Diyala Journal of Engineering Sciences

Journal homepage: https://djes.info/index.php/djes

ISSN: 1999-8716 (Print); 2616-6909 (Online)

Design and Implementation of an Interactive Embedded System as a Low-Cost Remotely Operated Vehicle for Underwater Applications

Ali Fathel Rasheed[1*], Rabee M. Hagem[2] and Abdul Sattar Mohammed Khidhir[3]

[1] Mathematics Department, College of Basic Education, University of Telafer, Mosul, Iraq
[2] Computer Engineering Department, College of Engineering, University of Mosul Mosul, Iraq
[3] Electronics Technology Department, Mosul Technical Institute, Northern Technical University, Mosul, Iraq

ARTICLE INFO

Article history:
Received June 24, 2024
Revised August 27, 2024
Accepted August 30, 2024
Available online September 1, 2024

Keywords:

Underwater embedded system
Low-Cost remotely operated vehicle
Underwater video capturing
Smart PID Controller
Complementary filter

ABSTRACT

The underwater environment is harsh and challenging for human life, prompting companies and researchers to develop advanced technologies for exploration. Building on previous work that applied a CNN-based method for underwater object classification, this paper focuses on the design and implementation of an interactive embedded system for a compact Remotely Operated Vehicle (ROV) with specific dimensions and weight. The primary goal is to capture real-time underwater video using remote control communication via Ethernet. The ROV is powered by five brushless motors controlled by a smart PID controller. Precise pulse-width modulation signals enhance stability during movements along three axes, enabling high-resolution video capture. The system utilizes the Raspberry Pi 3's computing power for motion control, positioning, temperature monitoring, and video acquisition. Experimental results demonstrate the system's capability to process 42 frames per second. A user-friendly graphical interface allows for remote ROV control across various operating systems. With a depth rating of 100 meters and speed of 0.148 m/s. This ROV surpasses human divers' limitations and holds significant potential for applications in surveillance, operations, maintenance, and measurement tasks underwater. The proposed ROV contributes to the advancement of underwater exploration technology by combining high performance with cost-effectiveness.

1. Introduction

Remotely operated vehicles (ROVs) have garnered significant attention in recent years, with numerous studies exploring their application in oceanic research [1, 2]. These underwater robots are particularly valuable for repair, support, and maintenance across various marine industries, including gas and oil extraction, aquatic infrastructure, naval defense, ocean science, and renewable marine energy [3-5]. Notably, ROVs excelled in recognition tasks, such as identifying and tracking underwater mines, as demonstrated in research focused on high-risk operations utilizing advanced prediction, diagnosis, and classification algorithms [6, 7]. Additional research endeavors have emphasized the role of ROVs in underwater surveillance, preparing them for tasks with clearly defined objectives [8, 9]. Moreover, there is significant interest in employing ROVs for laying optical fiber cables on the ocean floor [10, 11]. ROVs also play a critical role in marine archaeology, aiding in the investigation of hazardous underwater brine pools and the inspection of ancient submerged structures [12,13]. Maintaining station-keeping and tracking position are primary advantages of

* Corresponding author.
E-mail address: rabeehagem@uomosul.edu.iq
DOI: 10.24237/djes.2024.17312

these vehicles, ensuring that the vehicle consistently maintains its position relative to the structure at all times [14]. These vehicles are also highly effective in deep-sea exploration, in areas inaccessible to humans [15]. It is crucial to design and construct the robot or vessel in a cost-effective manner [16]. As water pressure increases with depth, it negatively impacts the stability of the underwater vehicle. The vehicle's control system works to avoid disturbances such as drag effects, ocean currents, and obstacles. The placement and orientation of thrusters, control surfaces, and the vehicle's shape are among the constraints that limit autonomous control [17, 18]. Due to these factors, maintaining the vehicle on the desired course and completing underwater missions is challenging. Enhancing the vehicle's control system is crucial, as it reduces both the time and cost of the mission [19]. Similarly, the precise allocation of thrusters ensures accurate movement and efficient use of energy resources [20]. High-performing control systems under these constraints enable a high degree of autonomy [21]. In such systems, an umbilical cable connects the operator to the vehicle, which is controlled from a surface platform [22]. Uncontrolled systems cannot achieve full control over their dynamics due to extreme underwater conditions [23]. Therefore, an optimally designed ROV with a suitable control unit is essential for completing underwater scenarios. Furthermore, the equipment and sensors used in an ROV impact the system's dynamic behavior [24]. Some ROV applications require precise positioning and navigation capabilities, achieved through fiber-optic gyroscope-based Inertial Navigation Systems (INS) [25].

Several pre-designed control systems exist for underwater vehicles, including sliding mode control, Proportional-Integral-Derivative (PID) control, and robust adaptive control [26]. Currently, the underwater robotics community is actively engaged in the modeling, design, and construction of intelligent ROVs [27].

This study focuses on developing an ROV system designed for remote control and monitoring via communication with a Surface Workstation. At the heart of the ROV is a Raspberry Pi 3 System on a Chip (SoC) board, chosen for its cost-effectiveness, multi-processing power, straightforward task management, and compatibility with Python—an open-source programming language (version 3.7), along with libraries such as OpenCV for image processing. This board interfaces with a Pixhawk flight controller connected to various sensors. Additionally, the integration of a complementary filter with an advanced Proportional-Integral-Derivative (PID) controller, as discussed in [28], is proposed to significantly enhance trajectory control. This improvement is critical for obtaining higher-quality underwater images using the camera linked to the Raspberry Pi. Moreover, communication protocols have been tailored to ensure flawless interaction between the 'QGroundControl' ground station and the drone’s control software, utilizing the Ardupilot engine library and advanced message analysis techniques. These enhancements contribute to a more responsive system capable of managing simultaneous operations more efficiently.

Key contributions of this work include the development of a cost-effective yet high-performance ROV system that integrates advanced control and communication technologies. The use of a Raspberry Pi 3 combined with a Pixhawk controller and customized protocols significantly improves the ROV's operational efficiency and image capture quality, providing a robust solution for underwater exploration.

The remainder of this paper is organized as follows: Section 2 provides an in-depth examination of the autopilot's software and hardware configurations. In Section 3, a description of the hardware components integrated into the ROV is presented. Section 4 explores the architecture of the pilot software, followed by a series of tests and the presentation of experimental outcomes that validate the functionality of the pilot system. Section 5 addresses the threats to validity, discussing potential limitations and vulnerabilities of the study. The paper is concluded in Section 6, which offers a

summary and recommendations. Lastly, Section 7 discusses future work and provides guidance for subsequent research.

2. Underwater vehicle autopilot systems

2.1 Ardupilot software

Ardupilot is a versatile, open-source autopilot software widely adopted in various vehicles, including unmanned underwater vehicles [29]. It features advanced data logging, analysis, and simulation tools, which contribute to its reliability and popularity. As depicted in Figure 1, the software architecture on the Pixhawk platform utilizes the ChibiOS real-time operating system to achieve precise timing and resource efficiency. The core architecture includes a hardware abstraction layer, sensor drivers, and a comprehensive software library, ensuring seamless integration with diverse hardware components. Ardupilot supports multiple vehicles through different versions, including ArduPlane, ArduCopter, ArduRover, and ArduSub. For those interested in further details, comprehensive documentation and resources are available [30].

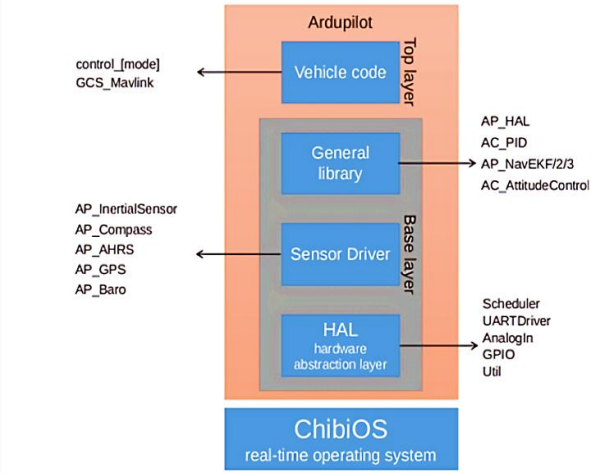


Figure 1. The architecture of the Ardupilot system [31]

2.2 Pixhawk

Pixhawk represents an independent open-hardware project that provides affordable, high-quality autopilot hardware designs widely used in academic, hobbyist, and industrial settings. This flight controller is particularly common among multirotor platforms due to its balance of cost-effectiveness and functionality. Importantly, it allows for integration with a computer through its two-port telemetry, as shown in Figure 2. Within its framework, Pixhawk effectively handles MAVLINK commands through the PX4 stack controller [32].

2.3 Raspberry Pi

Raspberry Pi is a single-board computer [33] known for its robust computational power housed in a small, affordable unit. When connected to the Pixhawk, it receives input power (ground and 5V) through the GPIO pins. Two GPIO pins are used for data exchange through UART communication, as demonstrated in Figure 2.

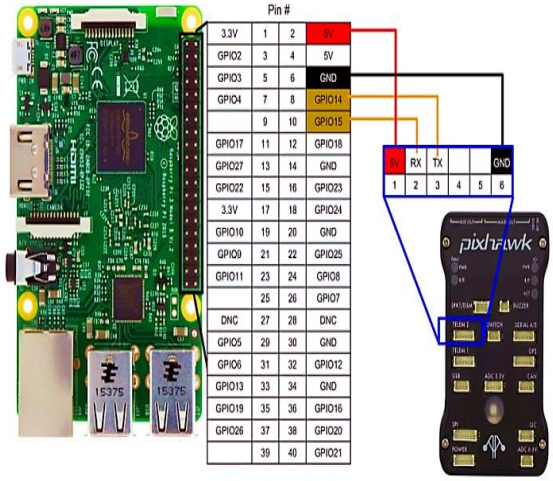


Figure 2. Raspberry Pi 3 to Pixhawk connection diagram [32]

3. Structure design and mechanical of the designed ROV

The intended mechanical design of the Design ROV focuses on creating a structure that is simple to assemble, lightweight, and agile, thereby enhancing the efficiency of task execution within a feasible period. To achieve a blend of optimal performance and cutting-edge design, computer-aided design tools were utilized, coupled with 3D modeling and analysis software. This approach allowed for simulations that helped verify the design's effectiveness [34].

3.1 Frame of the proposed ROV

The structure of the proposed ROV features a closed hull design resembling a turtle, chosen for its simplicity, ease of construction, and excellent hydrodynamic properties. Constructed from PLA filament and acrylic, the design balances lightness with durability. The structure is divided into three sections: the top flange, central tube, and bottom flange, as shown in Figure 3. The main acrylic tube houses the electronics, while a smaller tube encases the camera, which is crucial for navigation and image processing. The flanges ensure a secure, leak-proof seal. The implemented ROV was tested in a controlled environment, practical experiment has been conducted to estimate the speed of the proposed ROV through setting a 1 m distance in water to be covered by the ROV. The recorded time was 6.77 s. The ROV's speed is calculated to be approximately 0.148 m/s.

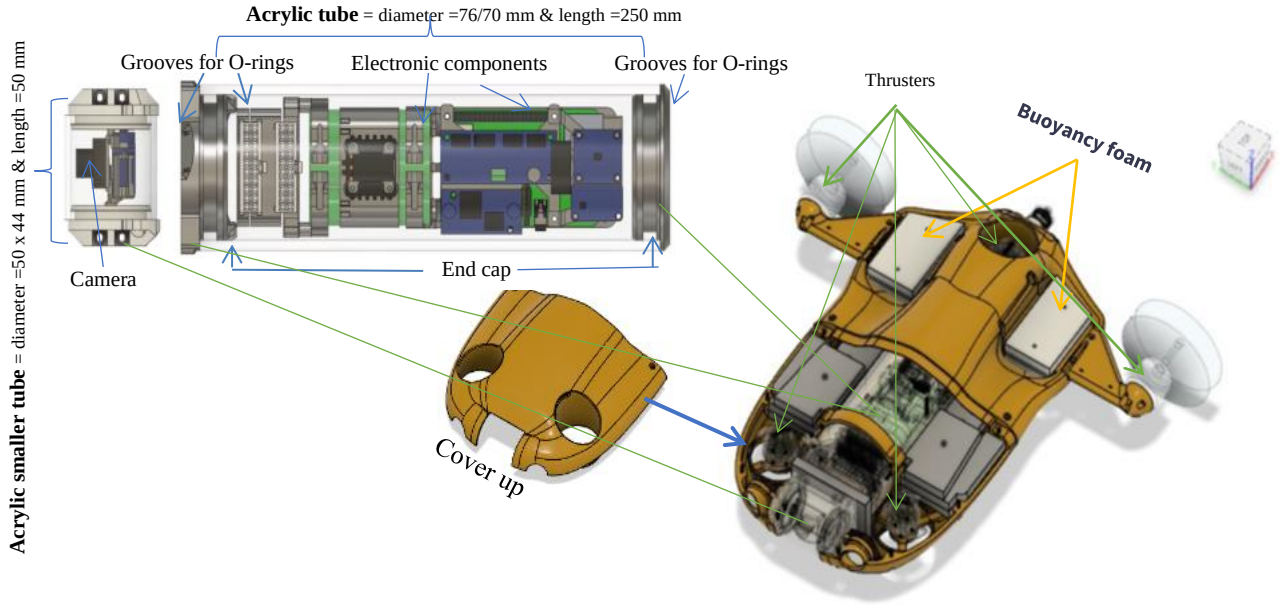


Figure 3. The proposed ROV structure

3.2 Water sealing part

To ensure effective resistance against water infiltration, static and dynamic sealing mechanisms are implemented. Flat gaskets and standard O-rings are employed for sealing the structure of the Implemented ROV, as shown in Figure 4. The dimensions of these seals are carefully selected to match the acrylic tube diameter. Additionally, sealed cable glands equipped with seals are utilized for securing both outgoing and incoming cables, including the cable used for communication. The chosen gaskets are pressure-rated epoxy, with two gaskets employed to ensure optimal sealing of the onboard system.

It is important to ensure that the correct proportions are marked on epoxy bottles. If the epoxy follows different proportions (e.g., 3:1 or 1:1), the weights of components mentioned later need to be converted to match the ratios. For instance, 12g of epoxy in 3:1 proportion equates to 9g of component A and 3g of component B.

The total weight of the Implemented ROV, including all components, is estimated to be 2.5 kg, with the epoxy contributing approximately 60g to this total. This detailed breakdown of component weights ensures that the Implemented ROV's sealing mechanisms are optimized for both weight and performance.

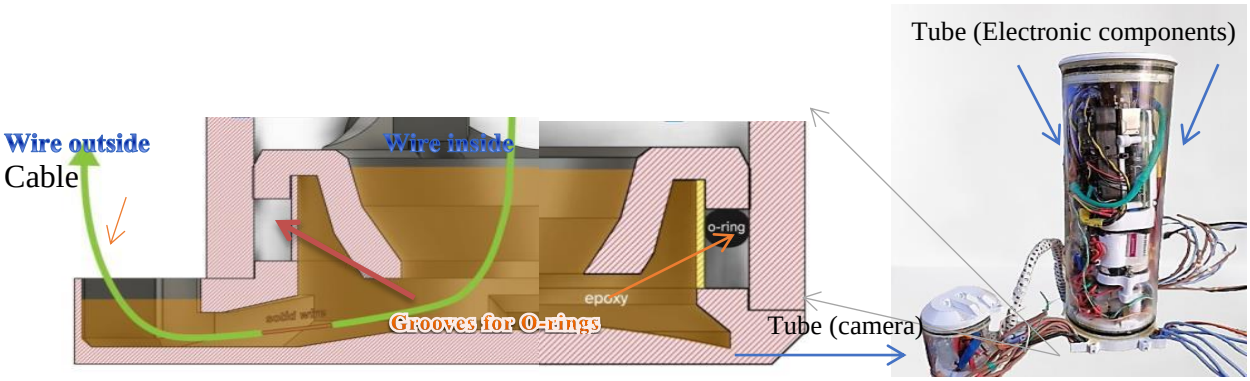


Figure 4. Cross-sectional view of the water sealing part, illustrating the sealing mechanisms at both ends of the assembled hull

3.3 Electronic design

Figure 5 illustrates the block diagram of the proposed system setup, consisting of two major components: the Implemented ROV and the surface workstation. These parts are connected through a Kevlar-reinforced Ethernet cable, allowing users to control the Implemented ROV via Virtual Network Computing (VNC) from personal computing devices like PCs, laptops, or Systems on Chip (SoC).

The hardware of the Implemented ROV includes a Raspberry Pi 3 and a Pixhawk microcontroller, responsible for executing parallel tasks such as video acquisition using a digital camera, measurement, and recording of various sensor data, motor control in coordination with the Pixhawk, battery voltage monitoring for energy consumption and internal temperature, and communication management with the remote control. The digital camera used is a Sony full HD 1080p with 2 Megapixels, housed in a waterproof tube as shown in Figure 4 to facilitate underwater operations.

Power Source: A 19,800 mAh lithium battery is used to supply 5V power to the Raspberry Pi 3 and, via USB, to the Pixhawk controller and digital sensors. Additionally, six 11.1V batteries of the same capacity are used to power the five brushless motors. These battery banks provide the Implemented ROV with up to two hours of autonomy. Additional components include Electronic Speed Controllers (ESCs) for the thrusters, a pressure sensor for depth measurement, lights for visibility, and a power module to manage power distribution.

The surface workstation comprises a gamepad controller for manual ROV control, a topside computer that acts as the primary interface for the operator, WEIPU waterproof connectors ensuring reliable communication between the Implemented ROV and the surface workstation, and an Ethernet network that manages communication between the Implemented ROV and the surface computer. This comprehensive setup ensures that the Implemented ROV can be effectively controlled from the surface, with all critical functions and data managed seamlessly. The robust communication system and efficient power management contribute to the overall reliability and performance of the Implemented ROV.

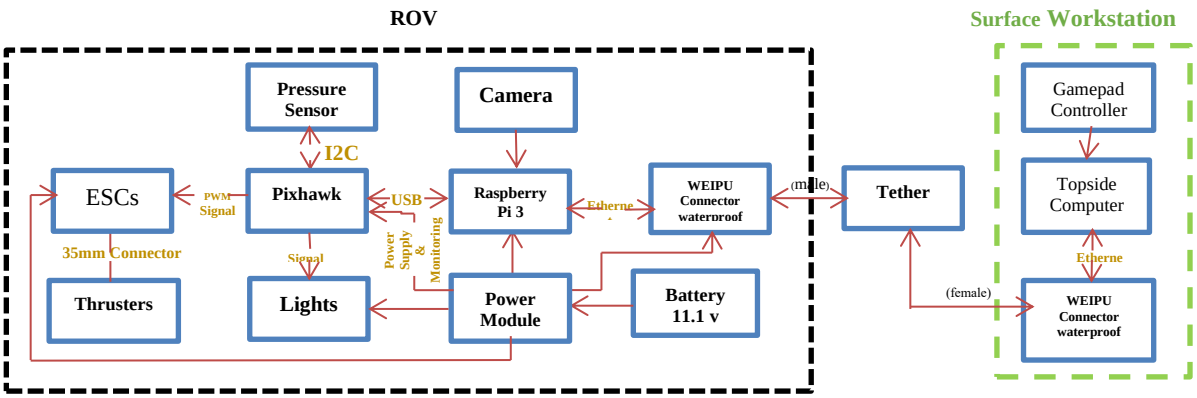


Figure 5. Block diagram of hardware components on the ROV design, illustrating the integration of ROV and surface workstation components, their interconnections, and functional roles

To drive the Implemented ROV and to convert the electrical energy into mechanical power, a brushless DC (BLDC) motor combined with a propeller is used. Essentially, this motor operates based on the attraction or repulsion of poles and magnets [35]. The rotor of this motor contains permanent magnets that keep constant poles, while the stator has windings that change magnetic poles depending on the current polarity in the stator windings. Figure 6 shows the BLDC motor structure, which includes twelve stator windings and eight magnetic poles on the rotor.

Figure 6. Internal Structure of a BLDC motor [33]

In this implementation of the BLDC motor, a DC source is responsible for providing the primary energy, which is then converted into AC voltage by an Electronic Speed Controller (ESC). To enable bidirectional rotation, the ESC needs to be modified by adding a relay, as shown in Figure 7. The equations below show how to determine the proper position and location of the Implemented ROV.

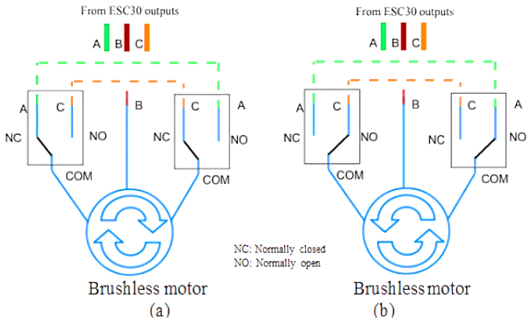


Figure 7. illustrates the electrical circuit configured to reverse the motors' rotation direction, with part (a) displaying clockwise rotation and part (b) showing counterclockwise rotation [36]

Equation 1 represents the transfer function for the complementary filter used to fuse the angular measurements from the accelerometer and the gyroscope. This equation combines these measurements to produce a more accurate estimation of the angular position [37].

$$\omega_e(z) = \frac{\omega_{c1}\theta_m(1-z^{-1})}{dt}dt + \omega_{c1}\theta_m(z) + \frac{(1-z^{-1})(1-z^{-1})/dt}{(z^{-1}-1)/dt + \omega_{c2}\dot{\theta}_m(z)} \quad (1)$$

where $\theta_m$represents the angular measurements in degrees obtained by the accelerometer, while $\dot{\theta_m}$ signifies the data of the angular velocity captured in degrees per second by the gyroscope. θ_e denotes as complementary filter output, also expressed in degrees. The parameters ωc1 and ωc2

correspond to the cut-off frequencies of the low-pass and high-pass filters, set at 5 Hz and 2 Hz, respectively. Additionally, dt represents the differential time measured in seconds. By applying the inverse Z-transform and rearranging the terms, the simplified form of the transfer function is directly obtained as shown in Equation (2):

Equation (2) provides a simplified relationship between the complementary filter output and the angular measurements, incorporating the inverse Z-transform [37].

$$\theta_e(z) = \left[\frac{(1-z^{-1})}{dt} + \omega_c\right] = \left[\omega_c\theta_m(z) + \dot{\theta}_m(z)\right] \quad (2)$$

Equation 3 shows the discrete form of the complementary filter output after applying the inverse Z-transform [37].

$$\theta_e(k) = \left[\frac{1}{dt + \omega_c}\right] = \left[\theta_e(k-1) + \omega_c\theta_m(k) + \dot{\theta}_m(k)\right] \quad (3)$$

Equation 4 provides the simplified result of the complementary filter output used for stability enhancement in the proposed [37].

$$\theta_e(k) = (1-\alpha).\left[\theta_e(k-1) + \dot{\theta}_m(k).dt\right] + \alpha\,.\,\theta_m(k) \quad (4)$$

where $\alpha = 0.02$ & k is the sample time $[0 - 60$ s].

The implementation of the complementary filter aids in enhancing the stability of the proposed ROV by significantly reducing noise, as depicted in Figure 8. This noise reduction contributes to an improvement in the quality of the captured images.

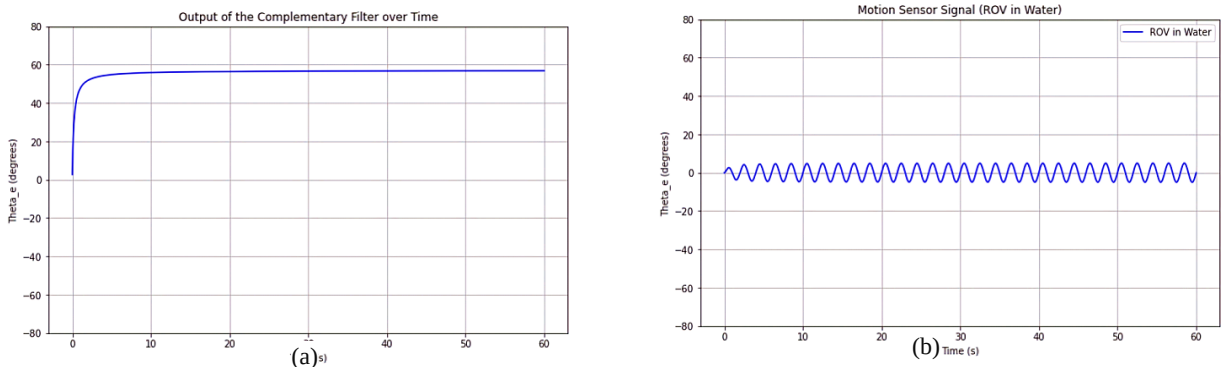


Figure 8. Illustrates the motion sensor signal obtained through the utilization of a complementary filter: (a) depicting the ROV in motion with direction in the one-axis, and (b) illustrating the ROV when parked in water

3.4 Actuation and control

ROVs are susceptible to parametric variations such as buoyancy, weight, payload, added mass, etc., and disturbances from external sources like currents in the ocean. The suggested subsystem Implemented ROV is controlled by a smart PID controller developed by Yanlin He et al. [37], which can adjust PID controller gains in real time using an Artificial Neural Network (ANN). This paper proposes incorporating a complementary filter with the output of the MPU6000 sensor to enhance the performance of the PID and increase the stability of the Implemented ROV, thereby improving the quality of underwater images.

In the discrete time domain, the digital PID algorithm is represented as in equation 5 [38]:

$$\tau(k) = K_p \cdot e(k). + K_i \sum_{i=0}^{k} e(i) + K_d\big(e(k) - e(k-1)\big) \quad (5)$$

where $\tau(k)$ is the control signal at sample time of k ; $e(k)$ is the position tracking error at sample time of k ; to represent $[e(k) = \eta_d(K) - \eta_f(K)]$; $\eta_d(K)$ It denotes the Intended trajectory; $\eta(K)$ is the realized trajectory without processing; $\eta_f(K)$ are the realized trajectory of the filtered data; K_p , K_i and K_d represent the proportional, integral, and derivative gain respectively. Figure 9 illustrates the block diagram of the self-tuning PID controller based on ANN [38, 39]. The complementary filter, shown in Figure 9, helps reduce noise levels in the MPU6000 sensor output. As shown in Figure 5, the actuation subsystem includes a Pixhawk microcontroller ESC30A, a power stage with five-speed controllers, twelve relays for controlling

thruster direction, and five A2122 1000 kV thrusters/motors with propellers. Figure 10 depicts the reaction of the controller with smart PID to an input of a unit step over a 500 ms period. The graph indicates that when there is a sudden change in the input, the stabilization response time is under 30 ms.

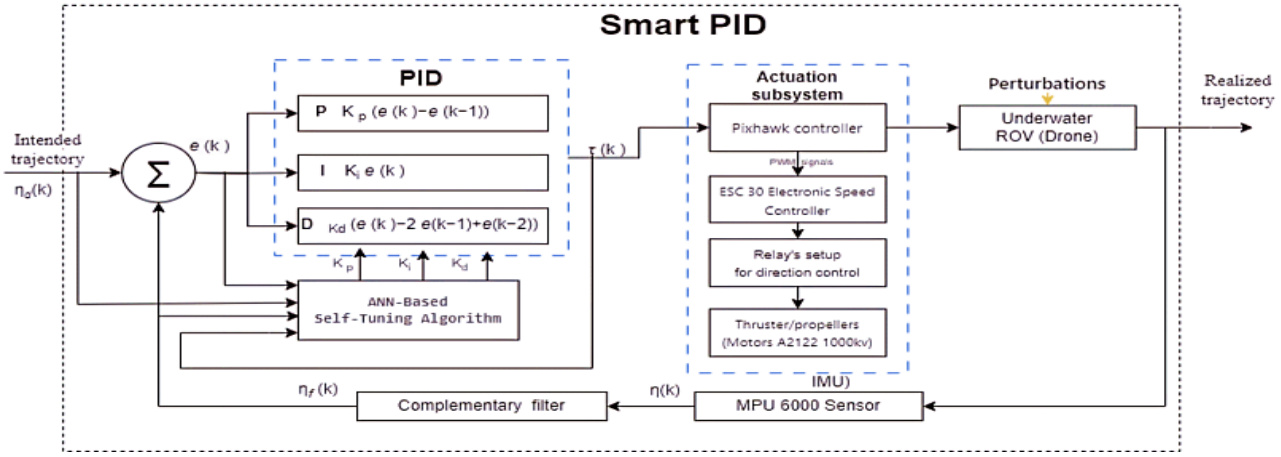


Figure 9. Block diagram of the smart PID controller for underwater proposed ROV

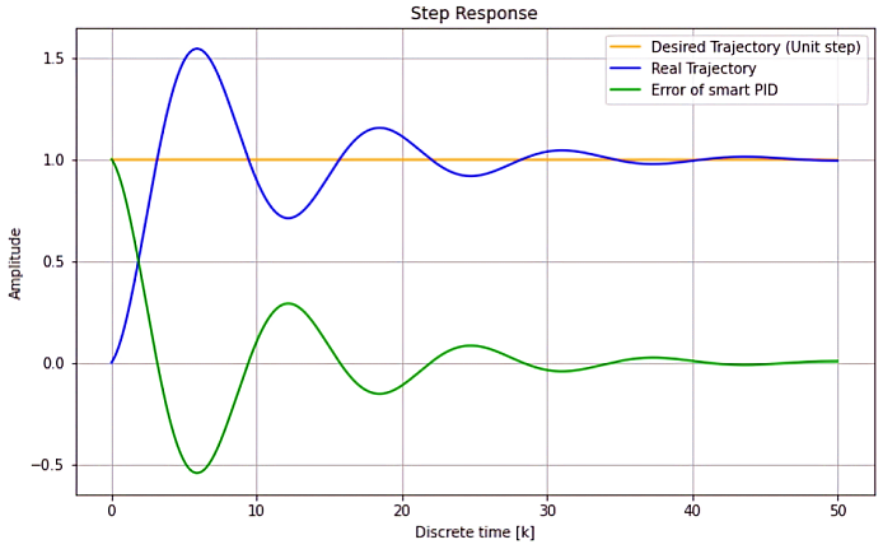


Figure 10. Simulation showing the response of the smart PID controller to a unit step input

3.5 Configuration of Thrusters

In the design, the thrusters take inspiration from BlueRobotics' models, specifically the T100 and T200 [40], utilizing their efficiency and reliability. These thrusters have propellers modeled after boat propellers, specifically the three-blade design, which aligns well with our application needs. Assembling these thrusters is straightforward, making them ideal for the engines that are implemented in our system. For propulsion, we chose A2212/13T BLDC motors paired with reversible 30A ESC controllers. These motors have sealed windings, removing the need for additional sealing. The fairing assembly, including the propeller attachment, uses fixing screws, with the motor and propeller attached to the fairing using the provided motor support, as shown in Figure 11 below.

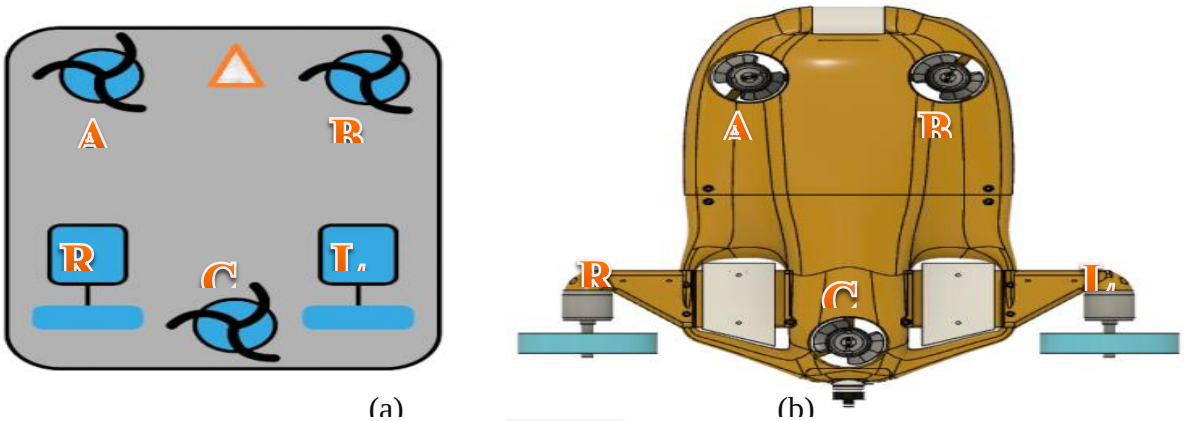


Figure 11. Thruster configuration includes (a) Motor Positioning [41], and (b) a 3D design view of the proposed ROV. Note that the direction of motion is represented by the red triangle

During our decision-making process, we computed the maximum force required for vertical movement, which is vital for choosing appropriate thrusters capable of both ascent and descent. We selected 5-DOF motors similar to the 6-DOF thrusters from Blue Robotics T100, recognized for their excellent thrust-to-volume ratio [42], providing sufficient space for integrating the necessary mechanisms. However, it should be noted that these thrusters consume more power during high-speed operations. Equation (6) outlines the calculation of the drag force [43], given by:

$$D = 0.5A\rho V^2\ C_{d\_}d \quad (6)$$

where D represents the force because of the drag; A stands for the cross-sectional area of the ROV; ρ denotes the density of the fluid, V indicates the speed of the object relative to the fluid, and C_d represents the drag coefficient. To optimal maneuverability and facilitate lateral movement, three thrusters were strategically positioned on a primary plane, each angled at 90° in the corners, as shown in Figure 11a. Two motors were placed at the front of the Implemented ROV (in a forward position), while one thruster was positioned at the rear (in a backward position). Additionally, two vertical thrusters were mounted on the vehicle's right and left sides. This arrangement enabled the vehicle to achieve 5-DOF (Surge, Sway, Heave, Yaw, Roll, Pitch), as depicted in Figure 11b.

3.6 Mechanical design considerations

The physical properties of the materials utilized in the construction of the Proposed ROV for underwater operation are detailed in [44]. In this paper, PLA Filaments (Polylactic Acid) and acrylic pipes reinforced with steel bars are employed as key structural elements. These materials were chosen because they meet the necessary mechanical specifications for the Proposed ROV to operate effectively and dictate the maximum operational depth of the vehicle. The depth limit is calculated using Equation (7) [43], which requires knowledge of the material's fracture point—specified by the manufacturer, which in this instance equates to a pressure of 1.2 MPa. Given these parameters, the maximum allowable depth for a Schedule 40 plastic pipe is calculated to be h=122.33 m.

$$P = \rho \cdot g \cdot h, \quad (7)$$

where P represents pressure; $\rho = 1000\ \mathrm{kg/m^3}$ denotes the density of fresh water; g stands for gravity; and h indicates depth in meters. Figure 12 illustrates the proposed ROV with 3D structural design, which accounts for hydrostatic and buoyancy factors relevant to the operational environment—whether it be a marine setting (saltwater) or a controlled setting (freshwater). The design features a weight-balanced configuration to streamline the integration of motor systems, developed using the 3D figure simulation software using, Autodesk Fusion 360. The physical dimensions of the Proposed ROV are 40.1741cm × 39.1465 cm × 9.8729 cm, resulting in an estimated

volume of $V = 15{,}525.3\ cm^3$ and a total weight of approximately W=2.5 kg. With the freshwater density set at $\rho = 1000\ kg/m^3$, the resultant buoyancy is calculated to be approximately 152.3N. This buoyancy is derived from Equation (8) [45], where E represents the total thrust, ρ the density of the fluid, g gravity, and V the volume. Consequently, the topology designed for the five motors in the proposed ROV requires a minimum thrust force of 2.6 N, to ensure optimal performance.

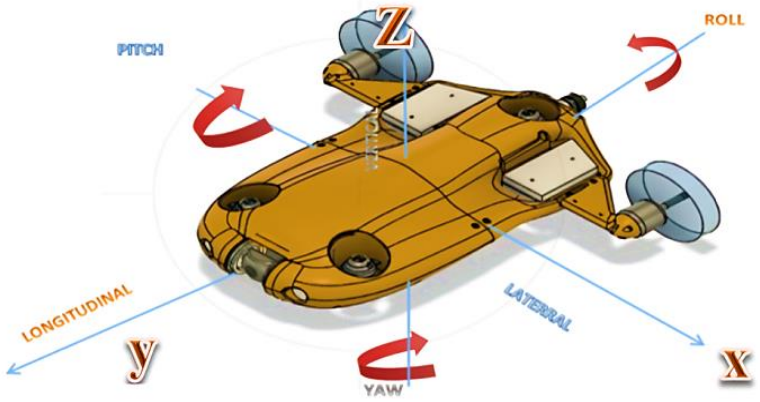


Figure 12. The motion of the proposed ROV in 5-DOF

Figure 11 offers a 3D design view of the Proposed ROV used in constructing its mechanical framework. This structure is specifically designed to house five brushless motors, as shown in Figure 12. Table 1 provides a detailed breakdown of the components used to implement the Proposed ROV, along with their estimated weights, contributing to the total weight of 2.5 kg. To ensure stability in water and to match the buoyancy calculated using Equation 8 [45].

$$\rho \cdot g \cdot V = E \quad ..(8)$$

where: E is the buoyancy force, ρ is the density of the fluid (e.g., water), g is the acceleration due to gravity, and V is the volume of the displaced fluid, the design incorporates eight steel bars. Furthermore, the finalized ROV structure allows for flexible camera placement, enabling positioning at various locations such as the front, underside, or sides of the Proposed ROV.

Table 1: The estimated weights and costs of the materials used for the implementation of the proposed ROV

Item	Quantity (No.)	Weight [kg]	Total Weight [kg]	Price ($) +Taxes (100$)
PLA filament	1	1.350	1.350	70
Electronic Acrylic Tube	1	0.255	0.231	30
Camera Acrylic Tube	1	0.050	0.050	25
Pixhawk 2.4.8	1	0.038	0.038	100
Raspberry Pi 3 model	1	0.100	0.100	**70**
ESC 4in1 (30A+)	1	0.005	0.005	30
ESC1 in 1(50 A)	2	0.003	0.006	25
Sony ELP camera	1	0.030	0.030	35
batteries18650	6	0.030	0.300	20
Brusless motor A2212	5	0.049	0.147	50
Payload* steel bars	8	0.12,5	0.100	5
Other Components **	1	0.143	0.143	40
			2.500	$500

* (when using salt water)

** (Wires, Connectors, nuts of various sizes, CAT6 Ethernet cable, Other Accessories**)**

3.7 Main Circuit Board of the ROV

The components of the proposed ROV operate in a master-slave configuration, with the Raspberry Pi serving as the central processing unit. The Raspberry Pi is equipped with a network card and is assigned an IP address, enabling the ROV to function as an autonomously connected device. This setup allows for remote control from a ground station computer on the surface through a tether connection. To provide autonomous navigation capabilities, the Pixhawk autopilot card is utilized in conjunction with the QGroundControl interface application, as depicted in Figure 13, ensuring stable and reliable operation. Additionally, to enhance algorithm performance, communication is established between the Raspberry Pi and the Pixhawk autopilot card, facilitating data acquisition from sensors and the execution of commands for the thruster drivers. This integration allows for seamless transmission of sensor data to the Raspberry Pi and supports the implementation of autonomous navigation and image processing software.

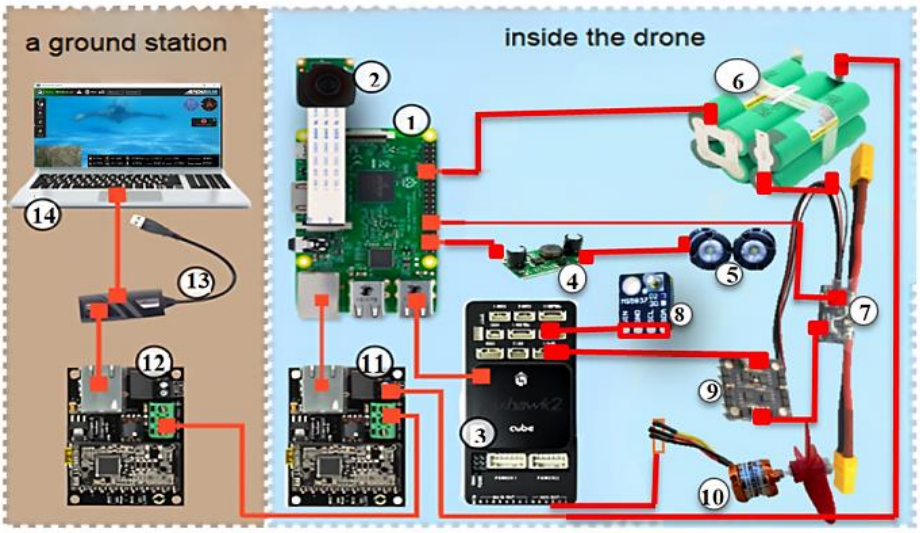


Figure 13. The numbered components are: (1) Raspberry Pi 3. ,(2) RaspiCamera. ,(3) Pixhawk v2. ,(4) Step-down 3W (LED×2). ,(5) Lamps (light). ,(6) battery LiPo 12.6V -3Cell. ,(7) Pixhawk power module. ,(8) Pressure sensor GY-MS5837 -30db. ,(9) Speed Controller (ESC ×5). (10) Thrusters (motor A2212 ×5). ,(11) and (12) Fathom (WEIPU Connector) ,(13) Ethernet-USB connector. (14) Laptop (ground station)

3.7.1 Sensor and depth measurement

The MPU6000 sensor, featuring both a gyroscope and accelerometer, is supplemented with an additional MEAS barometer. It comes equipped with I2C and SPI ports. However, due to potential conflicts with SPI libraries and command functionalities, communication with the Pixhawk boards is prioritized through the I2C protocol. To measure the depth achieved by the Implemented ROV, a piezoelectric pressure sensor (GY-MS5837-30db) was integrated. This sensor provides precise depth readings by detecting the exerted force caused by water on the Implemented ROV, allowing accurate determination of the vehicle's position along the Z-axis. Additionally, the module is equipped with a temperature sensor to provide essential information about the surrounding underwater environment.

3.7.2 Camera

The vital function of the proposed ROV is to conduct observations in an underwater environment, necessitating a camera that meets the predefined specifications. To fulfill this requirement, the proposed ROV is equipped with a camera that includes an OV4689 CMOS sensor from Sony. This sensor is the first 2-megapixel sensor in the camera module, offering high sensitivity and high-speed video support (MPEG 60fps at 1920 x 1080). Data

transmission from the camera to the Raspberry Pi is facilitated through the MAVLink Protocol (Mobile Industry Processor Interface–Camera Serial Interface), a protocol that is already established and standardized for the Raspberry Pi.

3.7.3 Power management for underwater vehicles

The power supply system for underwater vehicles presents a significant challenge related to the design process of the implemented ROV. Based on the distance from the ground station two feasible technical solutions are presented in [46]. The first way is a power supply system that is cable-based utilizing the required voltage and minimizing the voltage drops. The second way is a battery-powered system with sufficient capacity to complete a required task. The first option requires the integration of a step-down transformer within the ROV. However, this method adds extra weight and considerable volume, potentially compromising the structural integrity of the vehicle. Additionally, the inclusion of a power cable can increase drag on the vehicle, affecting its mobility. After a thorough feasibility study, this option was deemed impractical and was discarded.

The alternative, a battery-powered solution, was chosen, despite its limitations. Given the specific energy requirements of our system, it is crucial to select a battery with a capacity that can complete the specific required task. To address this, a power balance analysis is conducted, aggregating the power requirements of each component (refer to Table 2). The maximum energy consumption over one hour is estimated to be 119.86 Wh.

Table 2: The proposed ROV Energy budget calculation

ROV's components	**Voltage (V)**	**Max Current(A)**	**Energy (Wh)**	**(Number)**
Pixhawk 2.4.8	5	0.5	2.5	1
raspberry (model 3B+)	5	0.48	2.4	1
Sensor	3.3	0.00125	0.004125	1
A2212 Brushless motor (1000 Kv)	11.5	2.1	105	5
LED Driver Module (DC to DC Step-Down)	3	1.66	9.96	2

3.8 Algorithm design process

The Pixhawk and Raspberry Pi communicate via serial ports. To minimize data loss, data is transmitted from the Raspberry Pi to a computer via an Ethernet cable. The robot is controlled using the fully open-source ArduSUB Firmware. Control is provided by the QGroundControl software. Additionally, an autonomous interface is provided using C# from the .NET family. Autonomous commands written in Python will be called from the C# interface.

The robot operates in various predefined modes to adapt to different operational scenarios, controlled by the pilot as per task requirements. These modes include:

- Stabilize Mode: Maintains balance and altitude autonomously based on sensor data.
- Acro Mode: Stabilizes angular velocity.
- Manual: Direct motor control without sensor data.
- Depth Hold: Maintains depth when the pilot is hands-off.
- Position Hold: Balances position, attitude, and orientation without pilot inputs.
- Auto: Executes pre-stored missions autonomously, typically ignoring pilot inputs.
- Guided: Dynamically adjusts target position via external control.

3.8.1 ConnectMyROV

Function facilitates the establishment of communication between the on-board unit and the ROV. For a real drone, communication is set up via the serial port USB, whereas for the Hardware-in-the-Loop (HITL) drone, it

connects through the local host IP address 192.168.2.2 at port number 2770. The communication process is detailed in the flowchart presented in Figure 14.

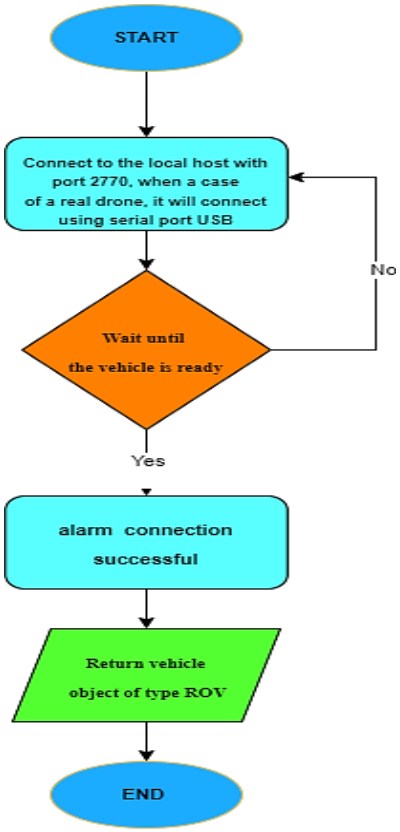


Figure 14. The connection to the Implemented ROV flowchart

3.8.2 Arm-and-Diving float (target depth)

This function is employed to activate the motors of the Implemented ROV and switch the flight mode to GUIDED mode to take control of the ROV using the Raspberry Pi 3. Then, Diving floats to a specified depth, and the function verifies each state to ensure proper operations for the Implemented ROV. This function is illustrated in the flowchart in Figure 15.

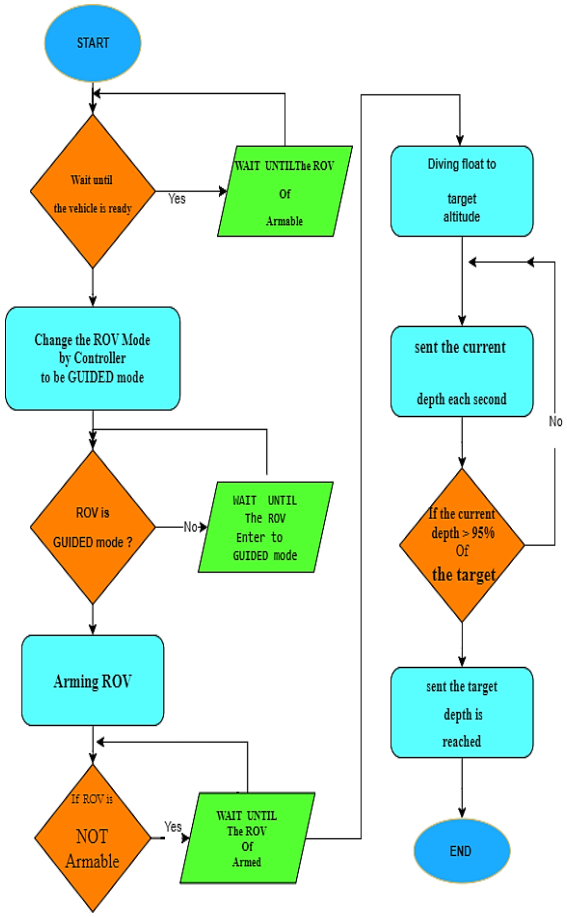


Figure 15. Arm and Diving float flowchart

3.8.3 Get-Distance-Meters (target-location, current-location)

The function calculates and returns the distance between two specified locations in meters. It requires two parameters: target_location and current_location, which represent the coordinates of the desired and current locations, respectively.

3.8.4 The goto(target_location)

The function directs the Implemented ROV to dive toward the specified target location. It continuously calculates the remaining distance to the target every second. When the remaining distance reaches 10 cm, the Implemented ROV is considered to have arrived at the target location. The operational flow of this function is depicted in the flowchart presented in Figure 16.

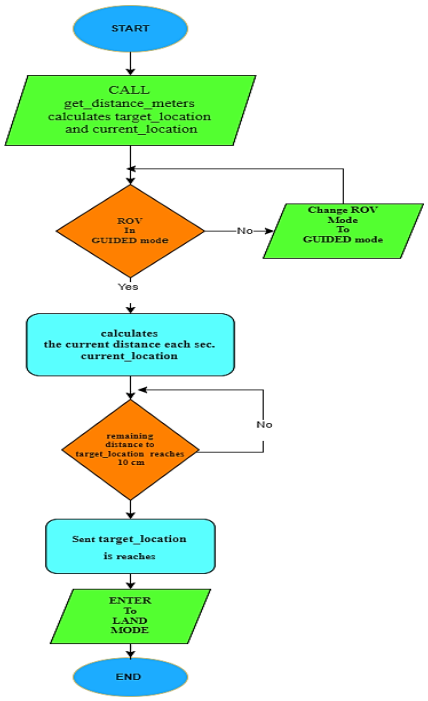


Figure 16. Goto target location function flowchart

3.9 Design overview of the proposed ROV

The design of the Proposed ROV is centered on achieving a balance between lightweight construction and durability, tailored specifically for underwater operations. The ROV features a closed hull design inspired by the streamlined shape of a turtle, optimizing it for superior hydrodynamics. The structure is primarily constructed from PLA filament and reinforced with acrylic pipes and steel bars, which collectively contribute to its resilience in harsh underwater environments.

Mechanically, the ROV is equipped with five strategically positioned brushless motors that enhance maneuverability while maintaining buoyancy and stability. The electronic system is integrated with a Raspberry Pi 3 as the central processing unit, coupled with a Pixhawk controller to manage navigation and control systems. Power is supplied by a battery bank, providing an operational runtime of up to 1.34 hours.

For navigation and operational control, the ROV is maneuvered via a joystick, offering precise control over its movement. The entire operation, including live camera feeds, is monitored in real-time through a Graphical User Interface (GUI) on the QGroundControl (QGC) platform. This setup allows for effective monitoring and task execution during underwater missions.

Additionally, the ROV is equipped with a CNN-based image recognition system, which enhances its capability to identify and classify underwater objects with high accuracy, even in challenging conditions such as high turbidity.

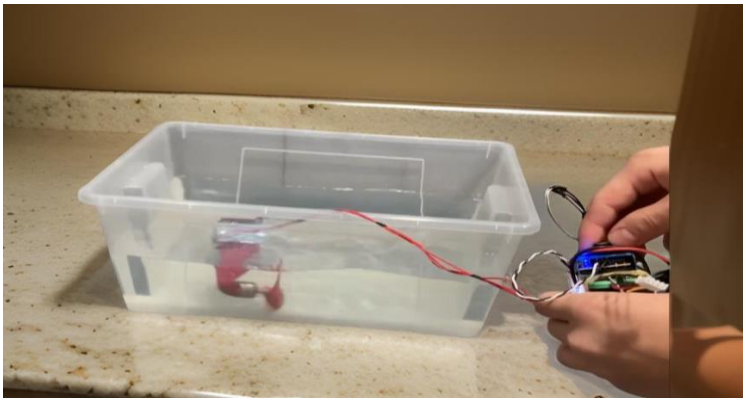
4. Experimental results

This section summarizes some of the experiments that have been conducted to test the performance of the Implemented ROV, approximate speed of the ROV, calculated to be around 3 m/s focusing on the motor operation test, temperature monitoring inside and outside the Implemented ROV, battery performance, and the Designed ROV's stability underwater, in addition to image capturing. A comparison of the Proposed ROV with two commercial systems will also be presented, focusing on the key features.

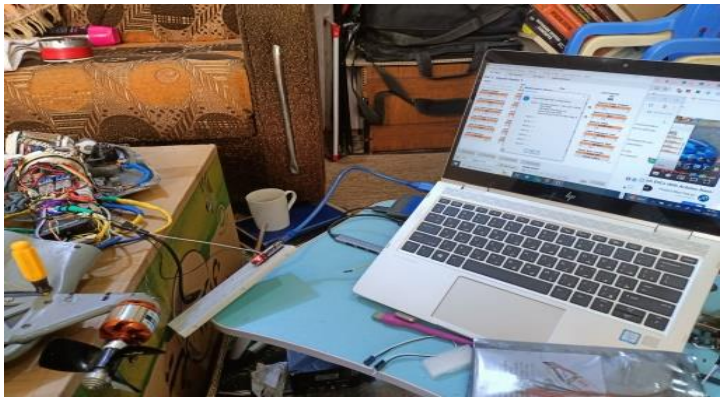
4.1 ROV performance evaluation

4.1.1 Motor operational test

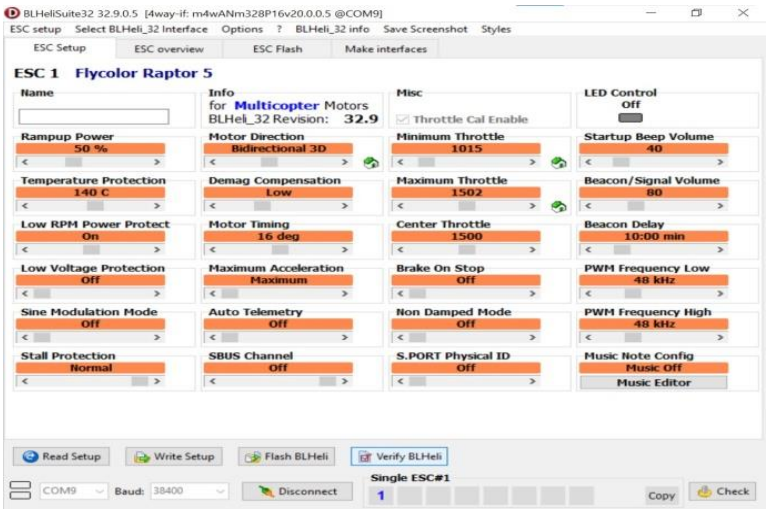
The brushless motor's operation and speed have been tested using ArduPilot development boards with a microcontroller (Arduino Nano). aiming to determine which is more suitable and efficient for generating the required PWM signals for the ESC. It was observed that the speed change of the motor that is initiated by the ArduPilot was abrupt. To address this issue, a program was developed and implemented on the microcontroller to generate PWM signals with higher resolution for smoother speed adjustments. Through experimentation, it was found that a time interval of 1 µs provided the most efficient results and optimal stability for the ROV underwater. Figure 17 illustrates an example of PWM signals generated by the microcontroller to control the brushless motors.



(a)



(b)



(c)

Figure 17. Illustrates the experimental setup: (a) shows the motor submerged in water and connected to a control device for speed adjustment via the ESC. (b) depicts PWM signal control of the motor speed using a laptop and Arduino Nano. (c) illustrates the use of BlHeli_32 Suite software for configuring the ESC on a PC

4.1.2 Temperature monitoring in ROV capsule

Figure 18 shows the integration of the MS5837 sensor within the ROV. This sensor has been used to measure the temperature inside the ROV. This data is transmitted to the Pixhawk microcontroller directly. A key aspect of our research involved conducting an experimental analysis to evaluate the temperature fluctuations inside the ROV. This analysis was prompted by the heat generated by the ESC during motor operations. An embedded algorithm was implemented in Pixhawk to get the temperature data from the MS5837 transducer. Additionally,

a dedicated algorithm was developed for the Raspberry Pi 3 to monitor both the internal and external temperatures of its chipset. This monitoring is crucial as the maximum operating temperature of the Raspberry Pi 3 must not exceed 65°C [47].

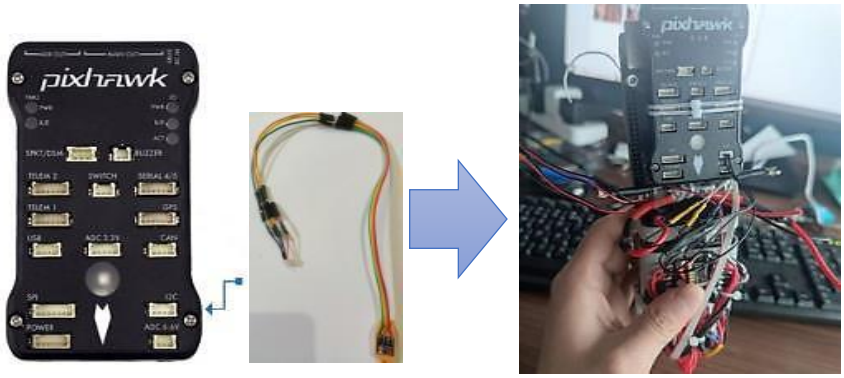


Figure 18. Connect the sensor to the Pixhawk through port I2C

The temperature sensor integrated into the ROV system is designed to monitor both the internal temperature of the ROV and the external water temperature. This dual monitoring is essential for assessing the ROV's operational environment and its internal thermal conditions. The external temperature readings reflect the surrounding water temperature, while the internal readings help in monitoring the heat generated by the ROV's electronic components.

As illustrated in Figure 19, these temperature readings are displayed on the ground station's laptop screen using the customized QGroundControl interface. The sensor's performance was evaluated under various conditions, including indoor and outdoor environments, as well as during submersion in water. The results of these tests are summarized in Table 3 and Figure 20.

Figure 19. Displays the internal and external temperatures of the Implemented ROV via the QGroundControl software at the bottom of the GUI

Table 3: Ground test results of temperature sensors and thermometers using the MS5837 sensor

		Temperature MS5837 sensor	
No.	**Test hours**	**Inside**	**Outside**
1	13.00	40.3° C	29.2° C
2	13.05	41.5° C	29.5° C
3	13.10	42.4° C	29.5° C
4	13.15	42.7° C	29.7° C
5	13.20	43.8° C	30.2° C
6	13.25	45.1° C	30.6° C
7	13.30	45.7° C	30.7° C
8	13.35	47.2° C	30.3° C
9	13.40	48.0° C	30.0° C
10	13.45	45.8° C	29. 8° C

In Table 3, the 'Inside' column represents the temperature measured inside the ROV hull, which reflects the temperature of the internal components. The 'Outside' column represents the temperature of the water surrounding the vehicle. These measurements illustrate how the operating environment affects the performance of the ROV.

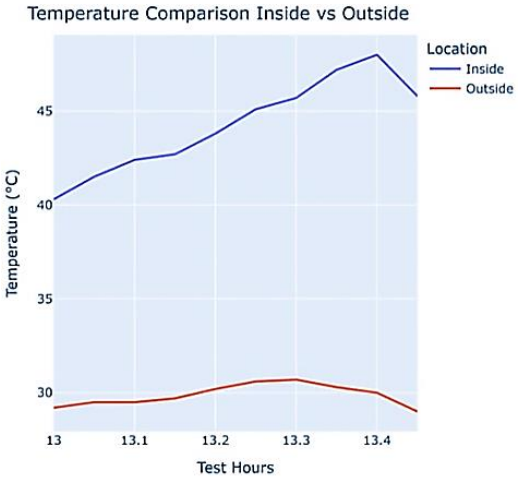


Figure 20. Temperature sensor and thermometer test results: comparison of internal and external measurements

4.1.3 Battery performance of the implemented ROV

The performance of the batteries during the Implemented ROV operation is illustrated in Figure 21. The battery bank operates at 11.1 V and distributes power to various system components as shown in Figure 5, including the Pixhawk controller, Raspberry

Pi, lights, and the ESC, which regulates the brushless motors. Under continuous operation at medium speed, the battery bank provides an operational duration of approximately 1.34 hours. However, this operational time can vary depending on the Implemented ROV's navigation patterns and the specific demands placed on the motors. Adjustments in speed or navigation style can significantly impact the overall battery life during the implemented ROV's missions**.**

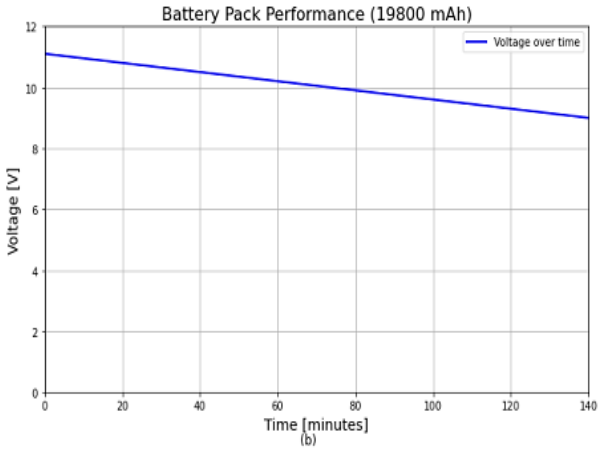


Figure 21. Performance of the 11.1 v battery pack (19800 mAh capacity) over time during underwater vehicle operation

4.1.4 ROV orientation and stability performance

The MPU6000 sensor is crucial in monitoring the orientation of the ROV, significantly contributing to its stabilization. The complementary filter integrates data from the accelerometer and gyroscope to provide a smooth output signal. To get highly accurate movement of the Implemented ROV for the short term, gyroscope data are primarily used due to their high precision and resistance to external forces. Over longer periods, accelerometer data are preferred because they do not drift.

In this experiment, the accelerometer sensitivity was set to 14,000 LSB/g, and the gyroscope sensitivity was set to 121°/s. Figure 22 illustrates the results of the stability tests conducted using the complementary filter. Figure 22a highlights the movements along the x-axis, while Figure 22b focuses on the y-axis. The data show that both the accelerometer and gyroscope produced stable outputs without significant noise or drift due to the complementary filter. This improved the quality of the images captured during its operation. Additionally, the complementary filter is straightforward and lightweight, making it ideal to be implemented in an embedded system such as the proposed system in this research.

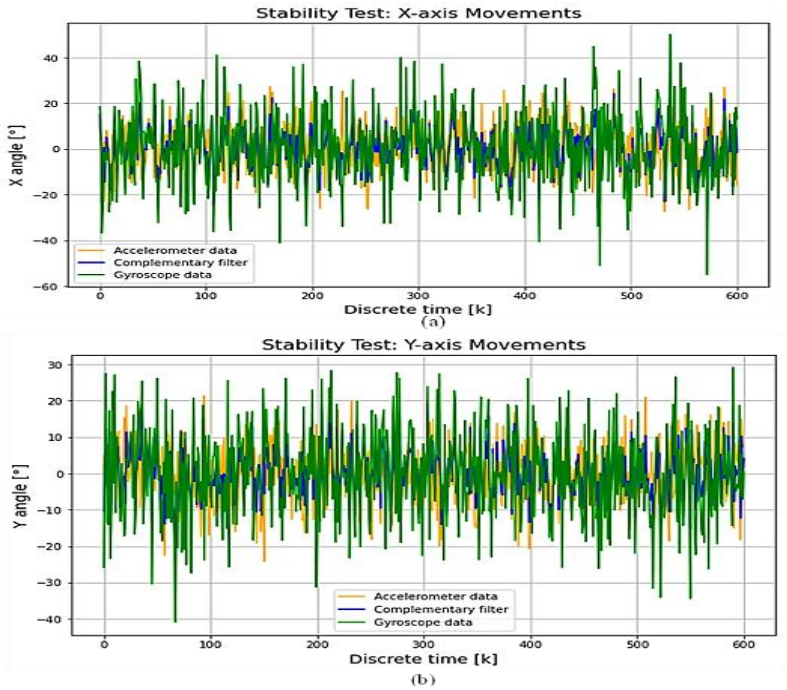


Figure 22. Dynamic stability evaluation of the implemented ROV using complementary filter, (a) X-axis movements; (b) Y-axis movements

Figure 23 shows the PID controller response. Figures 23a and 23b present the angles in the x-direction and y-direction, respectively. The results indicate that the actual trajectory closely matched the desired trajectory. Throughout the control process, the error remained minimal, nearly reaching zero, due to the use of the complementary filter, which effectively smoothed the signals. This signal smoothing greatly enhanced the implemented ROV stability and eventually improved the image capture quality.

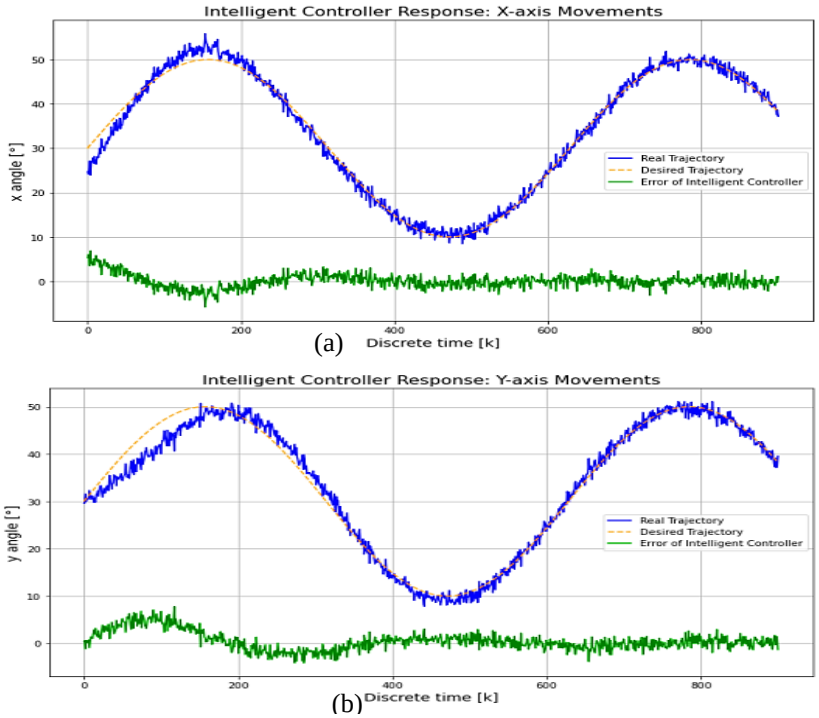


Figure 23. Performance of the smart PID controller response includes: (a) X-axis; (b) Y-axis

4.2 Design analysis of the proposed ROV

The flow details of the ROV assembly case, as depicted in Figures 24 and 25, can be analyzed using the AirShaper flow simulation plugin. The simulation results, as shown in these figures, indicate that the ROV is designed to effectively reduce hydrodynamic drag. The surface pressure distribution, illustrated by the color gradients in the figures, demonstrates that the designed ROV can move efficiently underwater. Specifically, areas of low pressure, represented by cooler colors, indicate that water flows smoothly around the vehicle, reducing drag and allowing for easier maneuvering in aquatic environments.

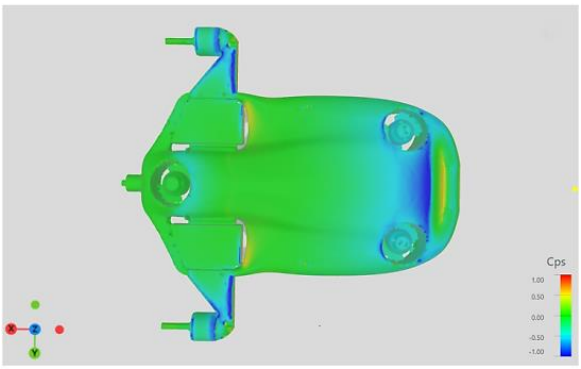


Figure 24. Surface pressure distribution on the designed ROV using AirShaper simulation

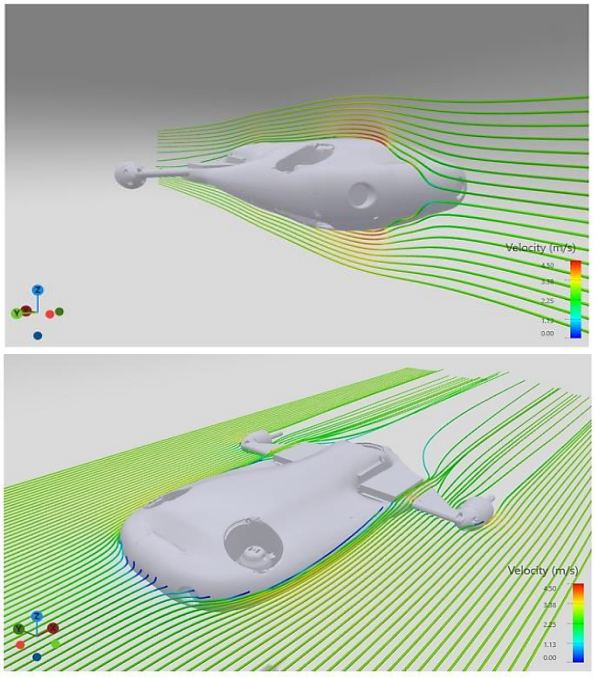


Figure 25. Velocity streamlines around designed ROV using AirShaper simulation

4.3 ROV testing in different environments

The proposed ROV was tested in different aquatic environments to evaluate its performance in various conditions. Initially, the proposed ROV was tested in a shallow, clean water environment inside a swimming pool, as shown in Figures (26 and 27). This test allowed us to observe the performance of the proposed ROV in a water environment free of impurities and water currents, which gave us a clear idea about its basic capabilities such as motion control and video capture.

Subsequently, additional experiments were conducted in a simulated river environment inside a water pool, where impurities such as kaolin clay were added to simulate natural water conditions with turbidity. This environment allowed us to evaluate the proposed ROV's ability to operate in more challenging conditions, focusing on its stability and image processing performance in the presence of water impurities.

The proposed ROV was designed to transmit data up to 100 meters using an Ethernet cable. However, since the required depth is not available in the Tigris River, the proposed ROV was tested at a limited depth in our study. This reflects the realistic testing conditions available, knowing that the proposed ROV is designed to operate at depths of up to 100 meters in deep water environments.

4.4 Submarine imagery analysis and tests

The experimental study aimed to evaluate the suitability of the Implemented ROV for various tasks. Before submerging the Implemented ROV in a swimming pool, a comprehensive test was conducted to prove its waterproof capabilities. Additionally, the Implemented ROV was operated for 50 minutes with the enclosure sealed to prevent overheating.

The enclosed swimming pool used in the experimental study measures (25 × 15 × 3) m (see Figure 26). Initially, the necessary objects for the Implemented ROV's tasks were placed in the pool. To prove the electrical enclosure's

insulation, the Implemented ROV was submerged underwater and secured with a rope without activating the power for ten minutes. Then, the pilot activated the power and navigated the vehicle using the GUI platform (QGC) with a joystick [48]. Throughout the experiment, the installed cameras enabled the pilot to observe the desired objects on the laptop screen, as shown in Figure 26.

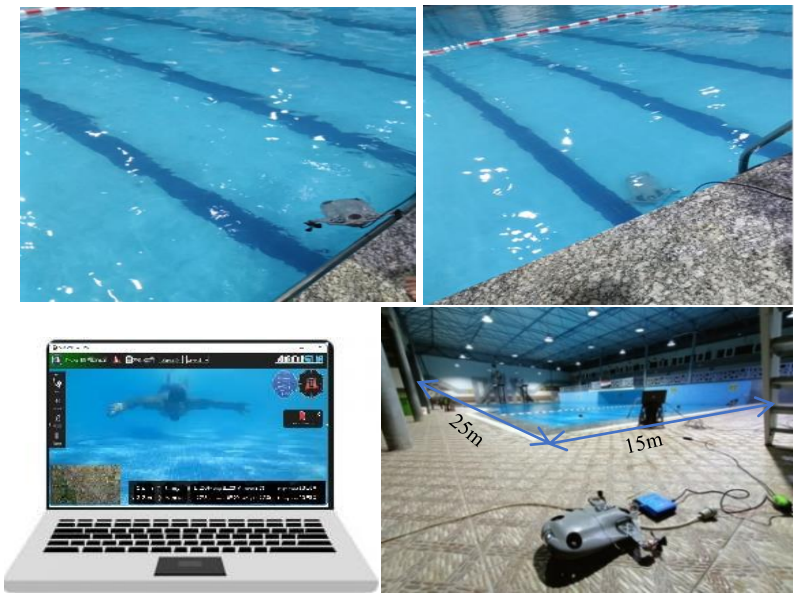


Figure 26. Experimental setup and dimensions of the swimming pool used for testing the implemented ROV

This section presents various images captured by the Implemented ROV in an underwater setting. For these tests, a camera was placed at the front of the implemented ROV; the video was recorded and transmitted to the ground station for image analysis. This setup allowed for the capture of images of objects like dolls, plants, and fish placed at the bottom of a 70 cm deep aquarium, as depicted in Figure 27. The images were captured in multiple stages, with kaolin clay added to generate turbidity, and a CNN-based underwater classification algorithm was used to detect objects. For instance, groups of three types of images are classified (Fish, Plant, and Human), highlighting the importance of the Implemented ROV in various underwater applications. As shown in Figure 23, the Implemented ROV captured video at up to 42 frames per second, from which images can be extracted, as illustrated in Figures 28-30. In all these instances, three frames can be observed as a realistic scenario, akin to the seabed with turbidity.

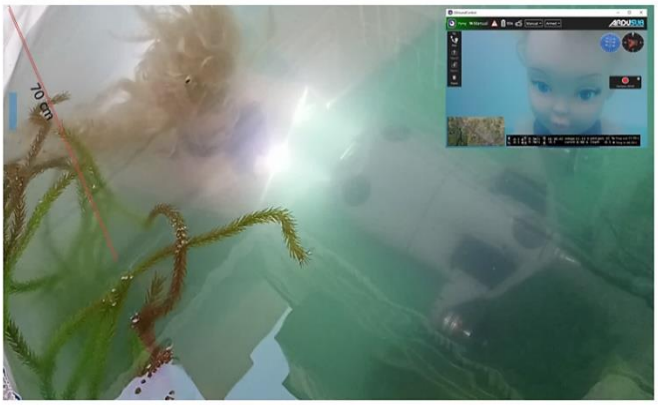


Figure 27. Simulated river environment in a water pond: image captured by the implemented ROV and transmitted to the ground station

Figure 28 illustrates images of dolls that resemble drowning victims, positioned at the bottom of a tank and captured in a sequence of images by the Implemented ROV. These dolls can be seen in different ways after varying concentrations of kaolin clay are added to the water to mimic the aquatic environment of the Tigris River. The differentiation of these scenarios is achieved through analyzing a series of underwater images containing various elements such as plants and fish, among others. These images are transmitted to the ground station to detect the presence of any objects at the bottom of the tank, as shown in Figure 27. This type of inspection holds significant scientific and practical value as it allows for underwater explorations without the need for diver intervention. This method serves as an effective alternative given the challenges divers face regarding limited visibility and difficulty in concentrating underwater, which can impede accurate perception.

Figure 28. Underwater images captured by the implemented ROV: dolls simulating drowning victims

Figure 29 presents a sequence of images depicting various segments of different aquatic plants utilized within a simulated pond setting. These images offer varied observations of the plant life, highlighting the diversity of species and the environmental conditions in which they thrive. This type of visual inspection, conducted remotely via the Implemented ROV, facilitates the detection and recognition of specific plants, a critical component in ecological assessments to distinguish and understand these aquatic species.

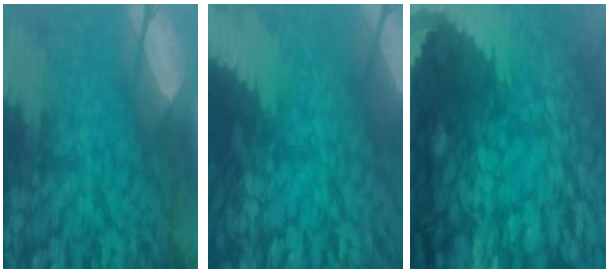


Figure 29. Aquatic Plant Life Observed by the Implemented ROV

Figure 30 presents sequences of images capturing a group of fish within a secluded glass tank, designed for detailed recognition and study. The images showcase various angles, particularly highlighting the noticeable water turbidity that has developed around the fish over time. This examination is highly valuable as it clarifies the living conditions of the fish and the potential risks these conditions might pose to the entire ecosystem.

Figure 30. Fish Populations Captured by the Implemented ROV

4.5 Feature comparison: proposed vs. commercial ROVs

The comparison of features between the proposed and commercial ROVs include; Functional Comparison: A comparison is made in Table 4 between the proposed ROVs and those available commercially. Although there are similarities in some areas, the proposed ROV stands out due to its utilization of parallel computing capabilities through Raspberry Pi with Pixhawk. This setup enables the simultaneous execution of multiple tasks and image capture at a rate of 42 frames per second

with a resolution of 800 × 640. Additionally, the integration of a complementary filter link to the IMU signals enhances the ROV performance. see Figure 9, leading to improved underwater image quality see Figures 28–30. Control of the vehicle is done through the open-source QGroundControl interface, which is compatible with various operating systems. The ROV's payload capacity allows for the addition of sensors, devices, and samples from a simulated freshwater river environment, with the proposed payload constituting 5% of the total vehicle weight. Furthermore, the design features of mechanical components and devices simplify maintenance and replacement, thereby boosting the self-reliance of remotely operated vehicle manufacturers in terms of technology.

Table 4: Comparison characteristics of the commercial ROV" vs the proposed ROV

Features	Blue ROV [49]	OpenROV [50]	Proposed
Architecture (open-source)	✓	✓	✓
Camera 1080p	✓	✓	✓
Autonomy time [h] (2–3)	✓	✓	✓
Communication (Ethernet)	✓	✓	✓
Maximum depth [m] (100)	✓	✓	✓
Processing type Unknown	✓	✓	
Parallel			✓
Frames per second (FPS) 30;	✓	✓	
42;			✓
Internet connectivity	✓	✓	✓
Controller algorithm (Smart PID)			✓
Remote control Joystick Graphic user (GUI)	✓	✓	✓
			✓
Dimensions (L×W×H) [cm]	45.7 × 33.8 × 25.4	8 × 20 × 40	40×39 × 9
Total weight [kg]	11.00	2.90	2.5
Total cost [USD]	2784.0	1695.0	600

5. Threats to validity

Like any experimental study, this research is subject to various threats to its validity, which must be acknowledged to ensure a comprehensive understanding of the results.

5.1 Internal validity

The experiments in this study were conducted in controlled environments, such as a swimming pool and a simulated river environment. While these settings allow for close monitoring and evaluation of the ROV's performance, they may not fully capture the complexities of natural underwater environments. Factors such as water currents, changing temperatures, and other environmental conditions could affect the ROV's performance differently in more realistic scenarios.

5.2 External validity

The results obtained from testing the ROV in a simulated swimming pool and river may not be fully generalizable to all underwater environments, particularly those experiencing extreme conditions. For example, while the ROV was tested to limited depths using a 30-meter Ethernet cable, its performance at greater depths, such as 100 meters, remains speculative and was not experimentally verified in this study. Additionally, due to the lack of deep-water environments in Iraq, testing was restricted, and the use of underwater drones was prohibited without security approvals.

5.3 Construct Validity

The capabilities of the ROV, including image processing using a CNN-based algorithm and real-time data transmission, were evaluated under specific conditions with controlled turbidity levels using kaolin clay. The effectiveness of the CNN algorithm in classifying objects in more diverse or natural underwater conditions with different types of turbidity or noise has not been fully validated. Furthermore, hardware components such as the Raspberry Pi 3 and Pixhawk were selected for their cost-effectiveness, but there may be limitations in processing power or accuracy that could impact the reliability of the overall system in more complex scenarios.

5.4 Data validity

The data collected during the experiments was accurate within the tested environments; however, the sensors used, such as pressure and temperature sensors, may behave differently in environments with higher pressure or more significant temperature variations. This could lead to inconsistencies in the data collected during actual underwater missions, potentially affecting the reliability of the ROV's operations.

6. Conclusions

The interactive, low-cost ROV design presented in this study demonstrates significant capabilities in underwater exploration and imaging. The ROV, housed in a 3D-printed enclosure with dimensions of 40.1741 cm × 39.1465 cm × 9.8729 cm and weighing 2.5 kg, operates efficiently at depths up to 100 meters, surpassing the 30-meter limit of traditional divers and accessing confined spaces. Key features include movement control across three axes, high-definition image capture, and integrated temperature sensing. The ROV is powered by five brushless motors managed by a smart PID controller, ensuring stability and smooth operation. The image capture and processing are further enhanced by a complementary filter and threading library, achieving up to 42 FPS on a Raspberry Pi 3. The open-source nature of the software and the flexibility of the mechanical design make the ROV adaptable to various applications, including underwater image capturing. Experimental results confirm the feasibility and effectiveness of the proposed ROV design.

7. Future Work

Future enhancements to the ROV could focus on increasing its dive depth beyond 100 meters to meet the demands of deep-sea exploration. Integrating advanced sensors, such as chemical detectors, upgraded sonar, and additional cameras, would expand its environmental monitoring capabilities. Improving autonomy through advanced algorithms and machine learning could allow the ROV to perform complex tasks like obstacle avoidance and object identification without human intervention. Prolonged battery life, achieved through new battery technologies or alternative energy sources, would enable longer missions. Enhancements to communication systems would ensure reliable data transmission in challenging underwater environments. A modular design approach could facilitate easier upgrades and customization, while an intuitive user interface, potentially incorporating augmented reality, would improve real-time navigation. Increasing onboard computing resources for real-time data processing would reduce reliance on surface-based systems. Additionally, it is essential to conduct environmental impact studies to minimize the ROV's effect on marine ecosystems. Finally, collaboration with multiple ROVs could enable more extensive and complex underwater operations.

References

[1] K. Vishwanath et al., "Study of Ocean Pattern's and Ocean Bed Using Applications of IOT," *2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE)*, 2024, pp. 1-6, doi: 10.1109/ic-ETITE58242.2024.10493788

[2] J.-K. Choi, T. Yokobiki, and K. Kawaguchi, "ROV-Based Automated Cable-Laying System: Application to DONET2 Installation," *IEEE J.*

Oceanic Eng., vol. 43, no. 3, pp. 665-676, Jul. 2018. DOI: 10.1109/JOE.2017.2735598.

[3] E. Anderlini, G.G. Parker, and G. Thomas, "Control of an ROV carrying an object," *Ocean Eng.*, vol. 165, pp. 307-318, 2018. DOI: 10.1016/j.oceaneng.2018.07.022.

[4] R. Capocci, E. Omerdic, G. Dooly, and D. Toal, "Fault-Tolerant Control for ROVs Using Control Reallocation and Power Isolation," *J. Mar. Sci. Eng.*, vol. 6, no. 2, p. 40, 2018. DOI: 10.3390/jmse6020040.

[5] X. Liu, F. Qi, W. Ye, K. Cheng, J. Guo, and R. Zheng, "Analysis and Modeling Methodologies for Heat Exchanges of Deep-Sea In Situ Spectroscopy Detection System Based on ROV," *Sensors*, vol. 18, no. 8, p. 2729, 2018. DOI: 10.3390/s18082729.

[6] S. Sivčev, M. Rossi, J. Coleman, E. Omerdić, G. Dooly, and D. Toal, "Collision Detection for Underwater ROV Manipulator Systems," *Sensors*, vol. 18, no. 4, p. 1117, 2018. DOI: 10.3390/s18041117.

[7] R. Capocci, G. Dooly, E. Omerdić, J. Coleman, T. Newe, and D. Toal, "Inspection-Class Remotely Operated Vehicles—A Review," *J. Mar. Sci. Eng.*, vol. 5, no. 1, p. 13, 2017. DOI: 10.3390/jmse5010013.

[8] D. Khojasteh and R. Kamali, "Design and Dynamic Study of an ROV with Application to Oil and Gas Industries of Persian Gulf," *Ocean Eng.*, vol. 136, pp. 18-30, 2017. DOI: 10.1016/j.oceaneng.2017.03.014.

[9] H.-T. Choi, J. Choi, Y. Lee, Y.S. Moon, and D.H. Kim, "New Concepts for Smart ROV to Increase Efficiency and Productivity," in *2015 IEEE Underwater Technology (UT)*, Chennai, India, 2015, pp. 1-4. DOI: 10.1109/UT.2015.7108257.

[10] H. Yao, H. Wang, Y. Li, Y. Wang, and C. Han, "Research on Unmanned Underwater Vehicle Threat Assessment," *IEEE Access*, vol. 7, pp. 11387-11396, 2019. DOI: 10.1109/ACCESS.2019.2891940.

[11] H. Berg and K.T. Hjelmervik, "Classification of Anti-Submarine Warfare Sonar Targets Using a Deep Neural Network," in *OCEANS 2018 MTS/IEEE Charleston*, Charleston, SC, USA, 2018, pp. 1-5. DOI: 10.1109/OCEANS.2018.8604847.

[12] G. Ferri, J. Bates, P. Stinco, A. Tesei, and K. LePage, "Autonomous Underwater Surveillance Networks: A Task Allocation Framework to Manage Cooperation," in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, Kobe, Japan, 2018, pp. 1-10. DOI: 10.1109/OCEANSKOBE.2018.8558813.

[13] G. Ferri, A. Munafò, and K.D. LePage, "An Autonomous Underwater Vehicle Data-Driven Control Strategy for Target Tracking," *IEEE J. Oceanic Eng.*, vol. 43, no. 2, pp. 323-343, Apr. 2018. DOI: 10.1109/JOE.2018.2797558.

[14] C. Yang, Y. Wang, and F. Yao, "Driving Performance of Underwater Long-Arm Hydraulic Manipulator System for Small Autonomous Underwater Vehicle and Its Positioning Accuracy," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 6, p. 1729881417747104, 2017. DOI: 10.1177/1729881417747104.

[15] G. Momber et al., "The Multidisciplinary Search for Underwater Archaeology in the Southern Red Sea," in *Geological Setting, Palaeoenvironment and Archaeology of the Red Sea*, N.M.A. Rasul and I.C.F. Stewart, Eds., Cham, Switzerland: Springer, 2019, pp. 605-628. ISBN: 978-3-319-98593-5.

[16] H. Munir, S. Hasnain, S. Khan, A. Cheok, H. Ali, and M. Shoaib, "Design and Fabrication of a Low-Cost Multi-Purpose Underwater Remotely Operated Vehicle," *Qeios*, 2023, doi: 10.32388/PWFY91.

[17] S. Heshmati-Alamdari et al., "A Robust Predictive Control Approach for Underwater Robotic Vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 4, pp. 1534-1543, Oct. 2019. DOI: 10.1109/TCST.2019.2939248.

[18] C. Yu et al., "Guidance-Error-Based Robust Fuzzy Adaptive Control for Bottom Following of a Flight-Style AUV with Saturated Actuator Dynamics," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1887-1899, May 2020. DOI: 10.1109/TCYB.2019.2891561.

[19] S. MahmoudZadeh, D.M.W. Powers, and R.B. Zadeh, "State-of-the-Art in UVs' Autonomous Mission Planning and Task Managing Approach," in *Autonomy and Unmanned Vehicles*, Singapore: Springer, 2019, pp. 17-30. ISBN: 978-3-030-19767-8.

[20] K.K. Bhardwaj et al., "Designing Energy-Efficient IoT-Based Intelligent Transport System: Need, Architecture, Characteristics, Challenges, and Applications," in *Energy Conservation for IoT Devices*, M. Mittal, S. Tanwar, B. Agarwal, et al., Eds., Singapore: Springer, 2019, pp. 209-233. ISBN: 978-3-030-29005-4.

[21] J. Lee et al., "Robust and Adaptive Dynamic Controller for Fully-Actuated Robots in Operational Space Under Uncertainties," *Auton. Robots*, vol. 43, no. 4, pp. 1023-1040, Oct. 2019. DOI: 10.1007/s10514-018-09708-4.

[22] E. Raugel et al., "Operational and Scientific Capabilities of Ariane, Ifremer's Hybrid ROV," in *OCEANS 2019-Marseille*, Marseille, France, Jun.

2019, pp. 1-7. DOI: 10.1109/OCEANS.2019.8867613.

[23] T. Elmokadem, M. Zribi, and K. Youcef-Toumi, "Control for Dynamic Positioning and Way-Point Tracking of Underactuated Autonomous Underwater Vehicles Using Sliding Mode Control," *J. Intell. Robot. Syst.*, vol. 95, no. 3-4, pp. 1113-1132, 2019. DOI: 10.1007/s10846-018-0900-2.

[24] S. Lack, E. Rentzow, and T. Jeinsch, "Experimental Parameter Identification for an Open-Frame ROV: Comparison of Towing Tank Tests and Open Water Self-Propelled Tests," *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 271-276, 2019. DOI: 10.1016/j.ifacol.2019.11.046.

[25] T. Matsuda, T. Maki, and T. Sakamaki, "Accurate and Efficient Seafloor Observations with Multiple Autonomous Underwater Vehicles: Theory and Experiments in a Hydrothermal Vent Field," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2333-2339, Jul. 2019. DOI: 10.1109/LRA.2019.2911291.

[26] N. Zendehdel, S.J. Sadati, and A.R. Noei, "Adaptive Robust Control for Trajectory Tracking of Autonomous Underwater Vehicles on Horizontal Plane," *J. AI Data Min.*, vol. 7, no. 3, pp. 475-486, 2019. DOI: 10.22044/jadm.2019.7247.1651.

[27] R.A. Armstrong, O. Pizarro, and C. Roman, "Underwater Robotic Technology for Imaging Mesophotic Coral Ecosystems," in *Mesophotic Coral Ecosystems*, Y. Loya, K.A. Puglise, and T.C.L. Bridge, Eds., Cham, Switzerland: Springer, 2019, pp. 973-988. ISBN: 978-3-030-24262-5.

[28] R. Hernández-Alvarado, L.G. García-Valdovinos, T. Salgado-Jiménez, A. Gómez-Espinosa, and F. Fonseca-Navarro, "Neural Network-Based Self-Tuning PID Control for Underwater Vehicles," *Sensors (Basel)*, vol. 16, no. 9, p. 1429, Sep. 2016. DOI: 10.3390/s16091429.

[29] B. He and A. Justice, "The Design of an Unmanned Aerial Vehicle Based on the ArduPilot," *Indian J. Sci. Technol.*, vol. 2, no. 1, pp. 12-15, 2009. DOI: 10.17485/ijst/2009/v2i1.23.

[30] "Ardupilot Overview," *Ardupilot Official Documentation*. [Online]. Available: https://ardupilot.org/. [Accessed: Aug. 2024].

[31] Z. Luo, X. Xiang, and Q. Zhang, "Autopilot System of Remotely Operated Vehicle Based on Ardupilot," in *Intelligent Robotics and Applications*, vol. 11579, Aug. 2019, pp. 240-250. DOI: 10.1007/978-3-030-27535-8_19.

[32] "Home Page - Pixhawk," 2014. [Online]. Available: http://pixhawk.org/. [Accessed: Aug. 2024].

[33] S.A.S. Alkadhim, "Communicating with Raspberry Pi via MAVLink," Xi'an Jiaotong University (XJTU) - State Key Laboratory of Electrical Insulation and Power Equipment, Jan. 18, 2019. [Online]. Available: https://ssrn.com/abstract=3318130.

[34] T. Fossen, *Guidance and Control of Ocean Vehicles*, New York, NY, USA: John Wiley & Sons, Inc., 1994.

[35] A. Setyawan and Mashoedah, "Remotely Operated Underwater Vehicle (ROV) Stabilization with Artificial Neural Networks (ANN)," in *J. Phys.: Conf. Ser.*, vol. 1833, no. 1, Art. no. 012068, 2021. DOI: 10.1088/1742-6596/1833/1/012068.

[36] O.A. Aguirre-Castro, E. Inzunza-González, E.E. García-Guerrero, E. Tlelo-Cuautle, O.R. López-Bonilla, J.E. Olguín-Tiznado, and J. Cárdenas-Valdez, "Design and Construction of an ROV for Underwater Exploration," *Sensors*, vol. 19, no. 24, Art. no. 5387, 2019. DOI: 10.3390/s19245387.

[37] M. N. Öz, S. Budak, E. Kurnaz, and A. Durdu, "Orientation Determination in IMU Sensor with Complementary Filter," *Turkish Journal of Forecasting*, vol. 6, no. 1, Jun. 2022. DOI: 10.34110/forecasting.1126184.

[38] Y. He, S. Guo, L. Shi, H. Xing, Z. Chen, and S. Su, "Design and Implementation of a Self-Tuning Control Method for the Underwater Spherical Robot," in *2017 IEEE Int. Conf. on Mechatronics and Automation (ICMA)*, Takamatsu, Japan, 2017, pp. 632-637. DOI: 10.1109/ICMA.2017.8015890.

[39] N. Aula and N. Abd, "Adaptive Inverse Neural Network Based DC Motor Speed and Position Control Using FPGA," *Diyala J. Eng. Sci.*, vol. 11, no. 3, pp. 71-78, Sep. 2018. DOI: 10.24237/djes.2018.11311.

[40] BlueRobotics, "T200 Thruster." [Online]. Available: https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/. [Accessed: Mar. 20, 2022].

[41] P. Mayer, M. Magno, and L. Benini, "Self-Sustaining Acoustic Sensor With Programmable Pattern Recognition for Underwater Monitoring," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 7, pp. 2346-2355, Jul. 2019. DOI: 10.1109/TIM.2018.2890187.

[42] R.D. Christ and R.L. Wernli Sr, *The ROV Manual: A User Guide for Remotely Operated Vehicles*, Oxford, UK: Butterworth-Heinemann, 2013. ISBN: 978-0-08-098291-7.

[43] P. M. Gerhart, A. L. Gerhart, and J. I. Hochstein, Fundamentals of Fluid Mechanics, 8th ed., Wiley, December 2018, ISBN: 978-1-119-54799-0.

[44] A. Rasheed, R. Hagem, A. Khidhir, and O. Hazim, "Underwater Robotics: Principles, Components, Modeling, and Control," *Al-Rafidain Eng. J.*, vol. 29, no. 1, pp. 154-176, 2024. DOI: 10.33899/arej.2024.182561.

[45] E. G. Haug, "Planck Scale Fluid Mechanics: Measuring the Planck Length from Fluid Mechanics Independent of G," Open Journal of Fluid Dynamics, vol. 13, no. 5, pp. 225-236, Dec. 2023. DOI: 10.4236/ojfd.2023.135019

[46] R. Capocci, G. Dooly, E. Omerdić, J. Coleman, T. Newe, and D. Toal, "Inspection-Class Remotely Operated Vehicles—A Review," *J. Mar. Sci. Eng.*, vol. 5, no. 1, p. 13, 2017. DOI: 10.3390/jmse5010013.

[47] Raspberry Pi Foundation, "Raspberry Pi Documentation," available online: https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#frequency-management-and-thermal-control. [Accessed: 26-Aug-2024].

[48] K. Sh. Rejab, "Wireless Mobile Robotic Arm Controlled by PS2 Joystick Based on Microcontroller," *Diyala J. Eng. Sci.*, vol. 10, no. 3, pp. 25-34, Sep. 2017. DOI: 10.24237/djes.2017.10304.

[49] Blue Robotics, "BlueROV2." [Online]. Available: https://pdf.nauticexpo.com/pdf/bluerobotics/bluerov2/69617-112501.html. [Accessed: Jun. 26, 2024].

[50] Scubanautics, "Open ROV Trident Underwater Remote Operated Vehicle." [Online]. Available: https://scubanautics.com.au/product/open-rov-trident-under-water-remote-operated-vehicle/.