Original article

Fully autonomous underwater spy-vehicle

Kozluca AK[1], Demirdelen C[2], Bozkurt KB[3*]

Submitted: April, 27, 2023, Revised: version 1, May 3, 2023
Accepted: May 4, 2023

Abstract
In recent years, nations have adopted technology as a means to excel. Technological competition among developed countries has begun to filter to the forefront in other fields. Efforts to stay up to date in a geopolitically changing world order include technological advances in military technology. Although rapid advancements have occurred in land and air based military technologies, underwater technologies have received less attention. The goal of this project is to develop a fully autonomous underwater vehicle which can reach designated coordinates, gather and process images along the way, evade attacks by launching out of the water and gliding using wings and not use GPS or external signals (thus making it invisible). The model is envisaged to detect and intervene in potentially dangerous attacks that may occur in territorial waters by relying on its invisibility, fully autonomous movement underwater and highly capable escape mechanisms. Earlier models with fish geometry have been fabricated in various countries. However, these models are not fully autonomous and lack the spying capabilities of the proposed model. A prototype of the model was created and analyzed using the engineering simulation software, ANSYS; then 3D printed. The electronics of the prototype were powered by an ESP32, system on a chip micro-controller circuit. The proposed vehicle has the capability to reach a predesignated destination, gather images, and return, without using GPS. Results of the underwater and surface tests carried out at the Şehit Göksel Koç Dam in Isparta, Kozluca, indicated that the model successfully performed fully autonomous movement and escape maneuvers.

Keywords
Underwater vehicle, Fully autonomous, GPS-free, Spy vehicle, Autonomous underwater vehicle, Swallow fish, Inertial Measurement Unit, Submarine, Torpedo, Escape maneuver

[1]Ali Kağan Kozluca, Ankara Science High School, Çiğdem, ODTÜ Ormanı, 1557. Sk. No:32, 06530 Çankaya/Ankara, Türkiye, alikagan.kozluca@hotmail.com
[2]Can Demirdelen, Ankara Science High School, candemirdelen3@gmail.com
[3*]Corresponding author: Kerem Berk Bozkurt, Ankara Science High School, kberkboz@gmail.com

1. Introduction

1.1 Autonomous vehicles

Autonomous vehicles are programmed and produced in such a way that they can reach a predesignated target, interpret the events around them, take precautions and react to various events, perform their task successfully and do all this on their own without the need for any human intervention (1).

1.2 Advantages of autonomous vehicles

Since autonomous vehicles are unmanned, there is no risk of human casualties in a situation that may pose a threat. Since they are computer controlled, the margin of error is lower than that of manned vehicles, making them more reliable. Once programmed, they can work continuously and effortlessly, as they do not require any additional commands or external input (1).

1.3 The importance of underwater espionage in the military field

In order to succeed in the military field, it is as important to develop strategies that are compatible with the aims, interests, and competencies of the country, as well as to access the information necessary for the creation of these strategies and to make sense of this information through a comprehensive evaluation process. Accordingly, states must have a strong network of intelligence in order to determine strategies for war. This network of intelligence is powered by strong tools and technology. Adversaries obtain information through various methods, one of which is underwater espionage (2).

1.4 Previous underwater autonomous vehicle studies

Adversaries compete on land, in the air and underwater. Various underwater autonomous vehicle studies have been conducted over the years. The "Hydrus" produced by Advanced Navigation, Sydney, Australia, can autonomously collect data underwater, but does not seem to have been specifically designed for security applications. It has no escape mechanism when faced with threats and cannot be used for spying (3). Boston Engineering, Waltham, MA, USA, has conducted autonomous underwater studies, but since their work is for safety purposes and not inherently designed for espionage, the vehicle can be easily detected and their maneuverability is poor. These limitations exist partly because of their relatively larger size and the usage of GPS (4).

1.5 Advantages of GPS silent technology

GPS systems require communicating with satellites, thus emitting signals and making the vehicle detectable by adversaries. Vehicles without GPS are considerably advantageous for espionage because this mode of signalling is not used (5), thereby evading detection.

1.6 Torpedoes and torpedo types

The task of the torpedoes with propellers and rudders is to lock on to underwater targets and destroy them. Straight runner torpedoes do not have a target search system. These torpedoes are effective at short range as they lack the ability to re-attack. They can be countered with maneuvers or harsh killing. Wire-guided torpedoes manuvour around the vehicle with wire guidance and perform secret searches, both passive and active. The specially designed guided wire system enables two-way data exchange between the torpedo and the attacker, maximizing combat power. Acoustically guided torpedoes target by listening to the characteristic sound of its target by using sonar. They are suitable for mid-range use (6, 7).

1.7 Precautions against torpedoes and reducing the likelihood of capture or destruction

There are many trap mechanisms against torpedoes, but modern torpedoes are able to detect these traps. With anti-torpedo systems, it is possible to neutralize the torpedo, but this is not the case for small sized systems. One way to neutralize torpedoes is to keep the distance until the torpedo runs out of battery power. This distance protection is achieved by vertical or horizontal maneuvers. For small-sized systems, the most effective method is to maintain the distance until the glove box runs out of battery (8-10).

2. Methods

2.1 Mechanical Methods

Anatomical characteristics of the Swallow Fish were examined in order to establish a design outline in parallel with those of the fish. Subsequently, the drag coefficient, lift coefficient, structural strength, dynamic flow analyses of the wing, camera and engine sections were designed. Various iterations were made keeping the results of the analyzes in mind until the ideal geometry was achieved. The slots and connection points of the electronic materials, which provided movement, were added in accordance with the dimensions of the design and materials. The design was re-analyzed. The overall design was finalized once the optimum operating conditions were met. All designs were made using FUSION 360, Autodesk, San Fransisco, CA, USA, and the analyses were made using the ANSYS engineering simulation software, Ansys, Canonsburg, PA, USA.

Fusion 360, is a cloud-based 3D computer-aided design (CAD), computer-aided manufacturing (CAM), and computer-aided engineering (CAE) software tool. It is used for designing, simulating, and manufacturing products in various industries such as mechanical engineering, industrial design, and architecture. Fusion 360 offers a wide range of features and capabilities such as parametric modeling, direct modeling, assembly modeling, and simulation, making it a versatile tool for product design and development.

ANSYS is a simulation software tool that is primarily used for engineering simulation, finite element analysis (FEA), and computational fluid dynamics (CFD) analysis. ANSYS is commonly used for solving complex engineering problems related to stress, thermal, and fluid flow analysis. It offers a wide range of simulation capabilities and can simulate real-world conditions to accurately predict the behavior of a product or system.

After the designing phase, the final product was 3D printed. It was produced as 9 pieces in total. PolyLactic acid (PLA) plastic was used as the production material. In parallel with the production, the connections of electronic materials were made, and their algorithms were written. The test model was created by combining the body and the electronic parts. The model was tested in the on-campus pool, in order to ascertain that it executed per design (11).

The primary reason for using PLA plastic in the prototype production was that it could be printed easily. The impact resistance of PLA is lower than Acrylonitrile Butadiene Styrene (ABS) plastic, but it was sufficiently high in our design. The price of PLA is lower than ABS, which reduces the cost of the prototype. PLA cannot withstand high temperatures, but this is not a concern as the temperature of the water is variable and low in the marine environment (12-13).

The Swallow Fish was chosen because it moves underwater using its wings. The swallow fish was a suitable model to base to project off of since its features and usage of wings is parallel to the proposed project, which also is a vehicle that utilizes wings underwater. The swallow fish also has an ability to maneuver very quickly when it expands its wings fully, and its wings are similar to the ones of birds, which can be used to evade threats by gliding in the air. First, the body and wing structure were examined. It was observed that the skeletal structure was curved from the anterior part to the posterior part. The curve in the front of the middle axis of the skeleton was more inclined than the rear. It was observed that the position of the wings was in the front of what would be the mid-axis of the fuselage. According to the results of the observations, the main design outline was created.

Figure 1: Examination of the Swallow Fish as a template for prototype design

As a result of the examination, the main body of the first model was created in FUSION 360. The body was designed with electronic materials in mind. This design was analyzed in the ANSYS software, and the positive and negative aspects of the design were determined. Below are the analyzes and designs made in Figure 2.In the next iteration, the body length was reduced, and the width was increased to better establish the center of balance. Thus, the weight was reduced, and the internal volume was increased.

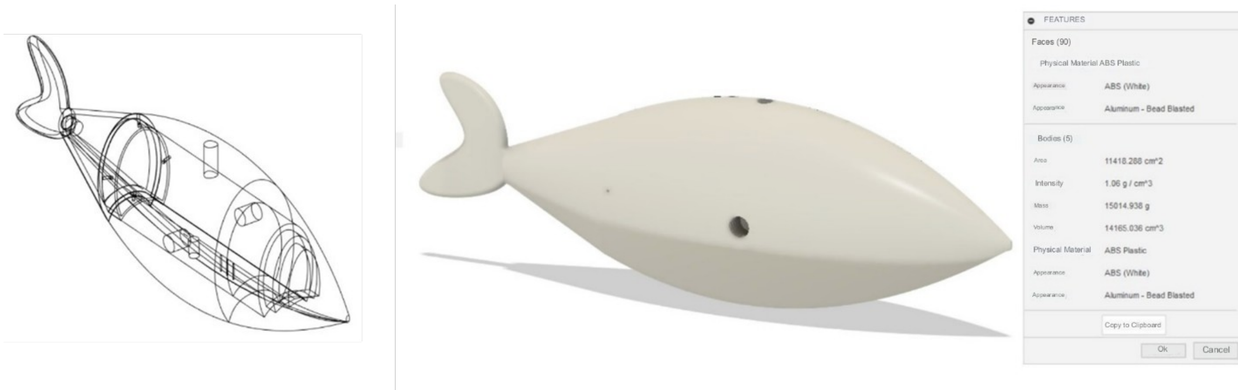


Figure 2: The Geometry of the First Fuselage and Its Structural Properties

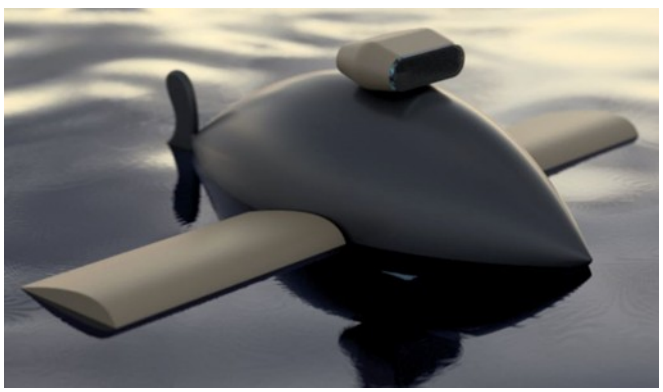


Figure 3: The Final Model in Water

After the fuselage was completed, the design process for the wings started. Wing dimensions (Length:242mm x Width:112.5mm x Width:30mm) were scaled up in proportion to the wing lengths of the fish. In order to minimize the weight (210g), the internal structure of the wing was designed to be hollow, with inner support columns bearing the load. In the first design, the wing was drawn in the *Low-Speed ULM* profile. When the wings were subjected to ANSYS analysis, it was observed that the lift and drag forces were significantly greater than the desired values. As a result, a symmetrical wing structure was designed, and the angle of attack was imparted to the wings by using motors. Lift and drag forces were kept at desired values using these motors.

Figure 4: Interior and Exterior Structure of the Wing

After the wing and fuselage designs were completed, the parts were assembled and fluid analysis, structural strength, lift, drag forces and thrust moment analyses were performed using ANSYS. The distributions of these values are presented as graphs and simulations. Since the wings were in a symmetrical profile, the lift force was very close to zero due to the *Bernoulli's effect* at 2 degrees. In order to adjust the depth of the vehicle underwater while in motion, the adjustable angles of the wings were utilized and upward or downward movement was achieved. The drag and lift forces of the wings were calculated with given angles.

The angles of the wings can be adjusted to achieve rotational motion of the vehicle during movement. The change results in a difference between the lift and drag forces between the wings. This difference, in turn causes a difference in the linear velocity measured across the two wings. As a result, the vehicle moves toward the direction where the linear velocity is smaller.

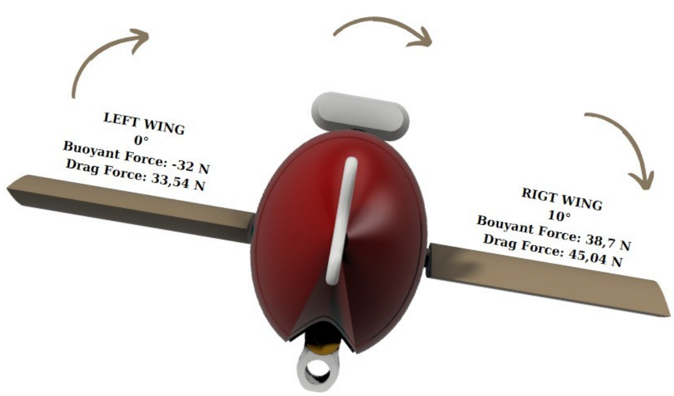


Figure 5: Rotational Motion Resulting from Force Difference, Calculated for 0° at Left Wing and 10° at Right Wing

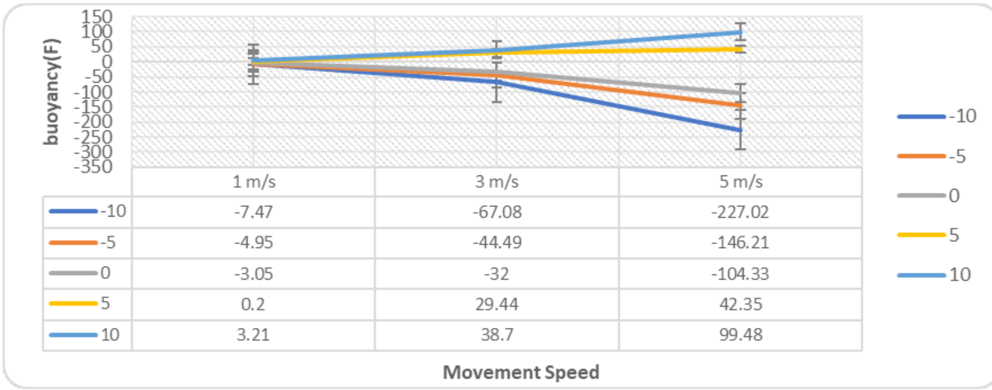


Figure 6: Buoyancy as a function of the movement speed of the vehicle. Each line represents the buoyancy for a complementary angles. The values for the buoyancy are presented in the table.

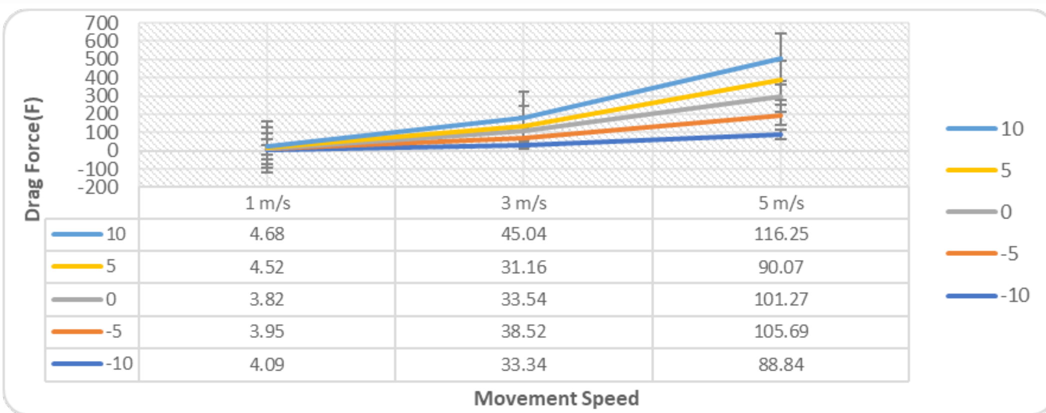


Figure 7: Drag force generated by the movement speed of the vehicle. Each line represents the drag force for complementary angles. The values for the drag force are presented in the table.

Right wing	Left Wing	Speed (m/s)	Angle (°) Left	Angle (°) Right	Lift Force Difference (ΔF)
-227.02	-104.33	5	-10	0	122.59
-146.21	-104.33	5	-5	0	41.78
-104.33	-104.33	5	0	0	0
42.35	-104.33	5	5	0	146.66
99.48	-104.33	5	10	0	203.81

Figure 8: Lift generated when the right wing is fixed, and the other is angled

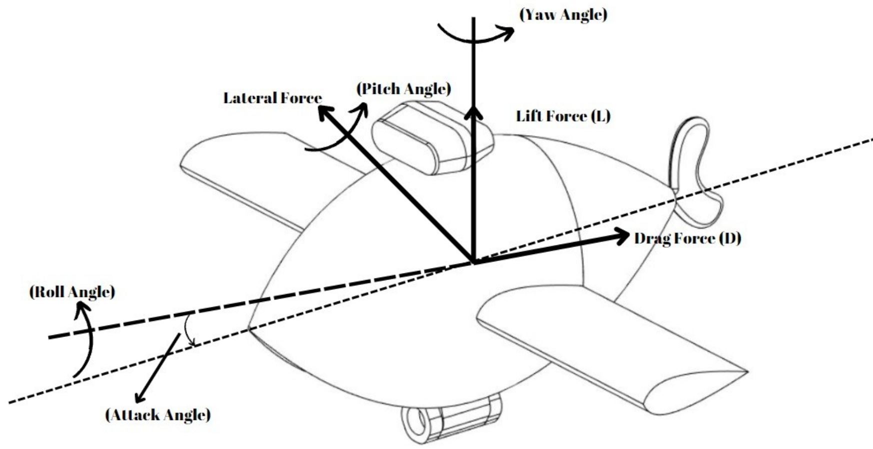


Figure 9: Names of the axes measured during movement

In the lower part of the body, an indentation in the size of 10% of the body diameter was created. This indentation continued along the entire body. A slot for the waterproof brushless motor was added to this indented section. The dynamic flow geometry of the model was prevented from being distorted by indenting and adding a motor housing. The drag force and the weight of the vehicle were taken into account when choosing a motor. The propeller thrust analysis was performed using ANSYS, the number of rotations of the engine and the propeller geometry were considered. The operating percentage of the engine was determined by comparing the thrust values in Figure 10 and the drag force acting on the vehicle at select speeds.

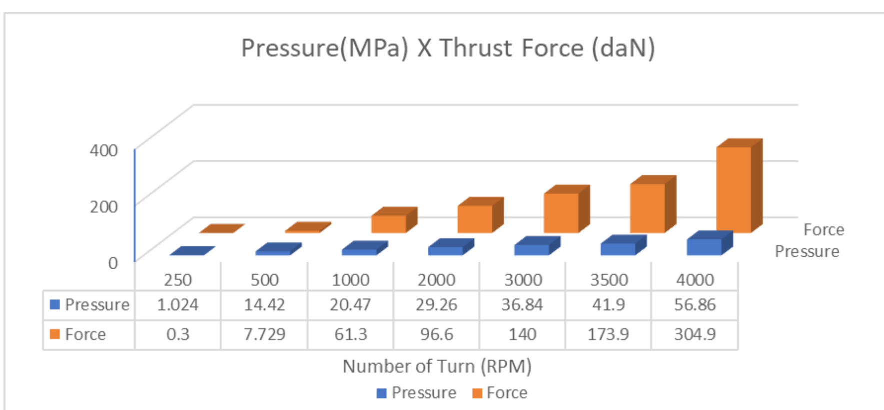


Figure 10: Thrust generated by the number of rotations of the engine and the pressure across the propeller

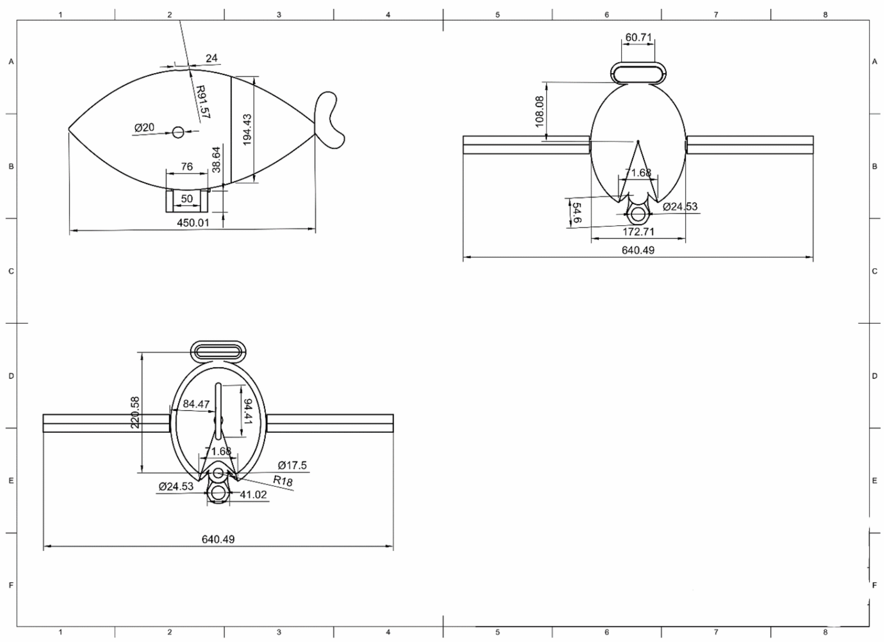


Figure 11: Technical drawing of the main body of the vehicle

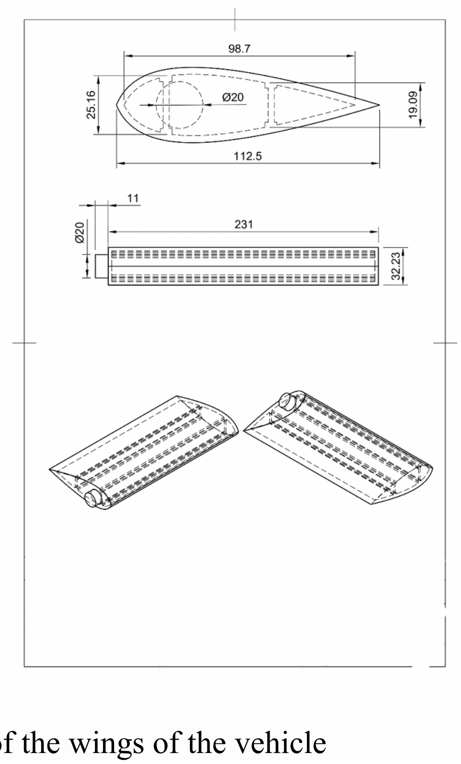


Figure 12: Technical drawing of the wings of the vehicle

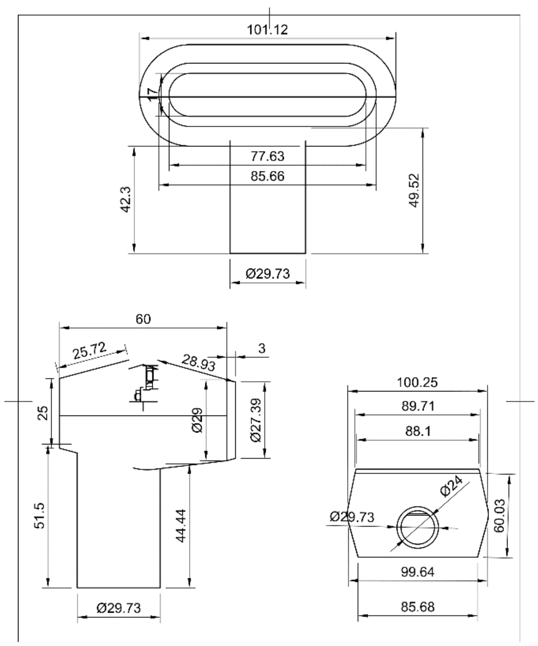


Figure 13: Technical drawing of the camera mount of the vehicle

A waterproof brushless motor with a power of 20,000 RPM was used. Since 20,000 RPM is a very high rotational speed, the thrust force is high. 20,000 RPM could only be achieved in the air since the density of the air does not create high pressure across the propeller during rotation. When run underwater, the motor could be used at a maximum of 3,500 RPM. Any value above 3.500 RPM created a high pressure across the propeller, resulting in the propeller breaking apart. The thrust of the engine was calculated in the range of 0-3,500 RPM in water. The thrust table and propeller pressure analysis are shown in Figure 11.

The pressure the propeller had to endure when spun with 3500 RPM was calculated at approximately 45 million Pascals. The resistance of ABS plastic against pressure is 49,986,990 pascals. As the pressure generated is close to the maximum resistance, the propeller encounters the risk of breakage. Hence, the maximum speed of the propeller was set to 3500 RPM.

The sound output of submarines ranges between 40 and 70 decibels. Since these submarines are mostly undetectable under most circumstances, it was reasonable to assume that the proposed vehicle would also be undetectable if sound output was below 70 dB. In order to ensure the invisibility of the model, it was necessary to provide movement below that value. For this reason, the acoustic power of the propeller in the model was calculated together with the ocean noise. The value turned out to be approximately 55 dB, which means that the sound generated by the vehicle is insignificant.

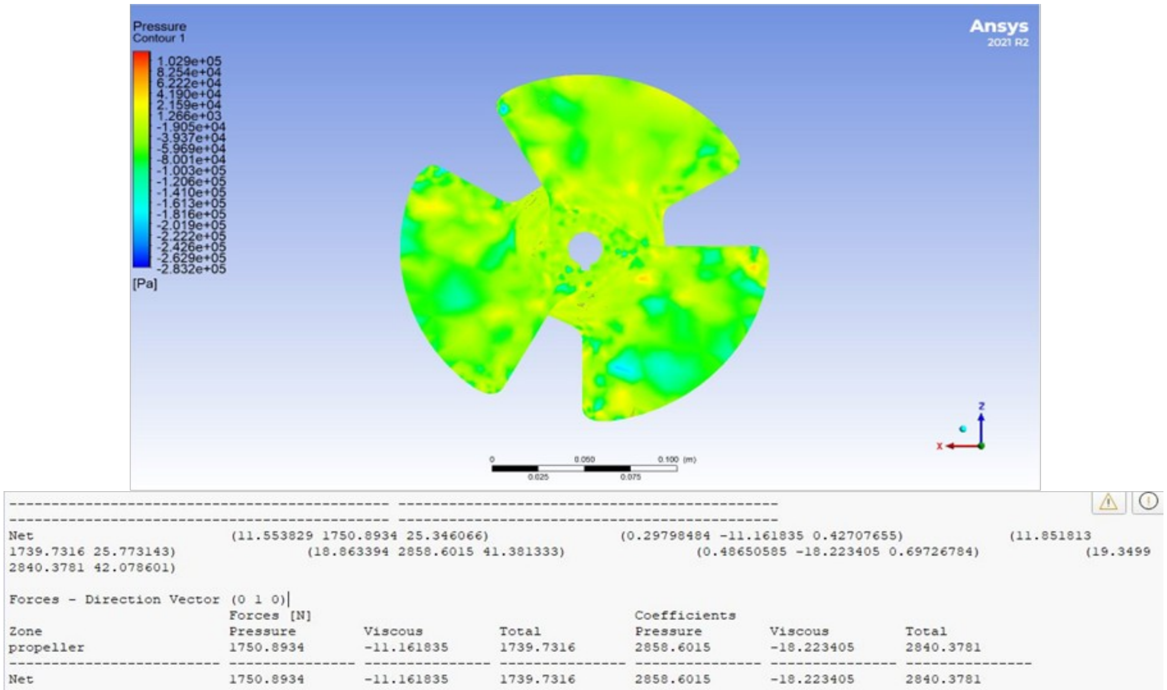


Figure 14: Analysis of the pressure generated across the propeller and the thrust of the engine

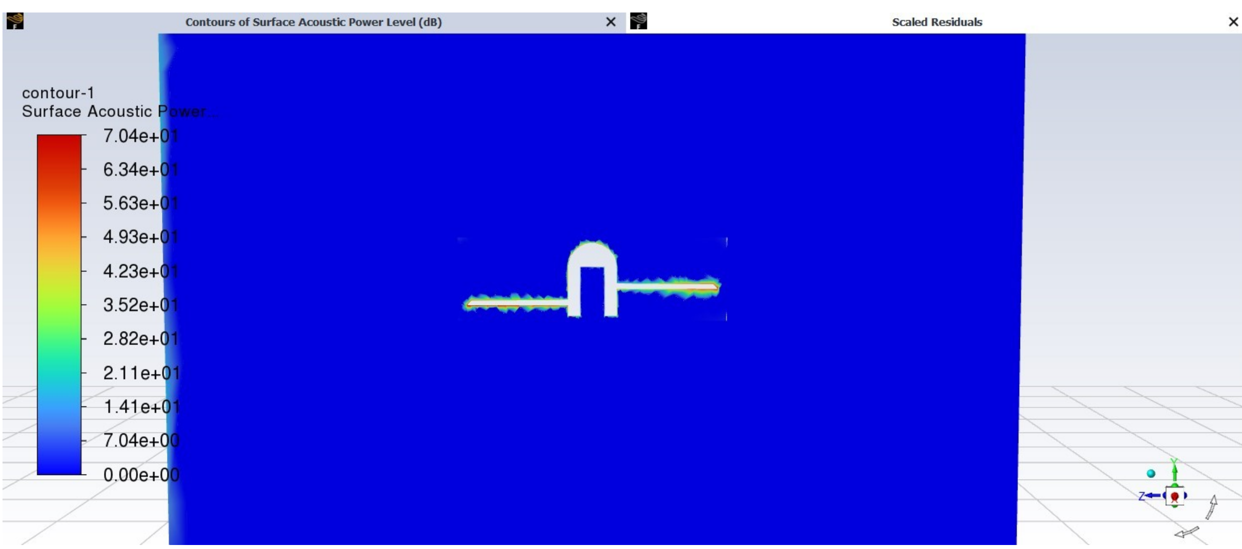


Figure 15: Acoustic analysis of the propeller

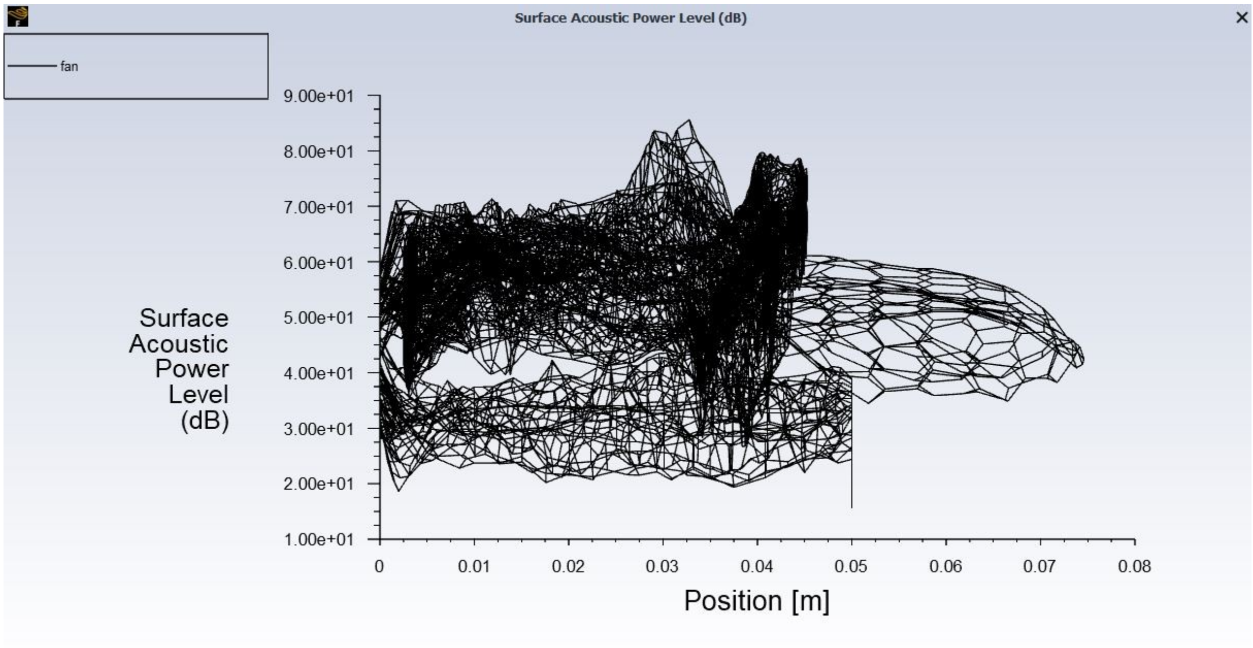


Figure 16: Acoustic analysis of propeller mixed with ocean noise

The route the vehicle takes is programmed using an algorithm. In order to create the route, the speed of the vehicle in the water needed to be determined in seconds. A flow sensor was used to measure the vehicle's velocity relative to that of water. The water passing through the flow sensor per second was measured in L/min. This value was later converted to m/s. The algorithm calculated how many meters the vehicle travels in the desired direction by collecting these values. This is done by measuring the heading angle using the (Inertial Measurement Unit) IMU sensor and calculating the sine and cosine values of that angle. The velocity is measured using the flow sensor in 0.1 seconds and is multiplied by the sine and cosine values. The x and y coordinates can therefore be calculated in 3D space.

The algorithm also keeps track of the time it takes to travel this distance. The algorithm constantly creates and updates a route. If the vehicle goes off-route for whatever reason, the algorithm backtracks the route it created. As it keeps track of all movement in any direction, the vehicle is able to retrace the route to the starting position by reversing the movements previously made. The vehicle is also able to connect to Wi-Fi, which can provide location information in close proximity. These features are especially important in information gathering missions.

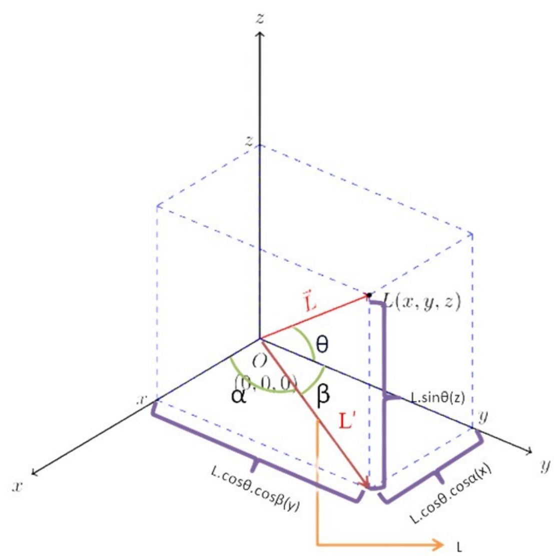


Figure 17: The formalization of route calculation without GPS using values from the flow sensor and motor speed

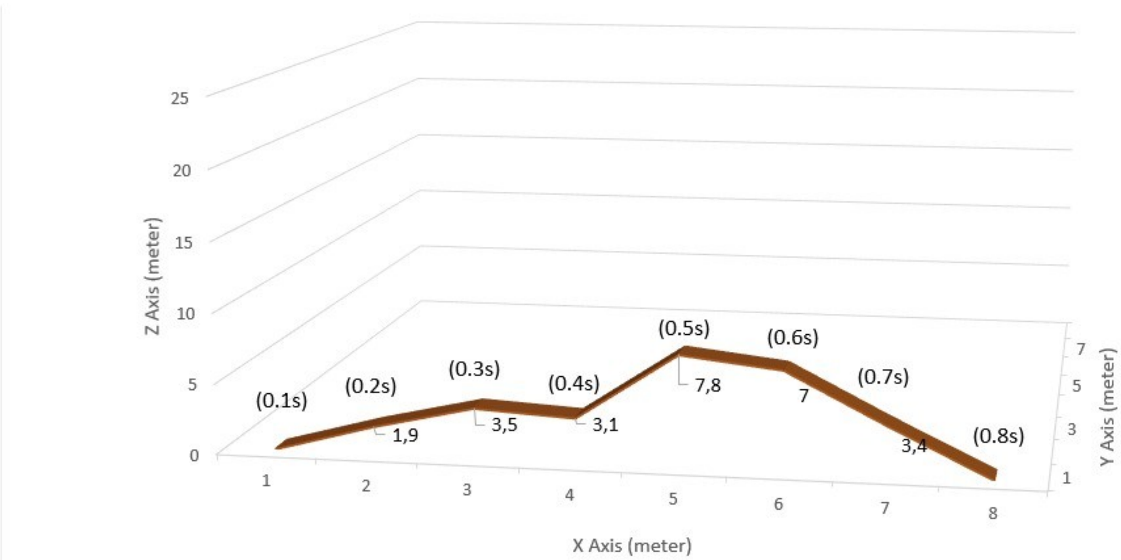


Figure 18: The position change route of the vehicle generated by the gyro IMU data. Position values are calculated in intervals of 0.1 seconds, the line represents the route taken in 3D space, each point in the 3D is calculated using the method presented in Figure 17. This figure is meant to be an example and is based on a hypothetical scenario.

A section to house the camera was designed at the top of the fuselage. The camera mount was designed in a removable manner. The mounting slot could be modified according to the camera model used. The camera mount is stationary, and the camera cannot move to change viewing angles. This disadvantage can be addressed in future versions of the project. The waterproof “Deneyap Camera” capable of infrared vision was used in the vehicle. Data can hence be obtained, regardless of light conditions and saved inside the vehicle. The surrounding elements are evaluated through the incoming images and data. Images are also analysed with artificial intelligence using the You Only Look Once (YOLO) algorithm (see below).

The camera is used to detect and identify threats through an image processing algorithm. In this case, enemy projectiles like torpedoes are defined as threats; this definition excludes marine life. Currently, there is no collision avoidance algorithm built in, which means that a collision with a sea animal could cause a deviation from the set course. This can be addressed in future versions of the vehicle. For image processing, the YOLO algorithm was used through Python and OpenCV, which is the most common Python library used for image processing. YOLO is an object detection algorithm utilizing convolutional neural networks. YOLO was preferred because of its capabilities to process data quickly. A model pre-trained to identify fish, and its corresponding weights files were used along with the algorithm. A new model from scratch was not trained due to resource and time constraints.

Fish are detected and identified using this software. The algorithm also gives a "confidence" value for the fish it detects, that is, a measure of certainty that the object was recognized correctly. The higher this value, the higher the probability that the object was correctly identified.

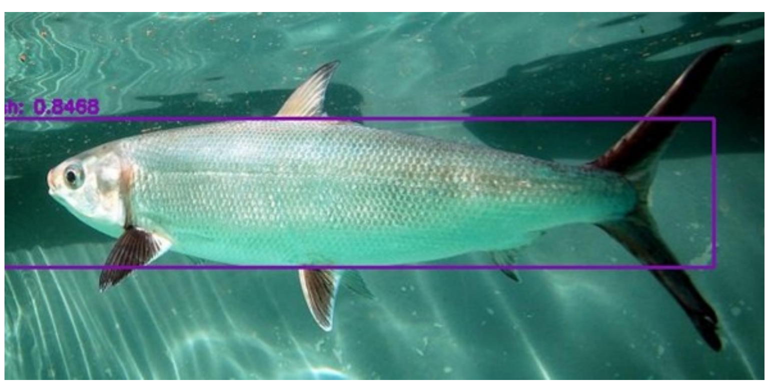


Figure 19: Underwater image processing with OpenCV and YOLO (14)

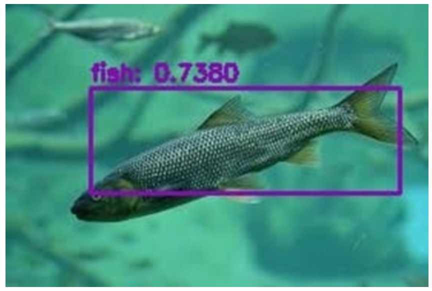


Figure 20: Underwater image processing with OpenCV and YOLO (15)

The body of the underwater vehicle was produced in two parts to ensure watertightness. The first part was the front body, and the second part was the rear body. A 20 mm shoulder length was designed between the bodies. The bodies are fixed to each other with 6 3mm screws. During fixing, a rubber seal was placed between the two bodies to ensure watertightness. The connection of the camera section to the body was designed as a tight fit (H8-n6: Hard-to-hammer). Thus, watertightness was also ensured in this part. A tight fit tolerance was also used in the wing parts to achieve watertightness however, a slightly lower tolerance was used as there is continuous movement in the wings (H7-f7: Moving engagement). Since the wing parts are movable, the possibility of water entering from the connection point is relatively greater. For this reason, the servo motors used were waterproof. A casing was designed around the servos to prevent the incoming water from damaging the electronic parts. A cover was designed to prevent the passage of water to the body. The servo was mounted with 4 screws. Then the cover was fixed with 4 screws. Servo connections were made with the waterproof connection tool.

Rounding rate and dimensions were adjusted according to the tests performed on the tail. As a result, a fixed tail section was created at the rear of the vehicle. The tail acted as a balncing mechanism due to the fluid being distributed equally on both lateral sides of the vehicle. The tail also mimicked the fish geometry in appearance.

As an assembly strategy, the flow sensor, along with its connections, was first placed in its slot on the front body. The motor, along with its connections, was placed in the motor housing located in the upper part of the sensor. This was followed by the placement of the waterproof ESC module and connections controlling the main motor. The battery was placed on the ESC. The connections of the flow sensor were made to the ESP32 card. Then the servos were placed, their connections were made, and the covers fixed. Three of the GY-85 modules that were used in the IMU were fixed in their slots and connected. The camera was placed at the top and its connections were made through the cable channel. After the assembly of the electronic parts was completed, the front body and the rear body were attached to each other and fixed with the help of screws. Finally, the wings were fixed to the servos.

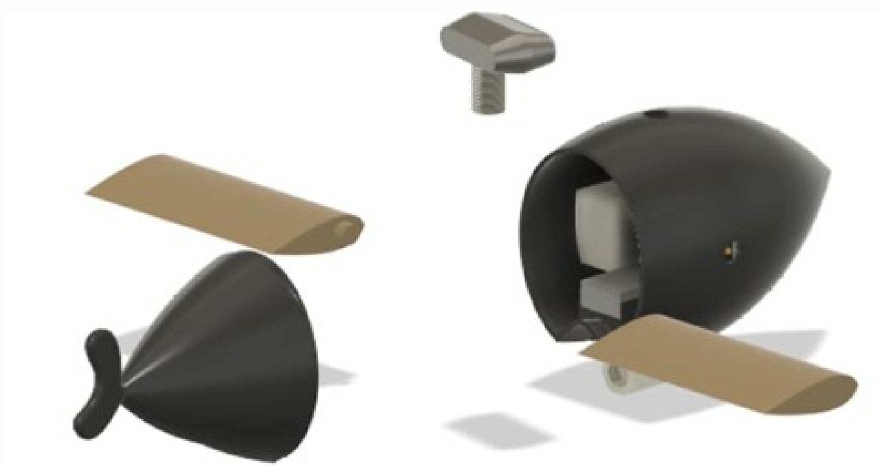


Figure 21: Blasted assembly image

When a threat is detected, the vehicle is designed to perform the escape maneuver. During this maneuver, the angles of both wings are increased so that they are brought to the desired 45-degree launch angle. After reaching the launch angle, the vehicle is launched out of the water using full power on the motors. As a result of being launched out of the water, it moves in the air for a while with a horizontal shooting motion. After this movement, it dives into the water with the same angle as the launch angle. Such a maneuver can be used to deactivate the threat. As a result of the maneuver, the threat (such as a torpedo) is left out of range for a while which can result in a successful evasion of the threat.

The vast majority of torpedoes activate their sonar when approaching the target to reduce energy use, which allows the torpedo to use its energy more efficiently. As a result of continuous escape maneuvering of the vehicle, the torpedo is forced to continuously operate its sonar. Thus, the torpedo can be rendered ineffective by fatigue and battery depletion.

The wings are moved during movement to prevent breakage. Aircrafts usually have flaps and ailerons at the rear of their wings to control direction. These can create the necessary difference of speeds of the air to provide rotation. Since the densities of water and air are very different, flaps and ailerons cannot withstand the pressure underwater. For this reason, the entirity of the wing surface was moved. Thus, PLA Plastic, which is the material of the wing, can withstand the forces it is exposed to. (PLA Strength: 7250 psi = 49.9 kiloPascal)

Figure 22: Escape maneuver diagram (16, 17)

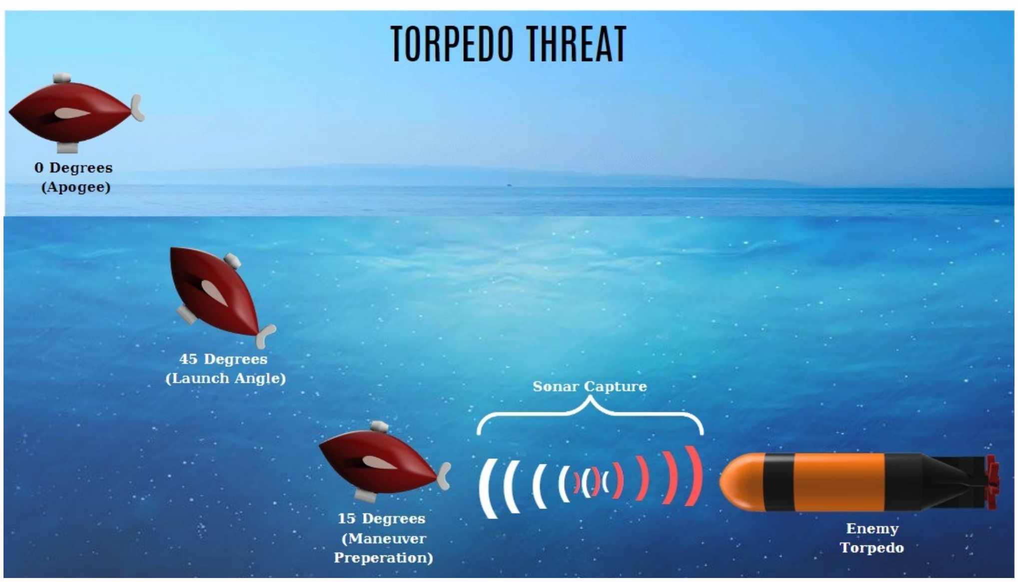
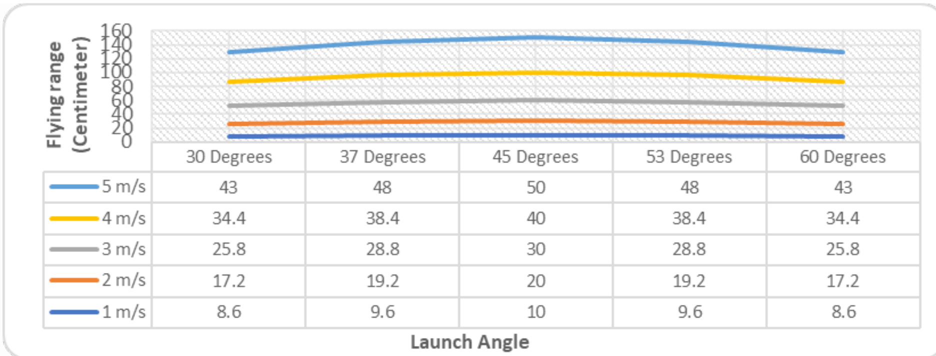
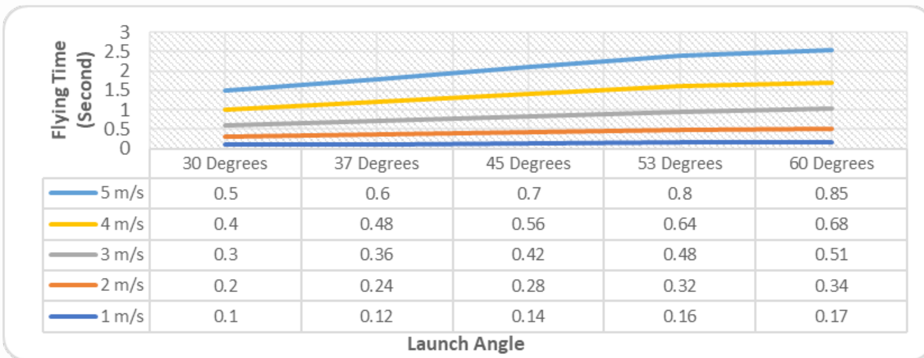


Figure 23. Torpedo prevention scheme (16-18)



	30 Degrees	37 Degrees	45 Degrees	53 Degrees	60 Degrees
5 m/s	43	48	50	48	43
4 m/s	34.4	38.4	40	38.4	34.4
3 m/s	25.8	28.8	30	28.8	25.8
2 m/s	17.2	19.2	20	19.2	17.2
1 m/s	8.6	9.6	10	9.6	8.6

Figure 24: Launch range calculated for various launch angles and vehicle speeds



	30 Degrees	37 Degrees	45 Degrees	53 Degrees	60 Degrees
5 m/s	0.5	0.6	0.7	0.8	0.85
4 m/s	0.4	0.48	0.56	0.64	0.68
3 m/s	0.3	0.36	0.42	0.48	0.51
2 m/s	0.2	0.24	0.28	0.32	0.34
1 m/s	0.1	0.12	0.14	0.16	0.17

Figure 25: Time in air calculated for various launch angles and vehicle speeds

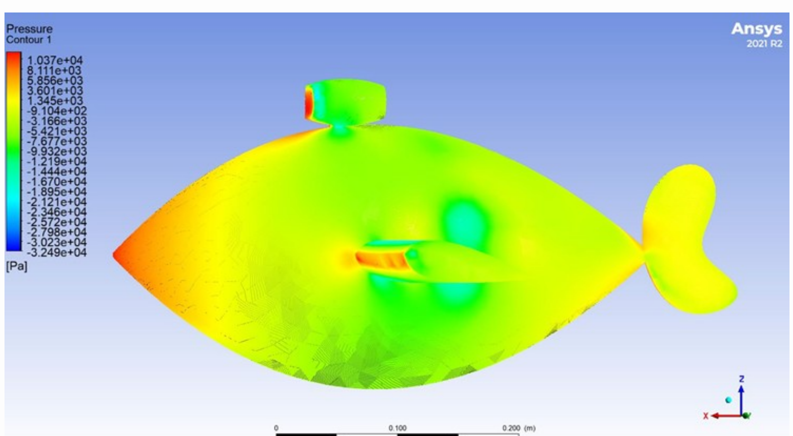


Figure 26: Pressure analysis in fluid

2.2 Electronics and software controls

2.2.1 Flow sensor (YF-B6)

The flow sensor provides an instantaneous estimate of the water passing through it in one minute in liters. The given value is processed and the speed of the vehicle in the water is calculated in m/s.

$$\frac{L}{min.} \times a = \frac{m^3}{s} \times \frac{1}{60.000} \times a \rightarrow \frac{L}{min.} \times a = m^2_{sensor} \times \frac{m}{s} \times a$$

The data coming from the sensor is instantly converted into m/s. Thus, the instantaneous speed of the fish is measured continuously. The same principle is used to calculate position in both water and air as the flow sensor is able to detect the amount of liquid passing through which includes air. Thus, along with the values from the Axis IMU Sensor, the position of the vehicle is calculated in both water and air.

2.2.2 Axis IMU sensor (GY-85)

Angles in the x, y and z axes are calculated with 1 GY-85 sensor. The change of direction of the fish is calculated using these values. These values provide the position of the fish in the water. Keeping the position of the vehicle in mind, the angles of the wings are changed with the help of servos. Thus, the position of the fish is brought to the desired values in the x, y and z axes. (Initial Values x:0, y:0, z:0)

2.2.3 Waterproof brushless motors

The motor operates at 20,000 rpm in air and 3,500 rpm in water. With its rpm and propeller geometry, it creates enough thrust to accelerate the fish to 5 m/s (18 Km/h). The motor's "rated voltage” is 11.1V, rated current is 17A and its rated power is 200W.

2.2.4 Battery

An 11.1V 3S Lipo 6200 mAh 45C battery was used to power the vehicle. It has enough power to feed the sensors and motors used. Its size and dimensions are suitable for use with the geometry of the vehicle. Motor connections are made with waterproof ESC converter, sensor connections are made with power distribution board (Matek PDB-XT60 Bec 5V & 12V).

2.2.5 ESC

Motor control was provided by the waterproof ESC 30. Thanks to the ESC, the motor is prevented from being damaged in case of sudden changes and excessive current. At the same time, it prevents the contact of the battery with water as a connection between the motor and the battery. ESC determines the motor speed and provides movement at a desired speed.

2.2.6 Servo motor

Voltage:4.8V-6.8V, Speed (6.8V): 0.14sec / 60° no load, Stalling Torque (6.8V): 21.5 kg / cm (DS3218MG) waterproof servo was used. The lift and drag forces applied to the wings were taken into account in the servo selection.

2.2.7 ESP32 Card

Flow sensor (YF-B6), ESC, GY-85 and servos are connected to the ESP32 Card, Espressif Systems, China; the card provides control over these sensors. The ESP32 Card supports 3.3V and 5V output voltage and 3.3V – 5.5V input voltage. All digital output pins support PWM and have LSM6DSM, 3-Axis Gyros and 3-Axis Accelerometer. Thanks to these features, it meets all the needs for the movement of the fish.

2.2.8 Raspberry Pi camera

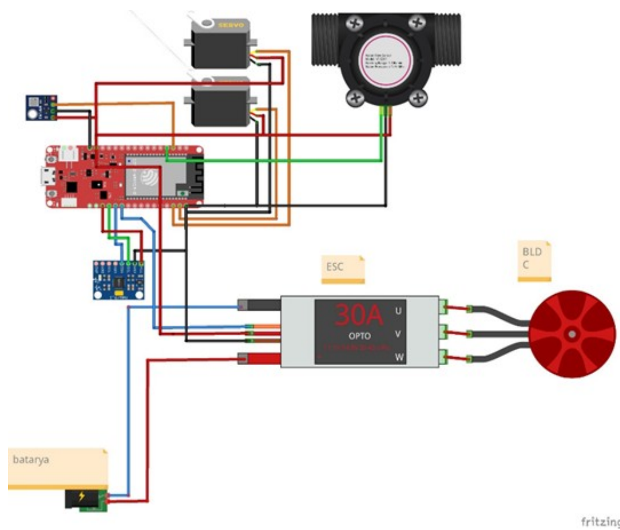


Figure 27: ESP32 card and sensor circuit diagram

The Raspberry Pi camera features high quality image detection, 8-megapixel fixed focus, 1080p, 720p60 and VGA90 support, Sony IMX219PQ CMOS image sensor and 15-pin ribbon cable. Images are taken from the Raspberry Pi Camera continuously. Incoming images are stored on the SD card on the Raspberry Pi 4. The stored images are processed using the image processing method described above. Thus, underwater mapping and ship/torpedo detection operations are performed.

2.2.9 Raspberry Pi 4

Broadcom 2711 processor, 2-line MIPI CSI camera port, 5V power input via USB-C (minimum 3A) was used and the operating ambient temperature is 0 – 50 °C. Raspberry Pi 4 was used in camera control due to its micro-SD card slot for data and operating system storage.

2.2.10 Depth sensor

“5837 Water Pressure Sensor MS5837-30BA” depth sensor was chosen. Power 3-5V was used in the circuit since the communication is I2C IIC compatible. Its 2mm sensitivity reduces the margin of error in the underwater route creation task. The up and down movement of the fish is controlled by this sensor.

Codes and commands are remotely sent to the vehicle over Wi-Fi using Over-The-Air (OTA) and ESP32 in order to control the vehicle. Different codes for different tasks are sent over the same Wi-Fi with OTA. After the vehicle completes its task and returns to the control center, it connects to Wi-Fi and transmits information like images, route information, energy consumed and past movements. This information is displayed in the interface. Since Wi-Fi is not used during tasks, the visibility of these transmissions is not a concern.

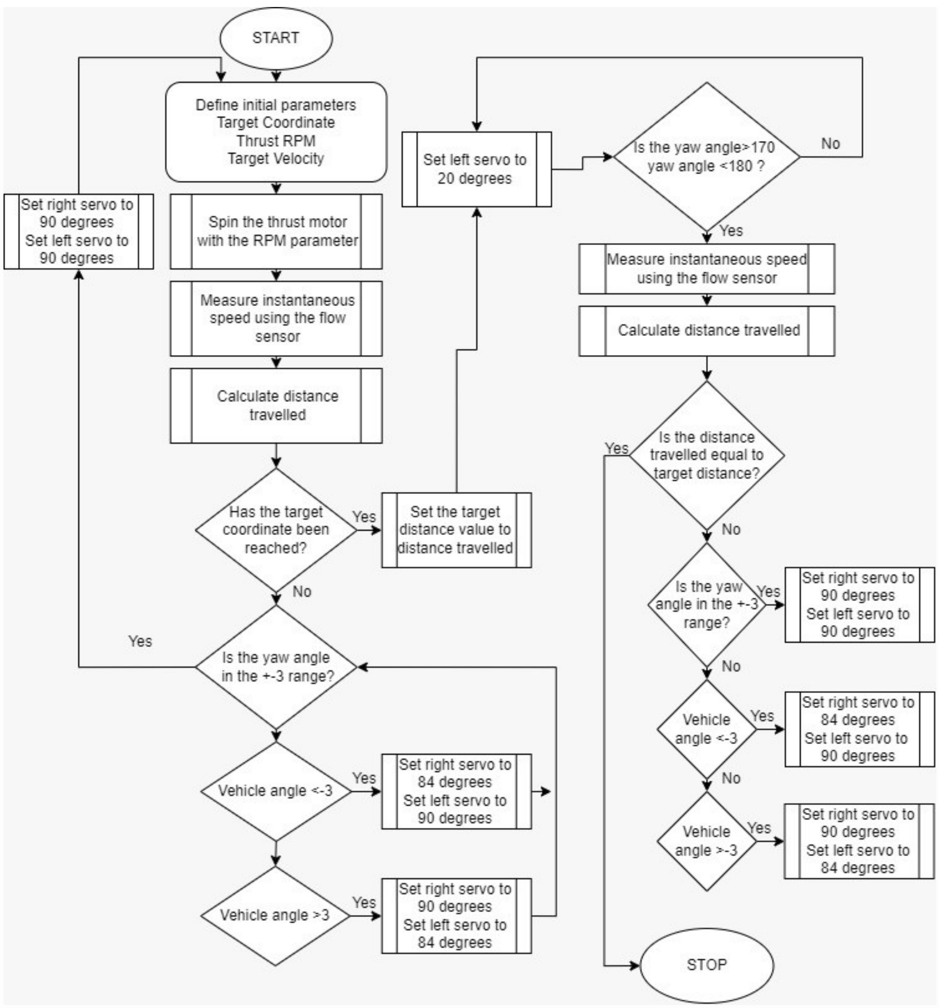


Figure 28: Algorithm and flow chart used during autonomous movement

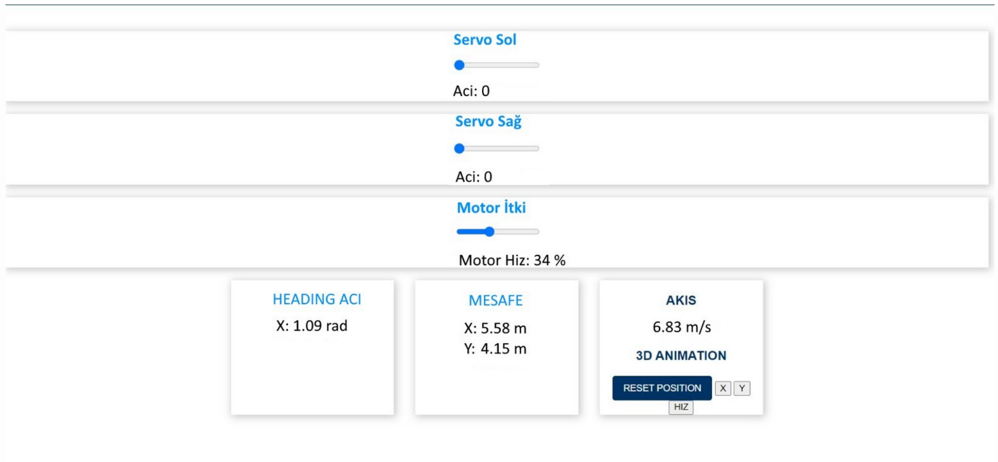


Figure 29: User interface

```
#include <Servo.h>
Servo ESC;
int potValue;
void setup() {
  ESC.attach(9,1000,2000);
}
void loop() {
  potValue = analogRead(A0);
  potValue = map(potValue, 0, 1023, 0, 180);
  ESC.write(potValue);
}
```

Figure 30: Brushless motor control code with ESC and battery

```
/*
Vcc ve GND yi bağladıktan sonra pini 2 ye bağla.
 */
byte statusLed    = 13;
byte sensorInterrupt = 0;  // 0 = digital pin 2
byte sensorPin       = 2;
// saniyede 4,5 cevap alınır.
// litre/dakika cinsinden
float calibrationFactor = 4.5;
volatile byte pulseCount;
float flowRate;
unsigned int flowMilliLitres;
unsigned long totalMilliLitres;
unsigned long oldTime;
void setup()
{
  // serial begin belirlenir(9600)
  Serial.begin(9600);
  pinMode(statusLed, OUTPUT);
  digitalWrite(statusLed, HIGH);
  pinMode(sensorPin, INPUT);
  digitalWrite(sensorPin, HIGH);
  pulseCount        = 0;
  flowRate          = 0.0;
  flowMilliLitres   = 0;
  totalMilliLitres  = 0;
  oldTime           = 0;
  attachInterrupt(sensorInterrupt, pulseCounter, FALLING);
}
```

Figure 31: Code-1, calculating velocity using flow sensor output

```
void loop()
{
  if((millis() - oldTime) > 1000)
  {
    detachInterrupt(sensorInterrupt);
    flowRate = ((1000.0 / (millis() - oldTime)) * pulseCount) / calibrationFactor;
    oldTime = millis();
    flowMilliLitres = (flowRate / 60) * 1000;
    // saniyede geçen suyu litre cinsinden bulmak için 60'a, saniyede geçen suyu milimetre cinsinden bulmak için de 1000 ile çarpın.
    totalMilliLitres += flowMilliLitres;
    unsigned int frac;
    Serial.print("Flow rate: ");
    Serial.print("m/s:");
    Serial.print(flowRate/31,8);
    //m³/sn cinsinden değeri m/s cinsinden bulmak için m²'ye bölmemiz gerekiyor. Sensörün kesitalanı yaklaşık 530mm²dir
    //ve gerekli işlemler yapıldıktan sonra L/dk cinsinden değerin m/s cinsinden olan değerin 31,8 katı bulunmuştur
    Serial.print("\t");
    Serial.print("Output Liquid Quantity: ");
    Serial.print(totalMilliLitres);
    Serial.println("mL");
    Serial.print("\t");       // Print tab space
    Serial.print(totalMilliLitres/1000);
    Serial.print("L");
    pulseCount = 0;
    attachInterrupt(sensorInterrupt, pulseCounter, FALLING);
  }
}
void pulseCounter()
{
  pulseCount++;
}
```

Figure 32: Code-2, calculating velocity using flow sensor output

```
#include <ServoESP32.h>
#include <PID_v1.h>
double Setpoint2;
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif
Servo solServo;
Servo sagServo;
int servoPinSol = D1;
int servoPinSag = D0;
float angle = 0;
long loop_timer;
int temp;
int state = 0;
unsigned long timer = 0;
MPU6050 mpu;
#define OUTPUT_READABLE_YAWPITCHROLL
#define INTERRUPT_PIN 2
#define LED_PIN 13
bool blinkState = false;
bool dmpReady = false;
uint8_t mpuIntStatus;
uint8_t devStatus;
uint16_t packetSize;
uint16_t fifoCount;
uint8_t fifoBuffer[64];
Quaternion q;
VectorInt16 aa;
VectorInt16 aaReal;
VectorInt16 aaWorld;
VectorFloat gravity;
float euler[3];
float ypr[3];
```

Figure 33: Code-1, controlling servos using IMU values

```
uint8_t teapotPacket[14] = { '$', 0x02, 0, 0, 0, 0, 0, 0, 0, 0, 0x00, 0x00, '\r', '\n' };
volatile bool mpuInterrupt = false;
void dmpDataReady() {
  mpuInterrupt = true;
}
void setup() {
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
  Wire.begin();
  Wire.setClock(400000);
#elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
  Fastwire::setup(400, true);
#endif
  solServo.attach(servoPinSol);
  sagServo.attach(servoPinSag);
  Serial.begin(115200);
  while (!Serial);
  Serial.println(F("Initializing I2C devices..."));
  mpu.initialize();
  pinMode(INTERRUPT_PIN, INPUT);
  Serial.println(F("Testing device connections..."));
  Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") : F("MPU6050 connection failed"));
  Serial.println(F("\nSend any character to begin DMP programming and demo: "));
  Serial.println(F("Initializing DMP..."));
  devStatus = mpu.dmpInitialize();
  mpu.setXGyroOffset(220);
  mpu.setYGyroOffset(76);
  mpu.setZGyroOffset(-85);
  mpu.setZAccelOffset(1688);
  if (devStatus == 0) {
    mpu.CalibrateAccel(6);
    mpu.CalibrateGyro(6);
    mpu.PrintActiveOffsets();
    Serial.println(F("Enabling DMP..."));
```

Figure 34: Code-2, controlling servos using IMU values

```
    mpu.setDMPEnabled(true);
    Serial.print(F("Enabling interrupt detection (Arduino external interrupt "));
    Serial.print(digitalPinToInterrupt(INTERRUPT_PIN));
    Serial.println(F(")..."));
    attachInterrupt(digitalPinToInterrupt(INTERRUPT_PIN), dmpDataReady, RISING);
    mpuIntStatus = mpu.getIntStatus();
    Serial.println(F("DMP ready! Waiting for first interrupt..."));
    dmpReady = true;
    packetSize = mpu.dmpGetFIFOPacketSize();
  } else {
    Serial.print(F("DMP Initialization failed (code "));
    Serial.print(devStatus);
    Serial.println(F(")"));
  }
}
void loop() {
  if (!dmpReady) return;
  if (mpu.dmpGetCurrentFIFOPacket(fifoBuffer)) {
#ifdef OUTPUT_READABLE_YAWPITCHROLL
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
    float yawAci = ypr[0] * 180 / M_PI;
    Serial.print(ypr[0] * 180 / M_PI);
    Serial.print("/");
    Serial.print(ypr[1] * 180 / M_PI);
    Serial.print("/");
    Serial.println(ypr[2] * 180 / M_PI);
    Setpoint2 = 20;
    double farkAci = yawAci - Setpoint2;
    Serial.print("giris:");
    Serial.println(yawAci);
    Serial.print("cikis:");
    Serial.println(farkAci);
    if (farkAci < -3)
    {
      solServo.write(20);
      sagServo.write(120);
    } else if (farkAci > 3)
    {
      solServo.write(120);
      sagServo.write(20);
    } else
```

Figure 35: Code-3, controlling servos using IMU values

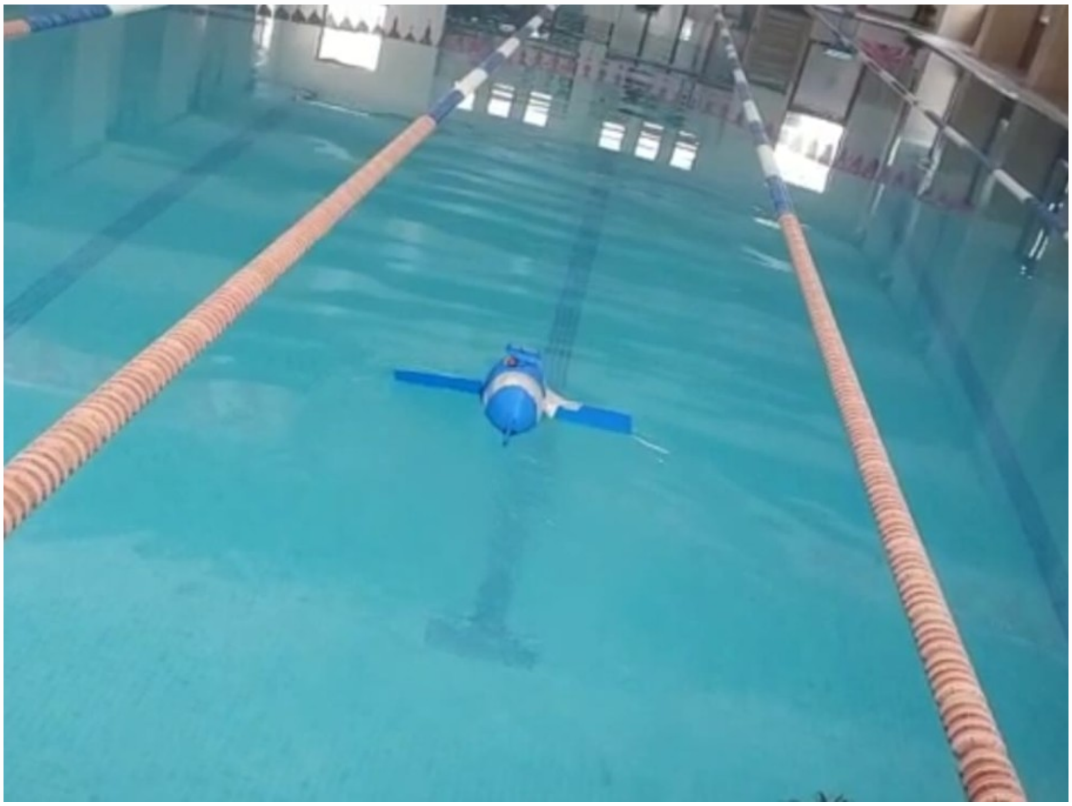


Figure 36: Test in Angora Fine Arts High School pool

Figure 37: Test in the Şehit Göksel Koç dam

3. Results and Discussion

The desired tasks of the underwater spy fish (reaching a certain point with its coordinates fully autonomously and without GPS, taking and processing images from any point on the way, mapping the route it takes on the way back and forth, successfully getting rid of the threat by activating the escape mechanism in case of danger) was successfully completed and tested. During the test, the spy fish traveled fully autonomously in the desired direction from the beginning of the pond to the end, and at the same time successfully fulfilled the desired task by taking and processing images. In a possible torpedo attack, thanks to the escape mechanism, the spy fish will jump out of the water at certain intervals, glide for a while, then dive again, consuming the battery of the torpedo and warding off danger. The vehicle's operating range is calculated to be around 10 kilometers.

4. Suggestions

The project can be used not only for espionage, but also for tasks such as security, underwater mapping, support to surface units, suicide missions by making some minor changes in the design and materials. The vehicle can be scaled to different sizes in order to fit the needs of different missions, however, some of the analyses would have to be remade. A stronger motor might need to be used if the vehicle is upscaled. Another material than ABS plastic could also be needed, depending on the results of the analyses.

An analysis of the frequencies of the waves generated by the propeller can be made using parameters of the propeller, water, and the motor. Since this calculation is very advanced and would require professional expertise in waves, the analyses cannot currently be made.

The image processing capabilities can be vastly improved using the ZED 2 camera and Jetson Nano Card. Using this hardware, the instantaneous speed of nearby sea animals and underwater vehicles can be detected. Using this technology, the species of the detected sea animals and the types of vehicles can be identified. This hardware can also scan the area and create a 3D map. The ZED 2 camera is able to calibrate itself, which eliminates the need to recalibrate when entering/exiting water. More capable IMU sensors can be used to decrease the rate of error in longer run times. However, this would also increase the production cost.

The current model has a few prominent flaws. Firstly, the camera mount is not movable. A movable camera system with a motor can be developed in order to provide flexibility and increase the field of view.

Another flaw is the lack of a collision avoidance algorithm. A collision avoidance algorithm can be developed in order to avoid any disturbances to the route in case of any collisions with marine animals. Finally, the model is fragile due to the use of ABS plastic, which was chosen due to budget concerns. In future models, composite material can be used to increase the durability of the vehicle.

5. References

1. Otonakit. Otonom Nedir?: Otonom Araçların Çalışma Prensibi (November 2022). Retrieved from https://www.otonakit.com/blog-detay/otonom-nedir-otonom-araclarin-calisma-prensibi

2. Çıtak, E. Savaş Stratejisinde İstihbaratın Önemi (September 30, 2019). Retrieved from Bingöl Üniversitesi Sosyal Bilimler Enstitüsü Dergisi at http://busbed.bingol.edu.tr/en/download/article-file/835616

3. Advanced Navigation. Hydrus: The Drone Revolution in the Deep (June 2022). Retrieved from https://l24.im/nqO

4. Boston Engineering. Imagine a Robotic Fish that Defends Against Underwater Threats - Boston Engineering Case Study (January 2023). Retrieved from https://l24.im/m9BgMhj

5. Denli, İ. GPS Nedir, Nasıl Çalışır, Açılımı Ne Anlama Geliyor? (2022). Retrieved from Webtekno at https://www.webtekno.com/gps-nedir-nasil-calisir-h113752.html

6. Aksoy, S. Yakalama (Intercept) Sonarları. Retrieved from Gebze Teknik Üniversitesi Su Altı Akustik Dalga Yayılımı ve Sonar Sistemler at https://l24.im/kaTOIX

7. Kline J, Haitz R. Principles and Application of Underwater Sound. Chapter 3: The Sonar Equation. (1950) Retrieved from: https://maritime.org/doc/sonar/chap3.php.

8. White, R. How do modern warships defend themselves against torpedoes?. Naval Post. (2021, March 21). Retrieved from https://navalpost.com/how-modern-warship-defend-themselves-against-torpedo/

9. ASELSAN. Sats - Su Altı Akustik Tespit Sistemi (January 2023). Retrieved from ASELSAN Deniz Sistemleri at https://l24.im/vMwsEtc

10. Aydınlık. Dev Adım! İşte ASELSAN Intercept Sonar Sistemi (May 30, 2018). Retrieved from https://www.aydinlik.com.tr/haber/dev-adim-iste-aselsan-intersept-sonar-sistemi-92317

11. Tasarımdan İmalata. Geçme toleransları. Retrieved from https://www.tasarimdanimalata.com/gecme-toleranslari/

12. 3 Eksen Teknoloji. ABS ve PLA filament çeşitleri arasındaki fark nedir? (2021, November 1). Retrieved from https://l24.im/ntJ

13. Langnau, L. How Tensile Strength Relates to 3D Printing (September 4, 2019). Retrieved from Make Parts Fast at https://l24.im/isROEQ

14. SEAFDEC. Farming Milkfish (January 2023). Retrieved from SEAFDEC/Aquaculture Department at https://www.se

15. Wallpaper Flare. Milk fish while under water (2023, January). Retrieved from https://www.wallpaperflare.com/milk-fish-under-water-photo-aquatic-animals-fish-animal-themes-wallpaper-dbpq

16. Pıxels. Underwater Near Ocean Surface with Rising Bubbles in Blue Sea is a Piece of Digital Artwork (May 17, 2020). Retrieved from IncrediVFX Store at https://l24.im/oa32

17. WEForum. Sea level rise: Everything you need to know (2022, September 29). Retrieved from https://www.weforum.org/agenda/2022/09/rising-sea-levels-global-threat/

18. Öztürk, S. Toy Torpedo Model (June 12, 2022). Retrieved from Grabcad at https://l24.im/c1LxIw3