exp\left(\frac{k_{q,\text{tot}}}{J_{\text{eq}}} \Delta\theta \right) \right]. \tag{48}$$

Numerically,

$$t_f = 4.09 \times 10^{-3} \operatorname{arcosh}[\exp(17.61\,\Delta\theta)] \quad \text{s}, \tag{49}$$

where $\Delta\theta$ is expressed in radians.

Equation (49) shows that the opening time is finite for any physically meaningful deployment stroke $\Delta\theta > 0$, while Eqs. (38) and (43) show that the opening speed grows monotonically and remains bounded. Therefore, the opening process is dynamically feasible and sufficiently well behaved for a mode-switching mechanism.

2.9.6 Interpretation and Design Conclusion

This tolerance analysis is conservative in two primary ways. First, the actual fin geometry is replaced by a circumscribed rectangular plate that fully encloses the real projected fin shape, thereby overestimating hydrodynamic drag. Second, the custom bevel-gear transmission and parasitic losses are absorbed into a conservative torque-delivery coefficient $\eta_d = 0.9$, which reduces the modeled torque actually available to the fin-opening mechanism.

At the same time, the inertia of compact near-axis hardware is neglected in the first-order model. This slightly offsets the conservatism introduced above, but its contribution is expected to be second-order relative to the fin inertia and the reflected motor inertia because these components lie close to the rotation axis.

With the present geometry and the measured motor parameters, the analysis gives

$$k_q = 5.39 \times 10^{-4}\,\text{N m s}^2, \qquad k_{q,\text{tot}} = 1.08 \times 10^{-3}\,\text{N m s}^2, \tag{50}$$

$$J_{\text{eq}} = 6.11 \times 10^{-5}\,\text{kg m}^2, \qquad \tau_{\text{drive,max}} = 0.2079\,\text{N m}, \tag{51}$$

$$\alpha_0 = 3.40 \times 10^{3}\,\text{rad/s}^2, \qquad \omega_\infty = 13.9\,\text{rad/s} \approx 133\,\text{rpm}. \tag{52}$$

These results show that the selected motor can initiate the underwater fin-to-propeller morphing motion from rest and that the opening-speed response is monotonic, bounded, and analytically predictable. Therefore, from the standpoint of underwater actuation torque and deployment dynamics, the proposed morphing tail-fin subsystem is feasible for the ECE 445 robotic fish design.

3 Cost

Table 8: Cost Table

Object	Number	Cost (RMB)	Bought/Planned
LCR Bridge Probes	1	120.9	Bought
Elbow-style wire stripper	1	19.8	Bought
Game controller	1	84.3	Bought
Raspberry Pi 5 Active Cooler	1	38.9	Bought
4x Adhesive-Lined Double-Wall Heat Shrink Tubing (4mm, 6mm)	2	9.24	Bought
Waterproof servo	4	616	Bought
Liquid Insulating Tape	1	26.9	Bought
Standard Heat Shrink Tubing (2mm, 3mm, 4mm, 5mm, 8mm, 12mm, 25mm)	14	72.66	Bought
Raspberry Pi 5 Development Board	1	972	Bought
Tube-type Wire Terminal	1	57.8	Bought
64GB Memory Card	1	139	Bought
Raspberry Pi 5 Expansion Board	1	299	Bought
Raspberry Pi 5 Power Expansion Board	1	79	Bought
Brushless Direct Current Motor	1	300	Planned
Electronic Speed Controller	1	50	Planned
Waterproof Circular Connector	6	35	Planned
3D Printed Components	1	300	Planned
Custom PCB and Components	1	120	Planned

Currently Total: 2535.5

4 Schedule

Table 9: Weekly Schedule Table

Week	**Bowen Zhang**	**Xuanyu Ke**	**Kaijun Zheng**	**Libin Wang**
03/23	Initial fish concept and waterproof enclosure design	Initial CAD of fish body and morphing structure	Define PCB requirements and select components	Define ROS architecture and control framework
03/30	Refine mechanical layout and internal structure; Draft design document	Refine waterproof housing; Support design document work	Complete schematic design and PCB layout planning; Support design document work	Build ROS node skeleton and basic control workflow; Draft design document
04/06	Continue fish body and tail CAD modeling; Finalize design document	Continue waterproof and structural CAD refinement	Complete PCB layout and prepare board fabrication	Set up ROS nodes and controller communication; Finalize design document
04/13	Refine body/tail model; Integrate with electronics and software	Finalize waterproof structure and manufacturing drawings	Begin PCB assembly and initial bench testing	Implement control algorithm and mode-switching logic
04/20	Tune mechanical fit and tail motion; Support subsystem testing	Tune waterproof structure and sealing; Support subsystem testing	Debug PCB, power, and actuator connections; Support subsystem testing	Tune control parameters and ROS communication; Support subsystem testing
04/27	Manufacture and assemble structural parts; Fix prototype mechanical issues; Begin prototype testing	Complete waterproof sealing tests; Support prototype testing	Install PCB and wiring; Check electrical safety; Support prototype testing	Complete software integration; Support prototype testing

Table 9: Weekly Schedule Table (continued)

Week	**Bowen Zhang**	**Xuanyu Ke**	**Kaijun Zheng**	**Libin Wang**
05/04	Lead mock demo; Refine mechanical structure based on demo results	Prepare mock demo; Improve waterproof reliability and assembly details	Prepare mock demo; Fix electrical issues and improve PCB stability	Prepare mock demo; Refine ROS nodes and control algorithm
05/11	Optimize mechanical structure; Document results for final report draft	Optimize waterproof design; Document assembly and testing results	Summarize PCB/electrical results; Improve electronics integration	Refine software performance; Document control results; Organize final project materials
05/18	Final mechanical tuning; Lead final demo and presentation prep	Final water-proof/assembly tuning; Support final demo and presentation prep	Final electrical reliability check; Support final demo and presentation prep	Final software tuning; Support final demo; Polish presentation
05/25	Final report writing, revision, and submission			

5 Ethics & Safety

5.1 Ethics

This project follows the IEEE Code of Ethics and emphasizes responsible engineering decisions that prioritize public safety, honest technical reporting, and awareness of the broader consequences of technology [2]. Since the transformable robotic fish operates with onboard electronics and moving propulsion components in water, the team must treat safety not only as a performance issue, but also as an ethical obligation. Design choices such as conservative waterproofing, electrical isolation, guarded testing procedures, and limited operating conditions are therefore justified because they reduce preventable risks to users and observers.

A second ethical concern is honesty and transparency in reporting project performance. The robotic fish is intended to demonstrate two propulsion modes, but the team should not overstate transformation reliability, underwater stability, or maneuvering capability before these functions are verified experimentally. Any limitations, including leakage risk, unstable transformation, limited operating duration, or incomplete propulsion performance, should be clearly documented in the design document and presentation. This approach is consistent with the IEEE expectation that engineers present claims and estimates truthfully and realistically [2].

Environmental responsibility is also relevant to this project. Because the robot is designed for underwater operation, system failure could introduce battery leakage, broken printed parts, or other contaminants into the water. To mitigate this risk, the team will use controlled laboratory water environments for testing, inspect the enclosure before and after each test, and avoid deployment in natural waterways. This can reduce environmental impact and avoids unnecessary disturbance to aquatic organisms.

Finally, the project should be used only as an engineering prototype for underwater locomotion research and demonstration. It is not intended for harmful interaction with wildlife, or unsafe operation around people. Restricting the system to controlled testing conditions is therefore both an ethical and practical design decision.

5.2 Safety

The primary safety risks in this project are electrical hazards in a underwater environment, mechanical hazards from moving propulsion components, and battery-related hazards. These risks directly influence both the design and the testing procedure.

The first major concern is electrical safety. Because the robotic fish needs to work in underwate environment. Water ingress could lead onboard power electronics and control hardware to short circuits, loss of control, overheating, or permanent damage to the system. To reduce this risk, all electronics will be enclosed inside a sealed waterproof housing, and all cable exits and joints will be protected using sealing features such as gaskets, O-rings, or waterproof connectors. The enclosure design and validation procedure should follow ingress-protection principles consistent with IEC 60529 [3]. Before each powered

underwater test, the team will perform a leak check and visually inspect the enclosure for seal damage.

The second concern is mechanical safety. In propeller mode, the rotating propulsion structure may create contact hazards, while the morphing mechanism may create pinch hazards during transformation. To mitigate these risks, the robot should only be actuated in a controlled tank when operators' hands are clear of the mechanism. A protective shroud or equivalent guard should be added whenever feasible, and propulsion power should remain disabled during adjustment, assembly, or troubleshooting. These precautions are consistent with machine guarding principles used to reduce injury from moving parts [4].

The third concern is battery safety. Because the robotic fish uses a battery inside a sealed enclosure, overheating or electrical failure may be difficult to detect during operation. The battery must therefore be operated within its rated voltage and current limits, securely mounted, insulated from conductive structures, and disconnected before maintenance. The team should immediately stop testing if abnormal heating, swelling, unstable voltage, or evidence of water ingress is observed. These precautions are also consistent with standard electrical safety practices for powered laboratory prototypes [4], [5]

Finally, safe testing procedure is essential. Initial validation should proceed incrementally: dry bench testing, unpowered waterproof testing, shallow submerged testing, and only then full powered underwater operation. During tank testing, at least one team member should be responsible for emergency power cutoff and retrieval of the robot. The system should not be deployed in open water during early-stage testing. These decisions are justified because the project goal is to demonstrate reliable engineering functionality, not to maximize performance at the expense of operator safety.

References

[1] NASA Glenn Research Center. “Shape Effects on Drag,” Accessed: Apr. 7, 2026. [Online]. Available: https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/shape-effects-on-drag/.

[2] IEEE. ““IEEE Code of Ethics”,” Accessed: Feb. 8, 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html.

[3] IEC. ““IEC 60529:1989+A1:1999+A2:2013 CSV”,” Accessed: Apr. 6, 2026. [Online]. Available: https://webstore.iec.ch/en/publication/2452.

[4] Occupational Safety and Health Administration. ““Machine Guarding”,” Accessed: Apr. 6, 2026. [Online]. Available: https://www.osha.gov/machine-guarding.

[5] Occupational Safety and Health Administration. ““Electrical”,” Accessed: Apr. 6, 2026. [Online]. Available: https://www.osha.gov/electrical.

[6] Princeton University. “Mechanics: Drag Coefficients.” For a flat plate normal to the flow, C_D is approximately independent of Reynolds number over the range shown; based on standard fluid-mechanics teaching material including Munson et al., Accessed: Apr. 7, 2026. [Online]. Available: https://www.princeton.edu/~maelabs/hpt/mechanics/mecha_54.htm.

Appendix A Transformable Tail Fin

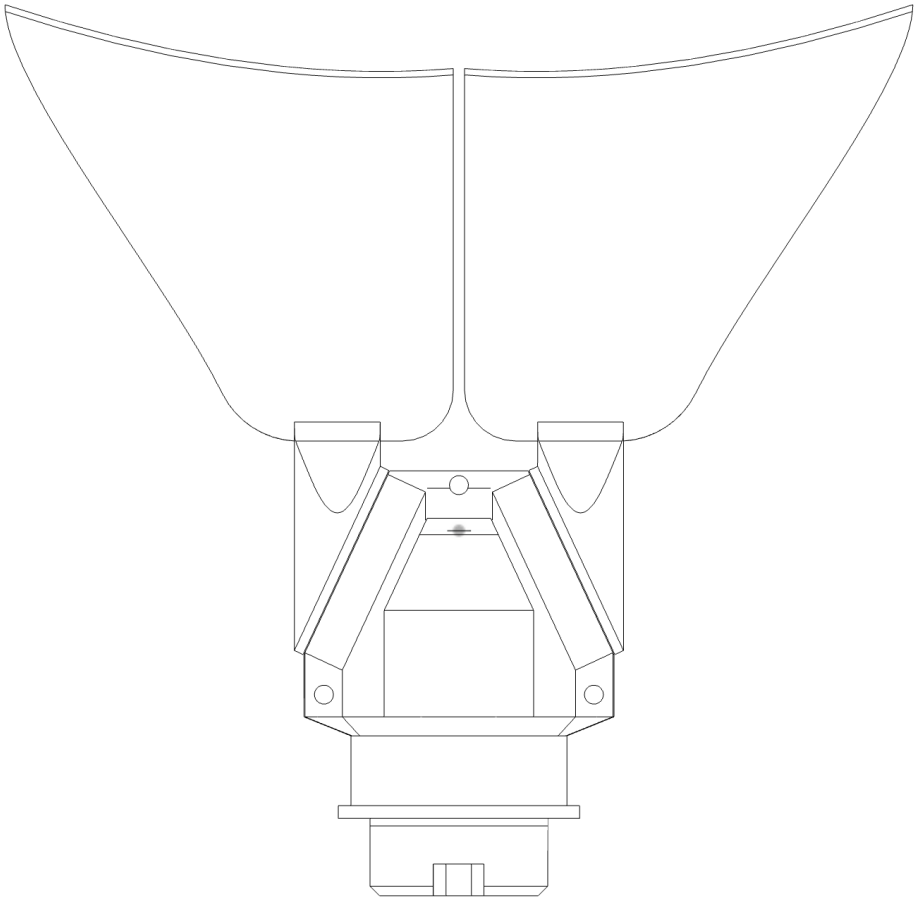

Figure 12: Fin Mode of the Driven Module

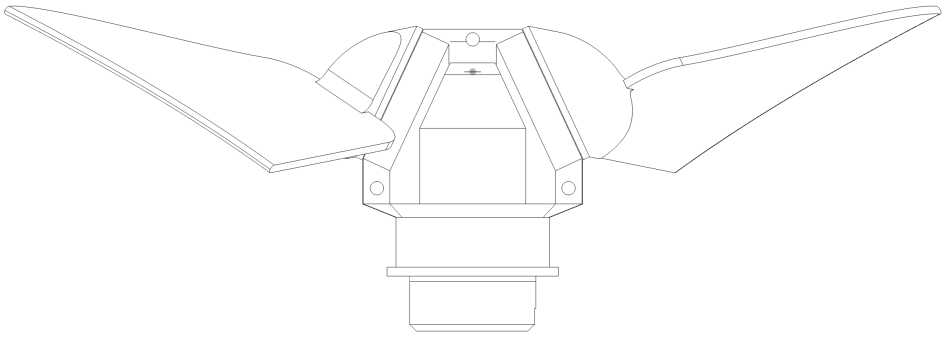

Figure 13: Propeller Mode of the Driven Module