Autonomous Underwater Vehicle Design and Development: Methodology and Performance Evaluation

Submitted: 23rd July 2023; accepted: 30th November 2023

Ismail Bogrekci, Pinar Demircioglu, Goktug Ozer

DOI: 10.14313/JAMRIS/4-2024/30

Abstract:
This study focused on the development of an Autonomous Underwater Vehicle (AUV) across four key domains: Mechanical Design, Software, Electronics, and Security. The Mechanical Design phase involved utilizing computer-aided drawing programs to create the AUV model. Important design considerations encompass manufacturability, cost, power, weight, and durability. Comparisons with nominal values of existing market products validated the precision of the produced designs. This study placed particular emphasis on the optimization of model weight, with a focus on ensuring the AUV's efficiency, lightness, and exceptional maneuverability within underwater environments. The Software stage entailed the development of AUV software to enable sensitive and effective vehicle operations. Efficient functioning, devoid of errors or complications, was imperative to ensure optimal autonomous driving and operational capabilities. Image processing algorithms were incorporated into the software to maintain high accuracy and provide dimensional and geometric information from targeted areas. Furthermore, the software phase involved the development of an image processing algorithm based on color analysis, further augmenting accuracy. The selection of electronic components for the AUV was also a vital consideration, alongside ensuring safety measures at every stage of the UAV's development.

Keywords: *autonomous underwater, image process, 3D design and production, self-controlled*

1. Introduction

Autonomous underwater vehicles (AUVs) are advanced systems that integrate sensor detection, information fusion, and vehicle control technologies. A key application of AUVs is underwater multi-target search. To improve search efficiency, accuracy, and speed, a team of AUVs can collaborate. This approach enhances system performance, reconfigurability, and flexibility while reducing costs and mission duration. Multi-AUV systems find utility in gathering environmental gradient data and conducting habitat mapping across diverse fields [1].

Two-thirds of our planet's surface is covered by water in the form of oceans, seas, and rivers, which remains largely unexplored. This study is motivated by this fact. With underwater communications in the context of the Internet of Things (IoT), an intelligent interconnection can be established between nodes and vehicles, providing remarkable services for underwater applications. Thus, the concept of the Internet of Underwater Things (IoUT) encompasses communication for both the Underwater Wireless Sensor Network (UWSN) and AUVs facilitated by Internet services [2]. The IoUT network enables the deployment of various underwater applications, including environmental monitoring, surveying, military purposes, and more [3].

Scientific AUVs have diverse applications, including serving as a test bed for new technologies [4]; collecting bathymetric data and mapping the seafloor [5]; conducting archaeological surveys [6]; monitoring the environment [7]; studying physical, biological, and chemical processes [8]; performing geological surveys [9]; and exploring extreme environments [10].

Commercial AUVs are utilized in several applications, including underwater infrastructure inspection [11], in the oil and gas industry [12], and in repair and maintenance operations [13].

Military AUVs serve a range of applications, including antisubmarine warfare [14], mine countermeasures [15], target localization [16], and search and rescue operations [17].

In the bibliometric analysis conducted for this study, three-field plots and co-occurrence networks of author keywords were employed to yield valuable insights into the interconnections and patterns within academic literature. The three-field plot (Fig. 1) provides a visual representation of affiliations, keywords, and co-authors' countries, where the height of the boxes serves as an indicator of research output. Furthermore, co-occurrence networks (Fig. 2) visually depict relationships between keywords in research publications, uncovering thematic interplay. These analytical tools are critical for grasping academic literature and research patterns, providing a detailed understanding of collaborative networks and current trends. For instance, Figure 1 illustrates that the United States, Spain, China, and Japan prominently feature in the AUV domain, while Figure 2 focuses on 50 nodes associated with AUV keywords.

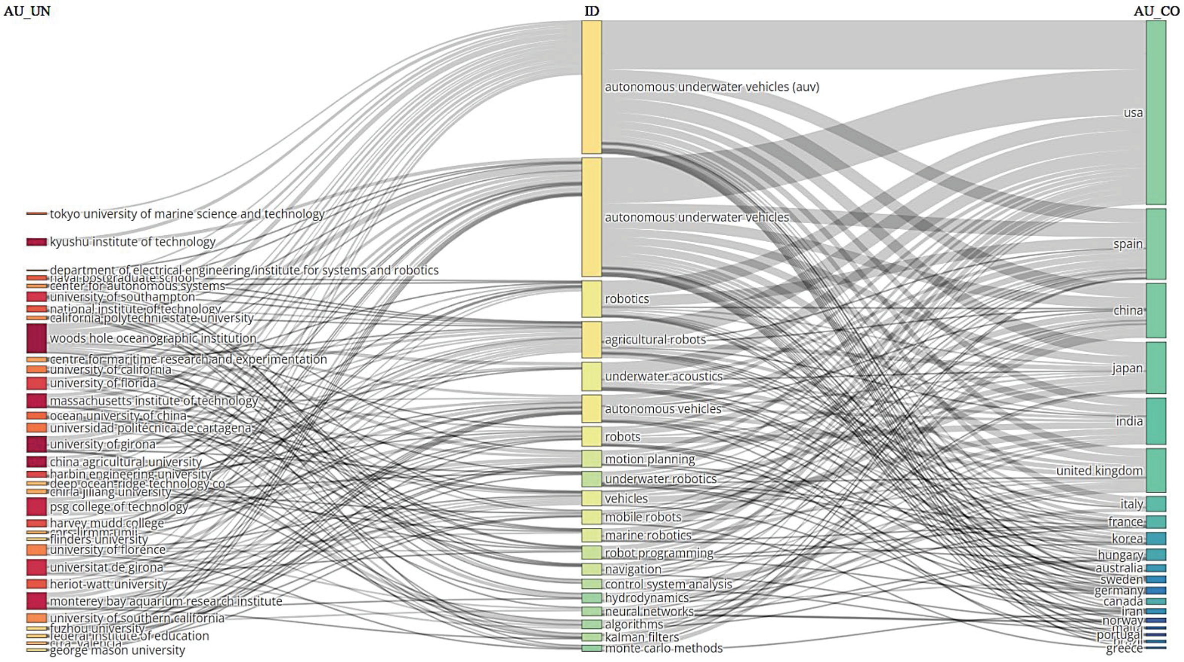


Figure 1. Three-field plot analysis (AU_UN (Authors' University)—DE (Keywords)—AU_CO (Authors' Country)

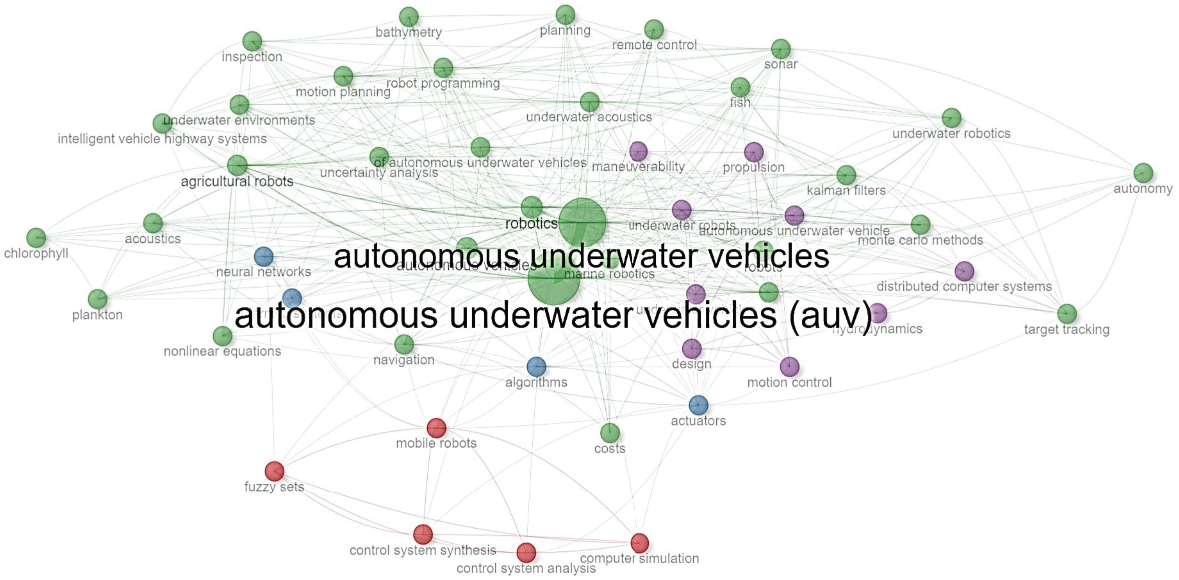


Figure 2. Co-occurrence network of author keywords

The primary aim of this study was to advance the evolution of an AUV by conducting a thorough investigation into four key domains: Mechanical Design, Software, Electronics, and Security. The pivotal objective was to guarantee the AUV's efficiency, achieve a lightweight structural composition, and facilitate superior maneuverability within underwater environments. Through a concentrated examination of these foundational domains, this research endeavors to augment the overall performance and operational capabilities of the AUV.

2. Materials and Methods

2.1. Mechanical Design Process Used in the Development of the AUV

As a starting point, the CAD program SolidWorks was selected to create and visually represent designs on a computer. During the design process for the mechanical body, the main criteria for selection are cost, weight, symmetry of the main body, security, time savings, and ease of manufacturing. Designing for underwater conditions presents additional challenges compared to designing for land-based applications. Factors such as vehicle stabilization, system steadiness, communication, and pressure parameters are more difficult to adjust and finalize in the design.

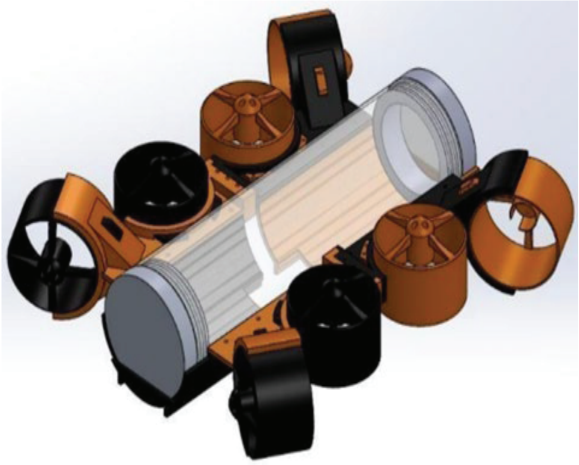


Figure 3. Basic dimensions (in mm) of the autonomous vehicle

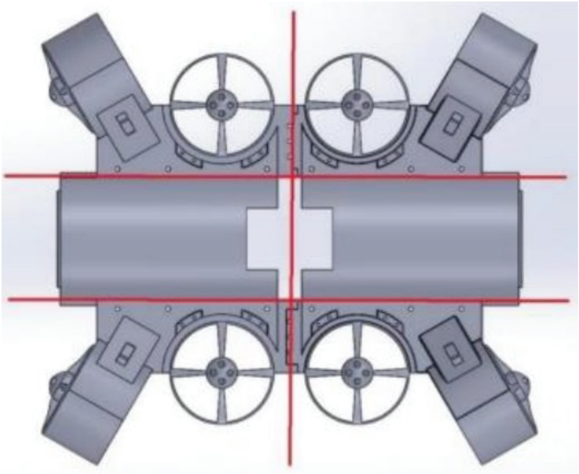


Figure 4. Main chassis

To address these parameters, various academic studies, drawing programs, and analysis tools were utilized. The overall mechanical design encompasses the main chassis, thrusters, propellers, a leak-proof plexiglass tube, and tube-stabilizing cuffs. Following thorough work based on the aforementioned criteria, the final design of the vehicle was achieved, as shown in Figure 3.

In streamlining the manufacturing process, the primary chassis of the Autonomous Underwater Vehicle (AUV) was meticulously designed to undergo 3D printing, comprising six distinct components. This modular approach facilitates ease of production, allowing each component to be printed separately and subsequently assembled to form the integral chassis structure. For visual clarity and reference, Figure 4 provides a detailed depiction of the six main constituent parts that collectively constitute the AUV's chassis. This method not only enhances the efficiency of the manufacturing process but also offers flexibility and scalability in adapting the design to specific requirements and modifications.

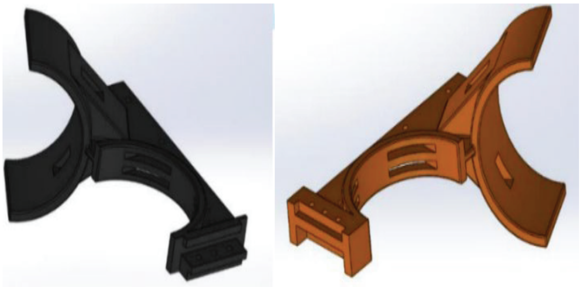


Figure 5. Male–female chassis parts

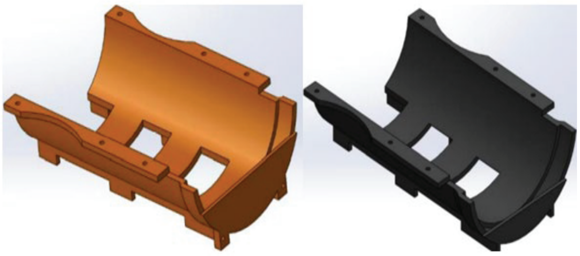


Figure 6. Lower chassis parts

The holders tasked with stabilizing the thrusters on the main chassis underwent rigorous stress testing and were purposefully designed to withstand the considerable forces encountered in underwater environments. In particular, the half-circle segment of the chassis designated for motor placement underwent reinforcement to mitigate the risk of potential breakage.

Utilizing advanced 3D printing technology, the sections where the thrusters will be positioned were intricately crafted. These sections feature twin female and male connections, augmented with M4 bolts to ensure a robust and secure attachment mechanism. The design and structural details of these female-male chassis parts are vividly illustrated in Figure 5, providing a visual reference for the meticulous engineering and durability considerations incorporated into the AUV's thruster stabilization system.

To accommodate the installation of a camera inside the vehicle, a gap was designed on the lower parts of the chassis. The assembly of the chassis is finalized using M4 bolts to connect the female-male parts. In order to ensure stability while on the ground, small stands in strips were added to the lower section of the chassis. Please refer to Figure 6 for a visual representation of the lower part of the chassis.

In order to maintain weight balance and symmetry, while enabling smooth forward-backward and right-left movement, four propulsion motors were strategically placed at the four corners of the main chassis, angled at 45 degrees. Additionally, to achieve vertical movement, another four propulsion motors were symmetrically positioned at the center of the chassis. This configuration ensures both aesthetic appeal and optimal weight distribution. Consequently, a total of eight (8) Brushless Direct Current (BLDC) motors are utilized in the vehicle.

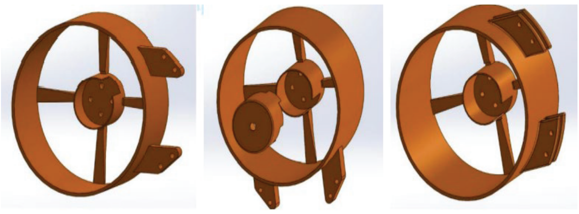


Figure 7. Thrusters

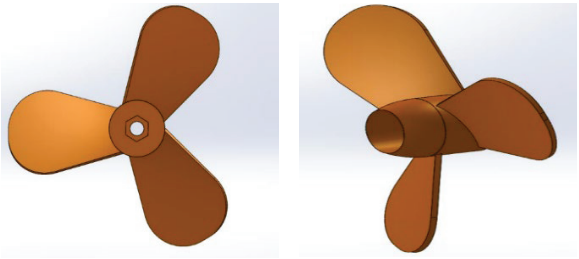


Figure 8. Propeller

The motors responsible for generating vertical movement are secured using M3 bolts, while the remaining motors, which facilitate horizontal movement, are tightly fixed using pins. Each motor is equipped with a lid cover for its housing. These covers not only provide protection but also enable the motors to be sealed with epoxy at their connection points, ensuring they are watertight. This safeguard allows the motors to operate safely underwater for extended periods of time (Fig. 7).

The propellers, as shown in Figure 8, designed in SolidWorks and 3D printed, underwent testing via CFD analysis and real-world conditions. The observation revealed that a single bolt was insufficient for secure attachment, prompting the addition of another strategically positioned bolt. CFD analysis highlighted substantial thrust, primarily due to a high pitch angle, but posed a challenge of the control card overheating. This underscores the importance of adopting a balanced optimization approach that addresses not only propeller performance enhancements but also the critical aspect of thermal management. Such an approach highlights the interconnected considerations integral to the processes of design, analysis, and practical application, ensuring a comprehensive and effective solution to the challenges encountered.

To prevent any contact between the electronic components and water, a robust solution was implemented. All electronic components are securely housed within a leak-proof plexiglass tube. This tube features lids at both the front and back ends. The use of a leak-proof plexiglass tube to house the electronic components is a robust solution for preventing contact between the components and water. By enclosing the electronics within the tube, one creates a physical barrier that shields them from any potential water damage or moisture intrusion.

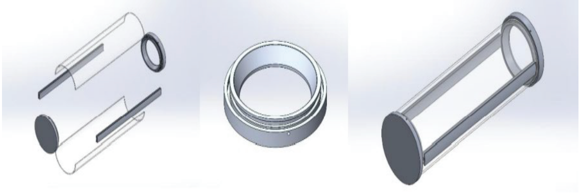


Figure 9. Leakproof plexiglass tube

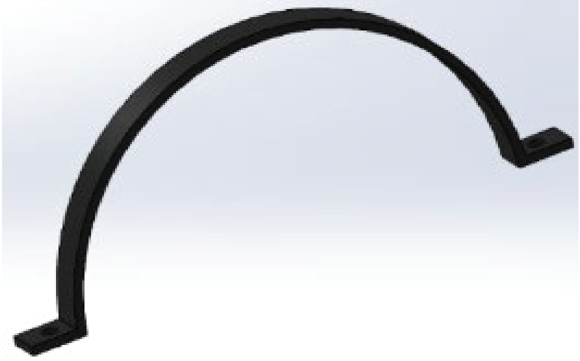


Figure 10. Tube fixing cuff design

The frontal lid incorporates a circular plexiglass material with an inner diameter of 114mm, allowing for camera viewing. On the rear lid, there are leak-proof connector holes specifically designed for the motor cables to pass through. To maintain the integrity of the system and prevent any leakage, the motor cables passing through the connectors are enclosed within a separate tube that is filled with epoxy. This additional tube measures 120mm in diameter and 400mm in length. Both the lids of this tube were reinforced with O-rings to ensure a completely leak-proof system.

Figure 9 illustrates the CAD assembly and lid design of the plexiglass tube, showcasing the meticulous attention given to maintaining a watertight enclosure for the electronic components.

The stability of the plexiglass tube is of utmost importance, and it is crucial to securely fasten the tube to the chassis. To address both the vibrations generated by the motors and the fluid movements caused by water, supporting fixing cuffs were strategically positioned and connected to the chassis using M4 bolts. The use of supporting fixing cuffs and M4 bolts to secure the plexiglass tube to the chassis is an effective approach to ensure stability. The vibrations generated by the motors and the fluid movements caused by water can potentially create stress and strain on the tube, which may lead to instability or even damage if not properly addressed. Figure 10 presents the design of the tube fixing cuffs, offering a visual representation of their configuration. The "Tube Fixing Cuff Design" refers to the specific engineering and structural arrangement created to securely fasten the plexiglass tube to the chassis of the vehicle, thereby ensuring stability during its operation.

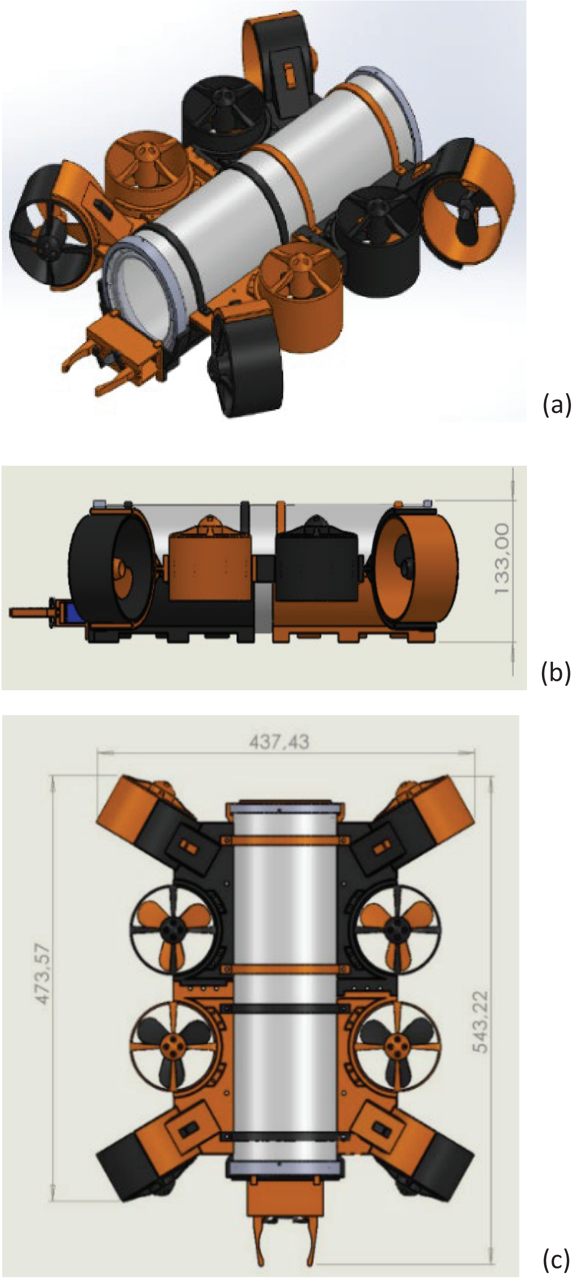


Figure 11. Isometric (a), frontal (b), and top (c) views of the design

Furthermore, Figures 11a–c present the isometric (a), frontal (b), and top (c) views of the design, providing a comprehensive overview of the finalized vehicle design and its overall appearance and features.

In order to enhance the AUV's functionality and enable it to perform additional tasks, a gripper arm design was meticulously crafted. This component serves as a mechanical moving part that can securely grasp and hold objects without exerting excessive pressure that could potentially damage them. The development process involved multiple iterations and redesigns, as the gripper arm proved to be a more complex and challenging component to manufacture.

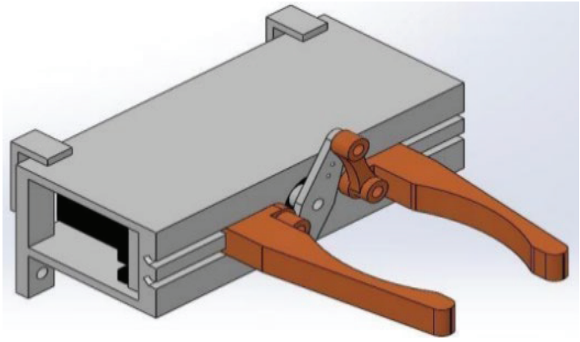


Figure 12. Frontal view of the gripper design

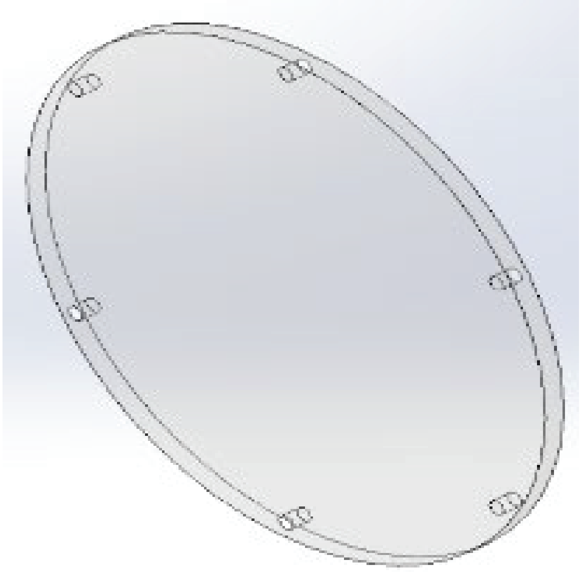


Figure 13. Acrylic circle

The final design of the gripper arm (Fig. 12) was intentionally kept as simple as possible to fulfill its intended task without introducing unnecessary complexity to other aspects of the AUV. To facilitate the gripping motion, a waterproof servo motor was utilized. The gripper itself is positioned at the front of the sub-frame and is capable of achieving an open diameter of up to 80mm and a closed diameter of up to 25mm, allowing it to effectively carry out its objective.

The plexiglass tube serves as a protective enclosure, effectively containing all the electronic components and ensuring a leakproof environment. Within this tube, the heaviest component of the system, the Lithium-Polymer (Li-Po) battery, is securely housed.

To enable the camera placed inside the acrylic tube to capture images, a window is provided to connect it with the outside world. This is achieved by incorporating an acrylic circle on the frontal side of the tube, which allows the camera to perform its function effectively. The acrylic circle (Fig. 13) has a diameter of 114mm and a thickness of 3mm, providing a clear and transparent view for the camera's imaging capabilities.

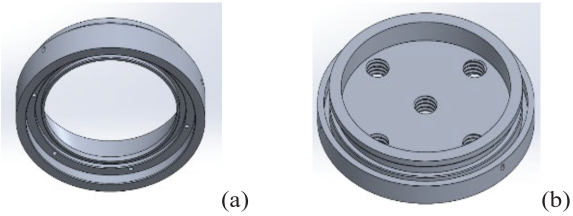


Figure 14. Frontal (a) and rear (b) side lids

Figure 15. O-rings

The tubes situated on both the frontal (Fig. 14a) and rear (Fig. 14b) sides of the acrylic tube are crafted from aluminum material. For the frontal lids, the design incorporates mortises to accommodate O-rings (Fig. 15), ensuring a secure and leakproof seal. In contrast, the rear lids are equipped with five pass holes to allow the cables from outside to pass through. Once the cable passages are completed, the lids are sealed using epoxy and brass material to further enhance the leakproof integrity of the system.

2.2. Materials Used in the Development of AUV

2.2.1. PLA+ Filament

The filament employed in 3D printing for producing materials is Polyactic Acid+ (PLA+). This organic biopolymer and thermoplastic exhibit a density of 1.24 g/cm^3 and a tensile strength of 43MPa. The optimal printing temperature ranges from 190°C to 220°C, while the recommended bed temperature falls between 60°C and 80°C.

Compared to other manufacturing options, 3D printing using PLA+ filament proves to be cost-effective. Once the temperature and printing settings are appropriately configured, it allows for printing without encountering significant issues. Moreover, PLA filament is easier to work with compared to other filament types, offering the added advantage of a brighter appearance. This contributes to an aesthetically pleasing finished product. Additionally, PLA is considered non-harmful to humans.

Table 1. Materials used for mechanical parts [18]

Mechanical Part Name	Material Used	Amounts
Male Chassis Part	PLA+	2
Female Chassis Part	PLA+	2
Rear Sub-frame	PLA+	1
Frontal Sub-frame	PLA+	1
Forward-Reverse Thruster	PLA+	4
Vertical Thruster	PLA+	4
Motor Covers	PLA+	8
Propeller	PLA+	8
Tube Fixing Cuff	PLA+	3
Tube Rear Lid	Aluminum	1
Tube Frontal Lid	Aluminum	1
O-Ring	Plastic	3
Tube	Acrylic	1
Front Cover Glass	Acrylic	1

2.2.2. Aluminum

The material chosen for the lids of the tubes is aluminum due to its favorable properties and suitability for the intended purpose. Aluminum was selected because of its ease of processing, low manufacturing tolerance, and excellent resistance to corrosion, which is particularly important since the lids will be predominantly underwater. The surface of aluminum naturally forms a thin yet durable oxide film, providing enhanced corrosion resistance. Additionally, aluminum is non-magnetic, possesses good electrical conductivity, and has a density of 2.70 g/cm^3. It has a relatively low melting temperature of 660.32°C and a tensile strength of 49 MPa.

2.2.3. Materials Used for Mechanical Parts

The design process for the physical parts was focused on minimizing the overall weight of the vehicle, ensuring its maneuverability on water. The final design achieved a total weight of 3357g, accompanied by a volume of 9,002,562.75mm^3. The chassis dimensions measure 473.57mm in length, 437.43mm in width, and 133mm in height. Below is a table listing all the mechanical parts incorporated into the vehicle:

2.3. Methods Used in the Development of AUV

2.3.1. 3D Printing

All the parts, except for the front cover glass, tube lids, and O-rings, were manufactured using a 3D printer. The choice of a 3D printer was driven by the convenience it offers in the manufacturing process. Additionally, the iterative nature of design changes, which required minimal to no modifications in the manufacturing process, and the overall cost-effectiveness of 3D printing were key factors in the decision.

To ensure robustness and prevent any issues during bolt tightening, all the components were printed with a high infill percentage of 85%. This increased infill percentage contributed to the overall strength and structural integrity of the printed parts.

2.3.2. Machining

The aluminum tube lids, utilized on both the frontal and rear sides of the AUV to ensure a leak-proof seal for the acrylic tube, were manufactured through machining processes. This method was chosen as the most appropriate means of fabrication, as it enables precise processing while maintaining a low tolerance rate.

2.4. Manufacturing Methods Used in the Development of AUV

After evaluating the diverse range of functions expected from the vehicle, including image processing, motor control, sensor integration, data processing, and automation, power consumption calculations determined that the most suitable power supply option for the system was the Lithium Polymer Profuse Battery 6S 25C (22.2V 22000mAh). This Li-Po battery provides sufficient power to run all electronic components without encountering any power shortages.

To ensure smooth and efficient movement underwater, the selected motors and electronic speed controllers (ESCs) were carefully chosen to work in harmony with the Li-Po battery. The motors are 6S compatible with a rotation speed of 30,000 RPM and require approximately 30A of current, while the ESCs have a current resistance of 40A.

For data processing, which includes information from sensors such as ultrasonic, depth, pressure, and the camera for image capture, a Raspberry Pi 4 was incorporated. Additionally, a Pixhawk PX4 flight control card was employed for vehicle automation. To prevent overheating, a 5V voltage regulator was chosen to power the sensors and camera instead of directly relying on the Raspberry Pi 4 and Pixhawk PX4.

Safety measures were also implemented, including an emergency stop button and a power distribution card capable of handling up to 400A, serving as safeguards for unexpected high voltage or overheating situations.

The sensors and camera are connected to the Raspberry Pi 4 which has Broadcom BCM 2711 central processing unit (CPU) with a 4-core advanced RISC machines (ARM) architecture Cortex – A72 to process the data received and give the vehicle autonomous abilities. Having a CPU of 1.5 GHz work frequency was selected as it is %10 faster than the previous model.

Being a highly advanced flight control card that works for the automation of the vehicle itself, the Pixhawk is widely used both in air vehicles such as multirotor planes and helicopters while also being used in remote-controlled vehicles. It was selected as an ideal flight control card with a 32-bit processor. The Pixhawk receives the necessary data to properly control the speed of the motors. The connection between the Raspberry Pi 4 and the Pixhawk was made possible with a telemetry device that secures a universal asynchronous receiver/transmitter (UART) communication. The power required to run the flight control card is received from the Raspberry Pi 4.

An analog power module was integrated to provide a stable power supply to the Pixhawk flight control card and the mini-computer Raspberry Pi 4 which also supports the measurement of the battery voltage and the current consumption. The preferred power module has a maximum input of 30V 90A and a maximum output of 5.3V 3A.

To provide the connection between the motors and the Pixhawk and to control the motors themselves, eight 40A ESCs compatible with 2-6S power input were used. With these ESCs, control of the motors was achieved with the help of the signals coming from the flight control card.

To prevent damage to the electrical components of the system from the high voltage coming from the power supply, a process control block (PCB) regulator was integrated into the system to regulate the incoming voltage.

To capture and process the images during the work activity of the vehicle, a 5MP camera with a fish eye lens that is compatible with the Raspberry Pi 4 Model B was selected.

To detect the position of objects underwater and avoid a collision with them, five units of leak-proof ultrasonic distance sensors were placed on the edges and the bottom of the chassis.

For the purpose of measuring the distance from the surface and the depth of the vehicle itself, and to know its positional information, depth and pressure sensors were included in the system.

To check the electrical components for temperature control and cut the power to the system in the case of overheating, a temperature and humidity sensor was included in the system. The DHT21 Temperature-Humidity sensor was chosen as it is highly reliable and can work stably even in long-term working conditions, in addition to having a fast response rate as it has an 8-bit microprocessor.

The board itself was handmade with copper cables, where the plate with black colored cables represents "–" while the plate with the red colored cables represents "+". It is a crucial part of the system as it allows the power coming from the source to reach every part of the system.

In case of a system overload emergency, a current sensor was integrated to proactively shut down the electricity supply before external intervention. Additionally, a leakage emergency stop button was added to the chassis for immediate deactivation in unforeseen circumstances, enhancing device safety protocols. Figure 16 provides a visual representation of the final placement of the electronic components in the acrylic tube, aiding operational understanding and facilitating potential troubleshooting scenarios.

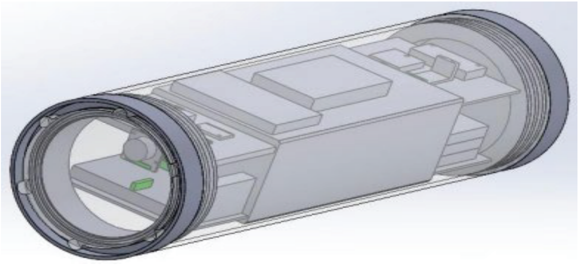


Figure 16. Image of electronic components inside the acrylic tube

2.5. Electronical Design Process Used in the Development of AUV

As the complexity of an AUV's performed job for it to accomplish increases, so does the decision-making process of said job too. In ROVs (Remotely Operated Vehicle), this problem is solved by highly trained individuals personally monitoring the situation during work and giving the correct and necessary inputs to the machine to solve or avoid any problems that could occur. But for an AUV to automate itself, algorithms come in to replace this human requirement.

Algorithms are used for an AUV to move from point A to B, avoid obstacles, perform communications, collect data, facilitate target and color detection, give instructions, and solve any problem that might occur with performing a computation.

The underwater environment in which an AUV operates is not a static location like a ground environment. Compared to the ground, underwater surroundings are more dynamic and more prone to change. It is not always possible for AUV path planning operations, such as obstacle avoidance and detection, to be done beforehand as there are many dynamic obstacles to consider such as marine life, reefs, ships, or even various-sized floating objects such as garbage. Articles in the literature were studied to examine the different ways of pathfinding of AUVs for static and dynamic obstacle situations.

All the data collected from the outside by the vehicle were transferred by the Raspberry Pi 4 via a radio frequency (RF) communication module to the main computer. On the system beside the flight control card, a ground control station alongside a companion computer, such as Raspberry Pi 4, was used. Python software language on the Raspberry Pi was selected. All the operations required from the vehicle were coded with Python via the Raspberry Pi. For the ground control station software, the QGroundControl program was used. ArduSub software, developed beside ArduPilot, was used for auto-piloting software. ArduSub works alongside the QGroundControl interface as a ground control station to give an advantage and ease of use for depth fixation, stabilization, and manual driving mode.

The algorithm was prepared for the autonomous software of the vehicle. With this design, an autonomous motion system, target detection and passing, color detection and positioning, target detection, and destruction algorithms were created for their tasks.

2.5.1. Autonomous Motion Algorithm

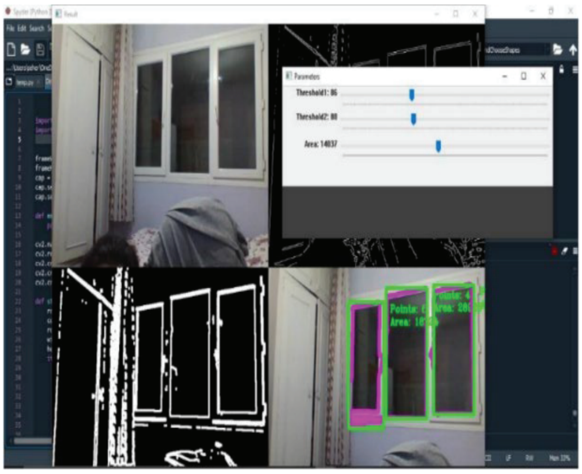
The process of writing an algorithm started with the communications system. An algorithm was written to deal with basic communications between the serial and the sensor according to the data transfer and returns feedback with information about the motor and its position. The next process was to define the starting preparation steps for the movement by checking the data the AUV receives with its sensors. The vehicle starts its movement from a random drop point to the closest corner via its implemented algorithm and software. After the AUV starts the zigzag motion, the vehicle begins to search for the chosen specified target object, which is designated as a submarine, and center itself facing the located submarine with the information received from the camera. Then, through the designated algorithm, the vehicle starts scanning for the designated target which is a circle. The data received through scanning the pool and the circles are kept in mini-PC for storage. The next step after locating and identifying the target object through its colors, is the destruction of said target.

The target object's ping frequency is detected and then the vehicle moves to the target object to directly impact it and destroy it. Another algorithm was written for the AUV to give the vehicle the ability to detect and pass through a door which was shaped as a rectangular hollow object, by centering itself with the information gained via its sensors.

2.5.2. Software Design Process

Here we outline the process of writing automation codes for the AUV to automate itself and fulfill its objectives successfully. Python software language program was chosen as it has an open-sourced library while also being adequate software for image processing. Python 3.9 was the version of the Python program chosen. It is a superb choice as it can be used for prototype development such as an AUV without having problems with maintainability. OpenCV library was used for the image processing library. The image processing tool performs the tasks given to the vehicle such as identification of surroundings, target detection of the door and for passing through it, and color detection for positioning and destruction of the designated target.

The evaluation of the images obtained through the sensors was done by image processing algorithms so that the ESCs and the motors that are connected to them can work in unison. The communication between the Raspberry Pi 4 and Pixhawk was provided with telemetry and general-purpose input/output (GPIO) pins.



Parameters

Threshold1: 86

Threshold2: 88

Area: 14837

Figure 17. Image of the rectangular shape recognition and taskbars

The Pixhawk receives the commands that are transmitted from Raspberry Pi 4 with the help of the "Pymovlink" library in the communication part, thus enabling the Pixhawk to check up on the motors to start the vehicle.

To test the image processing systems, a rectangular-shaped calendar was displayed to the camera from a distance of 30cm. Through its observations, the system recognized four different rectangles. During additional tests at various brightness levels, it was observed that in darker environments with a lesser degree of light, especially in situations of insufficient light levels at close range, object detection of rectangular shapes was not consistent enough. As such, it was concluded that the system can detect an object's rectangular shape accurately given that the environment has sufficient brightness levels or the object itself is under some light source.

To circumvent the issues that occurred in the rectangular shape detection system (Fig. 17) under insufficient brightness levels, sliders that can adjust the settings of the camera were added to perform a calibration to adjust to current light levels for better image processing.

Through the result of the literature review for the task of passing through the door, it was found that the You Only Look Once (YOLO) model was the most efficient method. YOLO is a real-time object detection algorithm that operates by dividing the input image into a grid and predicting bounding boxes and class probabilities for each grid cell. It has small yet simple structural features. With its neural network, it can quickly give a definite recognition output; therefore, it has a fair computation speed. The speed of the algorithm to recognize the object with a fast response time is a positive ability for the AUV. The model of the rectangular-shaped door which the AUV will pass through is shown in Figure 18.

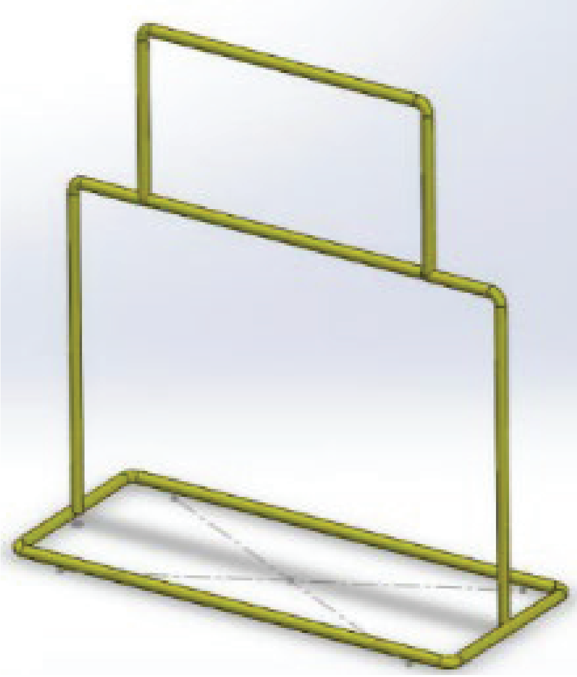
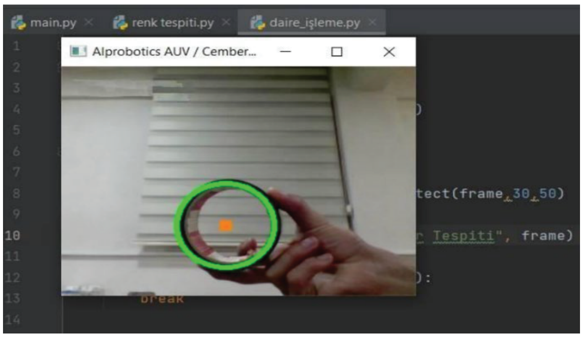


Figure 18. Image of rectangular door model

Gathering a dataset for training the YOLO model is an essential step in the process. We designed a model of the rectangular door using SolidWorks and captured 100 screenshots and photos from various angles. This dataset will help the model learn to recognize and localize the door in different orientations and perspectives. Dividing the dataset into training and testing sets is a common practice in machine learning. Using 80 photos for training and reserving the remaining 20 for testing allows one to evaluate the model's performance on unseen data and assess its generalization abilities.

During the development of the YOLO model in Python, the term "weighting documents" was mentioned. However, it is important to clarify that "weighting documents" is not a standard part of the YOLO training process. In YOLO, the primary objective is to optimize the model's performance by minimizing a designated loss function, which measures the discrepancy between the predicted bounding boxes and the ground truth annotations. This optimization process relies on backpropagation, where the model's weights are iteratively updated to minimize the loss function. The fine-tuning of the model's parameters occurs over multiple epochs until it achieves satisfactory accuracy on the training data. For effective training, it is crucial to strictly adhere to the standard YOLO training procedure and avoid any confusion or misconceptions arising from unconventional terms such as "weighting documents." By following the established training process, the YOLO model can effectively learn and demonstrate improved precision and reliability in detecting and localizing objects in real-world scenarios.



```
tect(frame,30,50)
r Tespiti", frame)
):
break
```

Figure 19. Circle detection

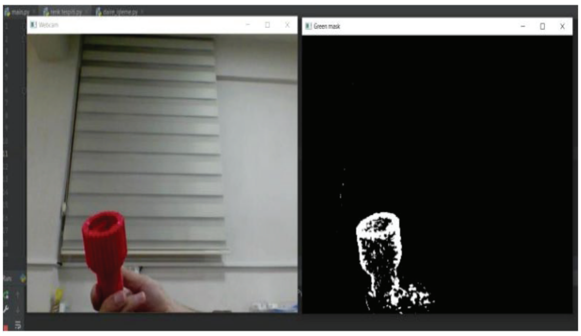


Figure 20. Color red detection test-1

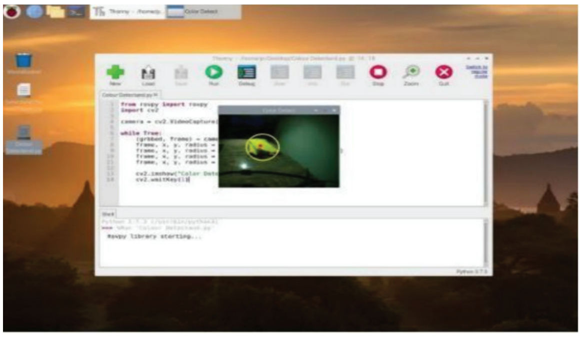
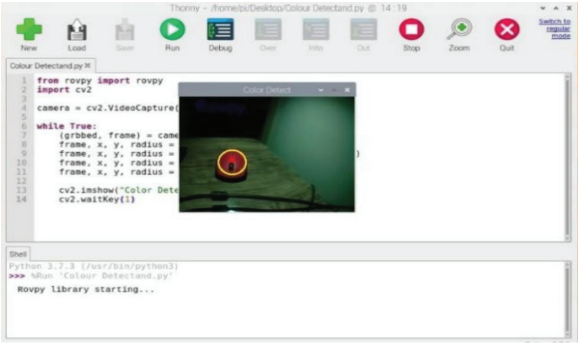


Figure 21. Yellow color detection test



```
from rovpy import rovpy
import cv2

camera = cv2.VideoCapture(

while True:
    (grbbed, frame) = came
    frame, x, y, radius =
    frame, x, y, radius =
    frame, x, y, radius =
    frame, x, y, radius =

    cv2.imshow("Color Dete
    cv2.waitKey(1)
```

```
Python 3.7.3 (/usr/bin/python3)
>>> %Run 'Colour Detectand.py'
 Rovpy library starting...
```

Figure 22. Color red detection test-2

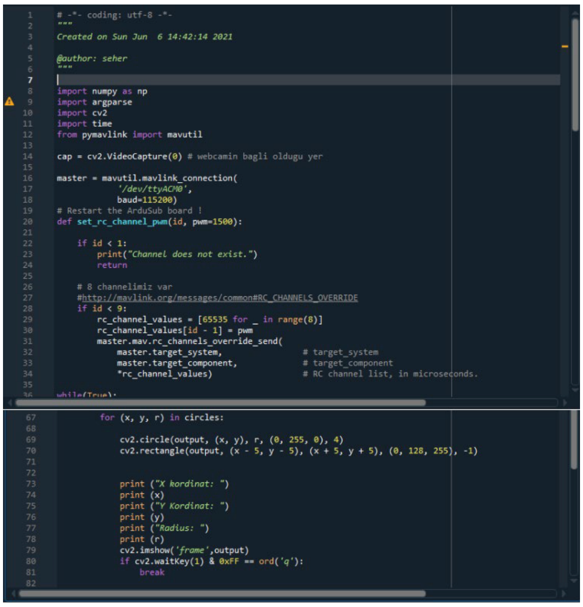
To achieve the target detection and destruction objectives, circle detection was initially tested using Python with a computer camera (Fig. 19). However, this approach proved inadequate due to the camera's poor precision and insufficient lighting conditions in the testing environment.

In subsequent tests, a switch was implemented, transitioning to the Raspberry Pi camera. This change resulted in noticeable enhancements in image processing and color detection capabilities, particularly for objects with red, blue, and green colors (Figs. 20–22). The codes for image processing were specifically tailored to the Raspberry Pi platform, leveraging existing color code libraries. The detection of colors was accomplished using hue saturation values (HSV) as a fundamental component of the image processing algorithm. The utilization of the Raspberry Pi camera, coupled with HSV-based color detection, proved to be a successful combination, significantly improving the accuracy and efficiency of identifying objects based on their distinct color signatures. This advancement in image processing paved the way for more reliable and precise detection and localization of objects in real-world scenarios.

Through the integration of the Raspberry Pi camera and employing color detection techniques based on HSV values, the system's ability to detect circles for target identification and other purposes was significantly improved. The enhanced precision of the Raspberry Pi camera, along with access to specialized color code libraries, allowed for more accurate circle detection, enabling the system to identify circles of specific colors. This capability is valuable for various applications, such as targeted tasks or achieving specific objectives. However, it is important to note that the technology's potential uses extend far beyond destructive purposes and can have diverse applications in a wide range of fields.

In the context of underwater exploration, the identification of the submarine and the accurate positioning of the AUV relative to it are critical tasks. These tasks are accomplished through sophisticated calculations performed with the aid of the powerful computer vision library, OpenCV. The system employs color processing techniques for object recognition during image processing. The goal is to detect and distinguish objects in the underwater environment based on their unique color characteristics. To facilitate this, pre-defined color codes and HSV values from the rovpy library, specifically designed for ROV tools, were utilized. By harnessing the capabilities of OpenCV, the system carries out calculations to detect and identify the submarine (Figs. 23–24).

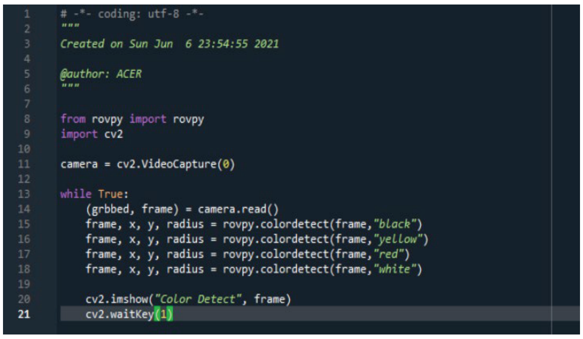
Color processing plays a pivotal role in object recognition, enabling the system to distinguish objects based on their respective colors. The rovpy library's color codes and HSV values are specifically tailored for ROV applications, considering factors such as water clarity, lighting conditions, and typical underwater object appearances. This specialized library streamlines the process of object recognition as it provides a set of ready-to-use color codes that match the prevalent underwater color schemes.



```python
# -*- coding: utf-8 -*-
"""
Created on Sun Jun  6 14:42:14 2021

@author: seher
"""

import numpy as np
import argparse
import cv2
import time
from pymavlink import mavutil

cap = cv2.VideoCapture(0) # webcamin bagli oldugu yer

master = mavutil.mavlink_connection(
            '/dev/ttyACM0',
            baud=115200)
# Restart the ArduSub board !
def set_rc_channel_pwm(id, pwm=1500):

    if id < 1:
        print("Channel does not exist.")
        return

    # 8 channelimiz var
    #http://mavlink.org/messages/common#RC_CHANNELS_OVERRIDE
    if id < 9:
        rc_channel_values = [65535 for _ in range(8)]
        rc_channel_values[id - 1] = pwm
        master.mav.rc_channels_override_send(
            master.target_system,                # target_system
            master.target_component,             # target_component
            *rc_channel_values)                  # RC channel list, in microseconds.

while(True):
```

```python
            for (x, y, r) in circles:

                cv2.circle(output, (x, y), r, (0, 255, 0), 4)
                cv2.rectangle(output, (x - 5, y - 5), (x + 5, y + 5), (0, 128, 255), -1)


                print ("X kordinat: ")
                print (x)
                print ("Y Kordinat: ")
                print (y)
                print ("Radius: ")
                print (r)
                cv2.imshow('frame',output)
                if cv2.waitKey(1) & 0xFF == ord('q'):
                    break
```

Figure 23. Codes used in underwater software applications



```python
# -*- coding: utf-8 -*-
"""
Created on Sun Jun  6 23:54:55 2021

@author: ACER
"""

from rovpy import rovpy
import cv2

camera = cv2.VideoCapture(0)

while True:
    (grbbed, frame) = camera.read()
    frame, x, y, radius = rovpy.colordetect(frame,"black")
    frame, x, y, radius = rovpy.colordetect(frame,"yellow")
    frame, x, y, radius = rovpy.colordetect(frame,"red")
    frame, x, y, radius = rovpy.colordetect(frame,"white")

    cv2.imshow("Color Detect", frame)
    cv2.waitKey(1)
```

Figure 24. Codes used in color detection software applications

Integrating OpenCV with the rovpy library's color codes and HSV values enabled the system to successfully accomplish its mission of identifying the submarine and accurately determining the AUV's position relative to it. This approach enhanced the system's image processing capabilities, facilitated precise object recognition, and eliminated the need for extensive manual calibration or custom color coding, streamlining the overall process.

The image processing systems play a crucial role in providing valuable data for the AUV's operations. They enable the AUV to detect colored rings placed around the submarine. By analyzing the captured images, the system can identify and locate these colored rings accurately. The system's sophisticated algorithms allow it to calculate the diameters of the rings, providing essential measurements for precise navigation. With the data acquired from the image processing, the AUV uses this information to position itself in the closest proximity to the submarine.

There were theoretical plans to incorporate an underwater microphone into the vehicle for the purpose of detecting a target pinging ball. The proposed system would have included additional components such as a transmitting and receiving transistor, as well as a signal generator capable of transmitting a 45 kHz acoustic wave. The intention was to synchronize the frequency range of the buzzer implanted on the vehicle with the 45 kHz frequency. This would enable the vehicle's motors to be directed based on the frequency direction using the assistance of the buzzer. However, these ideas and applications were not implemented in the current iteration and were left for future revisions.

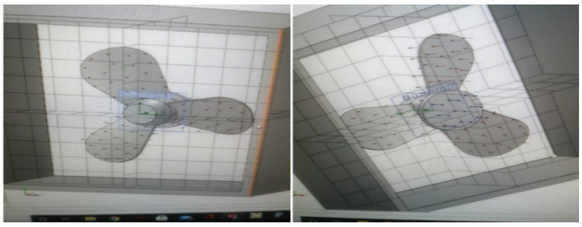


Figure 25. Technical drawing of a flow simulation of a propeller

2.5.3. External Interface

The system's external interface integrates control parameters for motor settings, underwater lighting systems, and sensor calibration settings. Effective data flow within the system is enabled by the QGroundControl software, facilitating communication and management. Motor controls and sensor data processing were implemented using the Pixhawk Toolchain, a robust software framework designed specifically for autonomous vehicles.

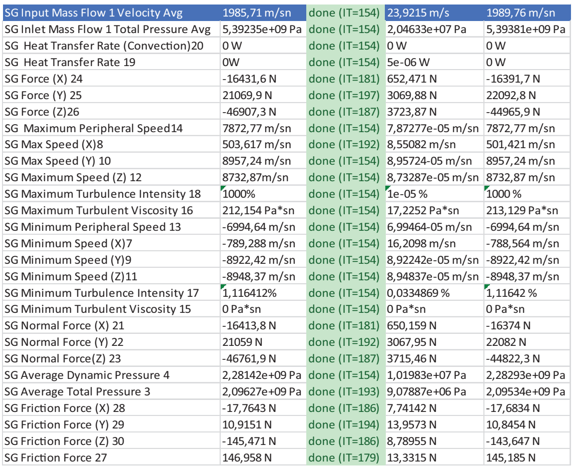
Image processing tasks were executed using the Python programming language on the Raspberry Pi 4 throughout all stages. Python's extensive libraries and user-friendly syntax provide a flexible and efficient platform for image processing within the system. By harnessing Python's capabilities on the Raspberry Pi 4, the system efficiently processes and analyzes image data, supporting various functionalities and objectives.

3. Results and Discussion

3.1. Flow Simulation of Propeller AUV

The propeller, initially designed using the SolidWorks program, underwent a comprehensive analysis to assess its thrust force and the adjustability of the propeller blade angle using a computer-aided flow simulation program. The simulation involved the adjustment of various parameters to investigate the propeller's performance.

Figure 25 presents images captured during the analysis, depicting the examination of ambient pressure and the analysis of inlet-outlet flow in the specific section with the designated flow direction. These visualizations offer valuable insights into the behavior of the propeller and its interaction with the surrounding fluid throughout the simulation process.



SG Input Mass Flow 1 Velocity Avg	1985,71 m/sn	done (IT=154)	23,9215 m/s	1989,76 m/sn
SG Inlet Mass Flow 1 Total Pressure Avg	5,39235e+09 Pa	done (IT=154)	2,04633e+07 Pa	5,39381e+09 Pa
SG Heat Transfer Rate (Convection)20	0 W	done (IT=154)	0 W	0 W
SG Heat Transfer Rate 19	0W	done (IT=154)	5e-06 W	0 W
SG Force (X) 24	-16431,6 N	done (IT=181)	652,471 N	-16391,7 N
SG Force (Y) 25	21069,9 N	done (IT=197)	3069,88 N	22092,8 N
SG Force (Z)26	-46907,3 N	done (IT=187)	3723,87 N	-44965,9 N
SG Maximum Peripheral Speed14	7872,77 m/sn	done (IT=154)	7,87277e-05 m/sn	7872,77 m/sn
SG Max Speed (X)8	503,617 m/sn	done (IT=192)	8,55082 m/sn	501,421 m/sn
SG Max Speed (Y) 10	8957,24 m/sn	done (IT=154)	8,95724-05 m/sn	8957,24 m/sn
SG Maximum Speed (Z) 12	8732,87m/sn	done (IT=154)	8,73287e-05 m/sn	8732,87 m/sn
SG Maximum Turbulence Intensity 18	1000%	done (IT=154)	1e-05 %	1000 %
SG Maximum Turbulent Viscosity 16	212,154 Pa*sn	done (IT=154)	17,2252 Pa*sn	213,129 Pa*sn
SG Minimum Peripheral Speed 13	-6994,64 m/sn	done (IT=154)	6,99464-05 m/sn	-6994,64 m/sn
SG Minimum Speed (X)7	-789,288 m/sn	done (IT=154)	16,2098 m/sn	-788,564 m/sn
SG Minimum Speed (Y)9	-8922,42 m/sn	done (IT=154)	8,92242e-05 m/sn	-8922,42 m/sn
SG Minimum Speed (Z)11	-8948,37 m/sn	done (IT=154)	8,94837e-05 m/sn	-8948,37 m/sn
SG Minimum Turbulence Intensity 17	1,116412%	done (IT=154)	0,0334869 %	1,11642 %
SG Minimum Turbulent Viscosity 15	0 Pa*sn	done (IT=154)	0 Pa*sn	0 Pa*sn
SG Normal Force (X) 21	-16413,8 N	done (IT=181)	650,159 N	-16374 N
SG Normal Force (Y) 22	21059 N	done (IT=192)	3067,95 N	22082 N
SG Normal Force(Z) 23	-46761,9 N	done (IT=187)	3715,46 N	-44822,3 N
SG Average Dynamic Pressure 4	2,28142e+09 Pa	done (IT=154)	1,01983e+07 Pa	2,28293e+09 Pa
SG Average Total Pressure 3	2,09627e+09 Pa	done (IT=193)	9,07887e+06 Pa	2,09534e+09 Pa
SG Friction Force (X) 28	-17,7643 N	done (IT=186)	7,74142 N	-17,6834 N
SG Friction Force (Y) 29	10,9151 N	done (IT=194)	13,9573 N	10,8454 N
SG Friction Force (Z) 30	-145,471 N	done (IT=186)	8,78955 N	-143,647 N
SG Friction Force 27	146,958 N	done (IT=179)	13,3315 N	145,185 N

Figure 26. Output image of flow simulation results-1

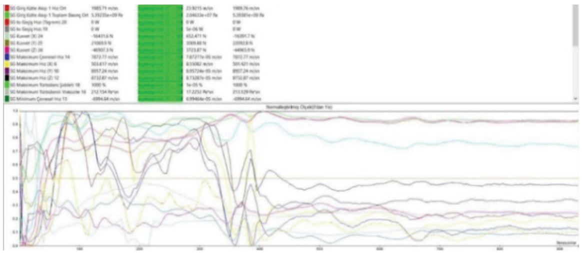


Figure 27. Flow simulation iteration-based graphic results-1

The gathered data analysis was successfully executed. However, upon further examination, it was observed that the output results of the analysis were derived from constant values used at the outset of the process. As a consequence, the obtained measured data were found to be 10 times higher than the expected nominal values. This discrepancy led to an inaccurate representation in the schematic visualization of the graph.

The resulting graph schematic did not align with the initially anticipated pattern, which was expected to display a linear increase transitioning into a constant flow after reaching the maximum level. The issue arose due to the utilization of incorrect values in the analysis. To address this, the analysis needs to be reevaluated, and the correct values should be utilized in the calculations. Rectifying the input values will enable subsequent analysis and graph generation to provide a more accurate representation of the data. This will facilitate a proper interpretation of the flow behavior and its relationship to the maximum level, allowing for a more reliable assessment of the system's performance. Conducting the analysis with precise and accurate values will ensure that the graph exhibits the expected linear increase and transition into a constant flow at the maximum level. This correction will enhance the reliability and validity of the analysis, enabling meaningful conclusions to be drawn from the graph (Figs. 26–27).

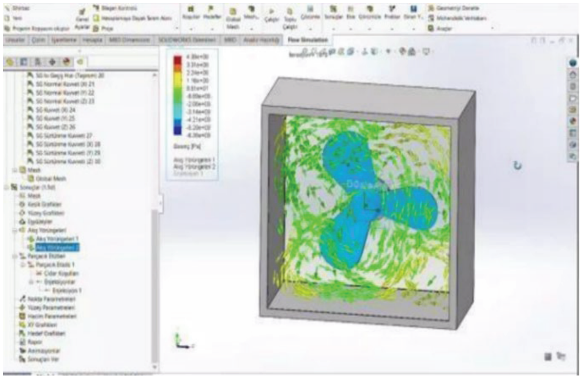


Figure 28. Flow simulation of propeller rotation

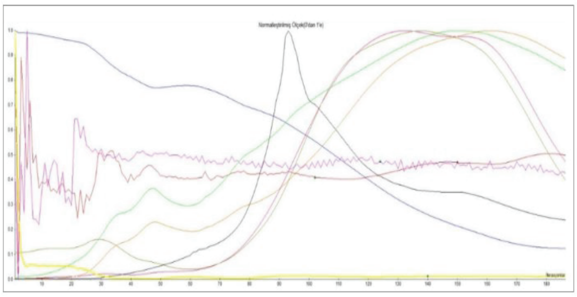


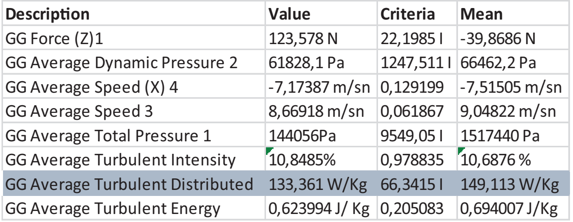
Figure 29. Flow simulation iteration-based graphic results-2

Further analysis was conducted on the propeller using a flow simulation program to determine the rotation direction of the propeller (Fig. 28) and investigate any potential turbulence generated during its rotation. This analysis aimed to gain insights into the propeller's behavior and understand the flow dynamics associated with its rotation.

By utilizing the flow simulation program, detailed examinations were carried out to visualize and study the propeller's rotation characteristics. This included assessing the direction of rotation and studying the flow patterns and turbulences that might arise during propeller operation.

The analysis provided valuable information regarding the propeller's performance and flow behavior. These insights can be used to optimize the propeller design, improve efficiency, and minimize any undesirable effects such as turbulence or cavitation.

During the analysis, the output values demonstrated a notable decrease after reaching a peak value, followed by linear decreases in subsequent iterations. This trend, along with the prolonged run durations, confirmed the accuracy of the analysis results. Since the motors used in the system provide unidirectional rotation, the propeller direction was assumed to be clockwise for the analysis. This assumption was based on the fact that the propellers in the AUV work in both clockwise and counterclockwise directions. The analysis results are depicted in Figs. 29–30.



Description	Value	Criteria	Mean
GG Force (Z)1	123,578 N	22,1985 I	-39,8686 N
GG Average Dynamic Pressure 2	61828,1 Pa	1247,511 I	66462,2 Pa
GG Average Speed (X) 4	-7,17387 m/sn	0,129199	-7,51505 m/sn
GG Average Speed 3	8,66918 m/sn	0,061867	9,04822 m/sn
GG Average Total Pressure 1	144056Pa	9549,05 I	1517440 Pa
GG Average Turbulent Intensity	10,8485%	0,978835	10,6876 %
GG Average Turbulent Distributed	133,361 W/Kg	66,3415 I	149,113 W/Kg
GG Average Turbulent Energy	0,623994 J/ Kg	0,205083	0,694007 J/Kg

Figure 30. Output image of flow simulation results-2

3.2. Results of the AUV

Following the completion of the electronic, mechanical, and algorithm preliminary designs, the implementation and testing phases were initiated. The first step involved 3D printing of the propellers responsible for motor movement and water flow in the pool. Subsequently, the assembly of the motors, propellers, and thruster was carried out. However, it was discovered that the nut used to connect the propeller to the motor shaft caused damage to the propeller skirts.

In the next stage, the assembly between the power unit, ESC, and the motor was established and subjected to testing. During this testing phase, it became evident that the ESCs and connection cables with a resistance of 30A were prone to overheating, which could potentially lead to malfunctions in the vehicle.

Furthermore, underwater testing was conducted to assess the performance of the motors and propellers. Unfortunately, the desired efficiency was not achieved due to a weak connection between the motor and the propeller. It was also observed that the blade angle of the propeller blades contributed to the inability to attain the desired efficiency.

Based on these findings, modifications were made to the propeller design to address the identified issues and improve performance.

To address the issues encountered during the testing phases, several modifications and adjustments were made to the design and components of the vehicle.

First, a nut slot was added to the propeller design to ensure a proper connection between the motor and the propeller. This modification resolved the previous issues related to the connection.

Next, the blade angle of the propeller was reduced to enhance its efficiency while operating underwater. This adjustment improved the propeller's performance and contributed to more efficient propulsion.

To mitigate the heat generated by the ESC and cables, an ESC with 40A resistance and thicker cables were integrated into the vehicle instead of the 30A ESC. This upgrade helped prevent overheating and potential malfunctions.

In the electronic design of the vehicle, two copper cables were initially employed for power distribution. However, it was discovered that these cables, like the ESC connection cables, were overheating beyond acceptable levels. Additionally, the Raspberry Pi minicomputer experienced the same overheating issue. Measures were taken to address these problems and ensure proper heat dissipation within the electronic components.

During the sensor testing phase, it was found that the sonar sensor's acoustic waves could not pass through the front cover frame in the tube, resulting in inadequate data. Subsequently, a connection was established between the Raspberry Pi and Pixhawk via the Pixhawk's USB port. Unfortunately, this connection method led to delayed data processing and inefficiencies in handling the requested data.

Finally, it was determined that the buzzer sensors intended for detecting the pinger ball underwater were not operating efficiently within the tube. Further improvements or alternative approaches were explored to address this limitation.

By implementing these modifications and adjustments, efforts were made to enhance the overall functionality and efficiency of the vehicle and its components.

In response to the encountered situations, several changes and adjustments were made to improve the system's performance and efficiency.

To address the overheating and power distribution issues, copper cables were replaced with more robust copper plates with a strength rating of 400A. This upgrade ensured reliable power distribution within the system. Additionally, a metal cooler was integrated to prevent overheating of the Raspberry Pi minicomputer and enhance the efficiency of the processor.

The connection between the Raspberry Pi and Pixhawk was established via the telemetry port, allowing for smoother and more reliable communication between the two devices.

To enhance the data collection capabilities of the sonar and buzzer sensors, water-resistant models of these sensors were ordered. This upgrade aimed to provide more efficient and reliable data acquisition in underwater conditions.

In the algorithm design process, an initial approach involved a single algorithm to handle all tasks. However, it was recognized that this approach might lead to challenges in completing the tasks effectively. As a result, a revised strategy was adopted, wherein algorithms were developed individually for each specific task. This iterative approach ensures that each task is addressed effectively and improves the overall performance of the system.

By implementing these changes and refining the algorithms, efforts were made to overcome the encountered issues and enhance the system's efficiency and performance.

4. Conclusion

With the growing demands and advancements in technology, there is a noticeable increase in interest surrounding unmanned and autonomous vehicles. Recognizing this trend, the deployment of underwater vehicles with such capabilities holds significant promise for our coastal nation. The potential contributions of these vehicles span various domains, encompassing scientific research and environmental preservation.

Notably, autonomous underwater vehicle studies are infrequent in our country. Against this backdrop, a research project has been undertaken to provide a significant addition to the current literature. The objective is to explore and advance autonomous underwater vehicles, thereby enriching the knowledge and comprehension within this domain. This initiative, by addressing the research gap, aims to uncover the potential benefits and applications of autonomous underwater vehicles tailored to our specific context.

This study conducted a comprehensive investigation into the mechanical, electrical-electronic, and software prerequisites essential for the creation of an unmanned and autonomous underwater vehicle. Emphasis was placed on identifying crucial materials indispensable for the vehicle's production, accompanied by a detailed exploration of the production methods and equipment integral to the manufacturing process.

The mechanical requirements encompass the structural design, propulsion system, buoyancy control, and other mechanical components crucial for underwater vehicle operation. The electrical-electronic aspects cover the necessary sensors, communication systems, power distribution, and control mechanisms required for autonomous functionality.

Additionally, the article explored the software requirements involved in programming the vehicle's autonomous behavior, including navigation algorithms, sensor fusion techniques, and mission planning.

Furthermore, attention was given to the materials and manufacturing techniques employed in the construction of the autonomous underwater vehicle. The article provided insights into the selection of suitable materials, fabrication methods, assembly procedures, and quality control measures to ensure the vehicle's robustness and reliability in underwater environments.

By addressing these key aspects, the article aimed to offer a comprehensive overview of the requirements, materials, production methods, and equipment involved in the development of an unmanned and autonomous underwater vehicle.

After conducting a thorough literature review, the minimum requirements for autonomous underwater vehicles were established. Based on these findings, a mechanical design was created using SolidWorks, a 3D drawing program, and the production was realized using a 3D printer. Furthermore, calculations were performed to ensure suitability for underwater conditions, and appropriate materials such as motors, ESCs, and power units were carefully selected and procured.

To enable autonomous capabilities, a combination of sensors, a Raspberry Pi minicomputer, and a Pixhawk flight controller board were incorporated. Essential driving and image processing software was developed to facilitate autonomous operations. Enhancements in the utilization of electronic sensors can greatly contribute to future studies, offering improved efficiency and guidance. Moreover, a more efficient integration of sensor selection and software for underwater sound detection can enhance the vehicle's adaptability to underwater conditions.

Recommendations for enhancing the performance and reliability of the AUV include exploring integrated solutions for waterproofing the vehicle's motors and conducting flow and strength analyses using diverse materials. These measures are deemed essential for extending the longevity and ensuring the reliability of the vehicle, particularly in challenging underwater environments. The findings underscore the significance of ongoing research and development efforts aimed at optimizing the performance and capabilities of autonomous underwater vehicles. Incorporating the suggested improvements in sensor integration, software development, and material analysis is crucial for further enhancing the vehicle's functionality and adaptability in underwater scenarios.

AUTHORS

Ismail Bogrekci – Department of Mechanical Engineering, Faculty of Engineering, Aydin Adnan Menderes University, Efeler, Aydin, Türkiye/ Alprobotics Inc., Co-founder, Aydin, Türkiye, e-mail: ibogrekci@adu.edu.tr.

Pinar Demircioglu* – Institute of Materials Science, TUM School of Engineering and Design, Technical University of Munich (TUM), Garching, Munich, 85748, Germany/ Dept. of Mechanical Engineering, Faculty of Engineering, Aydin Adnan Menderes University, Aydin, Turkey/Alprobotics Inc., Co-founder, Aydin, Türkiye, e-mail: pinar.demircioglu@tum.de, pinar.demircioglu@adu.edu.tr.

Goktug Ozer – Alpler Agricultural Machinery, R&D Manager, Aydin, Türkiye, e-mail: goktug.ozer@alpler.com.tr.

*Corresponding author

ACKNOWLEDGEMENTS

We would like to thank ADU BAP Research Infrastructure Project (project number MF-22005) for supporting this study. Alprobotics and Alpler Agricultural Machinery have contributed to the experiments.

References

[1] W. Linling, Z. Daqi, P. Wen, and Z. Youmin, "A Survey of Underwater Search for Multi-Target using Multi-AUV: Task Allocation, Path Planning, and Formation Control," *Ocean Engineering*, vol. 278, 2023, 114393; doi: 10.1016/j.oceaneng.2023.114393.

[2] E.S. Ali et al., "A Systematic Review on Energy Efficiency in the Internet of Underwater Things (IoUT): Recent Approaches and Research Gaps," *Journal of Network and Computer Applications*, vol. 213, 2023, pp. 1–22, 103594; doi: 10.1016/j.jnca.2023.103594.

[3] B. Zhang et al., "Autonomous Underwater Vehicle Navigation: A Review," *Ocean Engineering*, vol. 273, 2023, 113861, pp. 1–29; doi: 10.1016/j.oceaneng.2023.113861.

[4] A. Bahr, J.J. Leonard, and A. Martinoli, "Dynamic Positioning of Beacon Vehicles for Cooperative Underwater Navigation," In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3760–3767; doi: 10.1109/IROS.2012.6386168.

[5] C. German et al., "Hydrothermal Exploration with the Autonomous Benthic Explorer," *Deep Sea Research Part I: Oceanographic Research Papers*, vol. 55, 2008, pp. 203–219; doi: 10.1016/j.dsr.2007.11.004.

[6] B. Allotta et al., "The ARROWS Project: Robotic Technologies for Underwater Archaeology," *IFAC-PapersOnLine*; vol. 48, no. 2, 2015, pp. 194–199, ISSN 2405-8963; doi: 10.1016/j.ifacol.2015.06.032.

[7] F. Schill, A. Bahr, and A. Martinoli, "Vertex: A New Distributed Underwater Robotic Platform for Environmental Monitoring," In: *Distributed Autonomous Robotic Systems*. Springer Proceedings in Advanced Robotics, vol. 6, Cham: Springer, 2018; doi: 10.1007/978-3-319-73008-0_47.

[8] R. Katzschmann et al., "Exploration of Underwater Life with an Acoustically Controlled Soft Robotic Fish," *Science Robotics*, vol. 3 no. 16, 2018, eaar3449; doi: 10.1126/scirobotics.aar3449.

[9] T. Wu et al., " A Hydrothermal Investigation System for the Qianlong-II Autonomous Underwater Vehicle," *Acta Oceanologica Sinica*; vol. 38, 2019, pp. 159–165; doi: 10.1007/s13131-019-1408-4.

[10] G. Li et al., "Self-Powered Soft Robot in the Mariana Trench," *Nature*, vol. 591, 2021, pp. 66–71; doi: 10.1038/s41586-020-03153-z.

[11] A. Palomer, P. Ridao, and D. Ribas, "Multibeam 3D Underwater SLAM with Probabilistic Registration," *Sensors*, vol. 16, no. 4, 2016:560; doi: 10.3390/s16040560.

[12] R. Kimura et al., "Guidance and Control of an Autonomous Underwater Robot for Tracking and Monitoring Spilled Plumes of Oil and Gas from Seabed," In: *The Twenty-Third International Offshore and Polar Engineering Conference*. OnePetro, 2013, pp. 366–371.

[13] G. Marani, S.K. Choi, and J. Yuh, "Underwater Autonomous Manipulation for Intervention Missions AUVs," *Ocean Engineering*, vol. 36, no. 1, 2009, pp. 15–23; doi: 10.1016/j.oceaneng.2008.08.007.

[14] M.J. Hamilton, S. Kemna, and D. Hughes, "Antisubmarine Warfare Applications for Autonomous Underwater Vehicles: The GLINT09 Sea Trial Results," *J. Field Robotics*; vol. 27, 2010, pp. 890–902; doi: 10.1002/rob.20362.

[15] D.P. Williams, "On Optimal AUV Track-Spacing for Underwater Mine Detection," In: *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 4755–4762; doi: 10.1109/ROBOT.2010.5509435.

[16] M. Prats et al., "Reconfigurable AUV for Intervention Missions: A Case Study on Underwater Object Recovery," *Intelligent Service Robotics*, vol. 5, 2012, pp. 19–31; doi: 10.1007/s11370-011-0101-z.

[17] G. Casalino et al., "Underwater Intervention Robotics: An Outline of the Italian National Project MARIS," *Marine Technology Society Journal*, vol. 50, 2016, pp. 98–107; doi: 10.4031/MTSJ.50.4.7.

[18] G. Ozer, "Development of Autonomous Underwater Vehicle," Unpublished M.Sc. Thesis, 2023-M.Sc.-032. Aydın Adnan Menderes University, Türkiye, 2023.