ROBOMARINE
UNDERWATER ROV

Table of Contents

Appendix

About the Team

Our team has four members - Aditya Iyer, Kreneal Shah, Krish Shah, and Vedant Dadhaniya. We have all been interested in robotics since a very young age, and throughout the years we have participated in many robotics competitions like the First Tech Challenge and the World Robot Olympiad. In November 2022, we began this project to make a highly versatile underwater robot. In the span of making this robot, we have been guided by the coaches and instructors at The Robotronics Club, a robotics centre in Ahmedabad.

Aditya Iyer is a robotics enthusiast and has been learning robotics since 2017. He is a multi-time participant in the World Robot Olympiad since 2018. With an interest in Programming and Engineering Design, he has several projects under his belt.

His previous projects include Autonomous Trolley robots, Infrared-based Headgear for the blind, and IoT-monitored greenhouses. He has also developed many programs in the field of algorithms, networking and encryption. For example, the Pattern Predictor, TSP-Solver, Chunk Calculation Optimizer, and many more.

He is the author of all the RoboMarine programs and currently manages the GitHub repository.

Kreneal Shah has been fascinated with robots since a young age. He started robotics in 2015 and has competed in several WROs since 2018.

He has created projects like Automatic waste disposal system, Hands Free shopping cart for the same.

Kreneal is responsible for the Hardware of the ROV.

Krish Shah is a student at Ahmedabad International School and has been learning robotics since 2015. He has participated in many World Robotics Olympiads since 2016. He has also participated in tournaments like First Lego League and First Tech Challenge.

He is passionate about designing, building, and programming machines that can perform various tasks autonomously or under human control. Krish has always tried his best to make a change in the society.

Getting an opportunity to make a change really interested him in joining the RoboMarine team. He manages the hardware and mechanics of the ROV.

Vedant Dadhaniya is a STEM enthusiast with special interest in programming and robotics. He has been pursuing robotics for 6 years, recurringly participating in competitions like the World Robot Olympiad and World STEM and Robotics Olympiad. Having learned programming languages like Python, Java, C# and platforms such as Creo, Blender, Unity, he has worked on multiple robotics and coding projects and otherwise.

He is also interested in chess in which he has ranked in the top 5 in multiple state and regional tournaments. He currently manages the design and hardware of the project.

Chapter 1: Introduction

Section 1.1: Intro to Underwater Robotics

Underwater Robotics encompasses all fields of robotics that deal with the construction and deployment of robots meant for complete aquatic submersion. As of now, there is little research done in this field compared to others, such as space.

"We have better maps of the surface of Mars and the moon than we do of the bottom of the ocean."

— Dr. Gene Feldman

RoboMarine is a step towards the research of underwater robotics. Through this project, the RoboMarine Team aims to provide a highly versatile tool for aquatic research and exploration.

Section 1.2: Origin of the idea

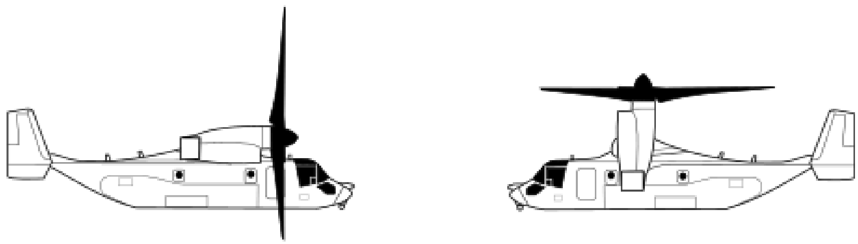
The idea for RoboMarine originated when the RoboMarine Team — a group of students at The Robotronics Club, Ahmedabad — attempted to make a robot that could move freely in all three axes while underwater. It was also decided that the robot would have an in-built camera to capture underwater footage.

There were multiple proposed designs for the robot, some of which will be displayed in the next section. These designs were all built with different purposes in mind, whether it be speed, precision, weight, or diving depth. Eventually, we settled on a standard design with 4 motors on the horizontal plane and 2 motors on the vertical axis; although this may be subject to change in future iterations of the robot.

Section 1.3: Ideation and Previous Ideas

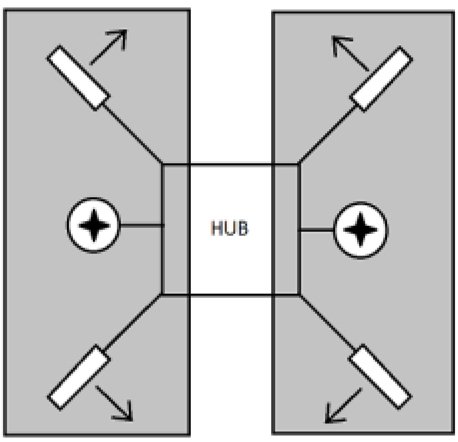
The first idea for the robot was a 3-motor design, with 2 motors in front and 1 at the back. All motors would initially face upwards. The motors in front would turn by up to 90 degrees both front and back. This would allow the front motors to output thrust in any direction. The back motor would provide an upward force equal to that provided by the frontal motors to ensure that the robot stays horizontal.

This model was inspired by the aerial vehicle V22-Osprey, which used propellers that could rotate from a forward position to an upward position. Below is a picture of the V22-Osprey, courtesy of Wikipedia.



The second idea included a six-motor design. Four motors would be placed on the horizontal plane, facing 45 degrees from the robot. The remaining two motors would be placed on the vertical axis to provide downward thrust to the robot. This idea is very similar to the current robot; the only difference is that this idea uses a plastic box hub and does not include a gyroscopic sensor, which the current robot does.

Attached below is the top-view diagram of the robot.

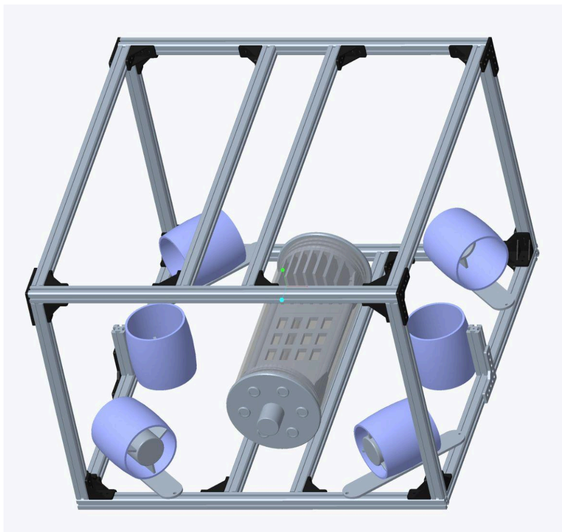


After testing the above robot in a controlled environment, it was noticed that the robot could not control its tilt. Hence it was decided that a gyroscopic sensor would be added. Moreover, the plastic hub was replaced with a transparent acrylic tube casing.

Chapter 2: Building the Robot

Section 2.1: ROV Anatomy

In this section we'll look at all the parts required in building the ROV (Remotely Operated Vehicle). We'll look at sections of the robot and the technicalities associated with each one of them. The ROV at its current stage of development consists of 2 units - the Diver and the Buoy. The Diver refers to the submersible part of the ROV which carries out the underwater exploration. The Buoy is the floating device linked to the Diver which floats on the surface and helps in communicating with the operator



1. The Frame

   This is the starting point. This is the main body of your build where all other parts of your build will be mounted. The frame is made of ultra-durable aluminium extrusion beams so budding operators can pilot without the risk of damaging the structure

   The frame is assembled using ABS right-angle connectors and hex nuts. The design frame size is 450mm x 420mm x 255mm. It consists of 12 420mm extrusion beams, and 4 225mm extrusion beams.

2. Thrusters

   These are the components that give your ROV the ability to move. Common brushless DC motors have an open casing, which cannot be used for an underwater purpose. For this we have used thrusters with brushless DC motors specially designed for underwater use. The motor is rated 1000kv. Don't confuse this kv with kilo-volts. Kv rating in dc motors refers to the RPM produced per 1 volt of input. So

In our case, if we supply 1V to our 1000kv motor, it will spin at 1000 RPM at full speed.
Note : lower kv motors produce much more torque, so they can spin larger propellers at greater speeds whereas larger kv rated motors produce low torque but spin smaller propellers at higher speeds

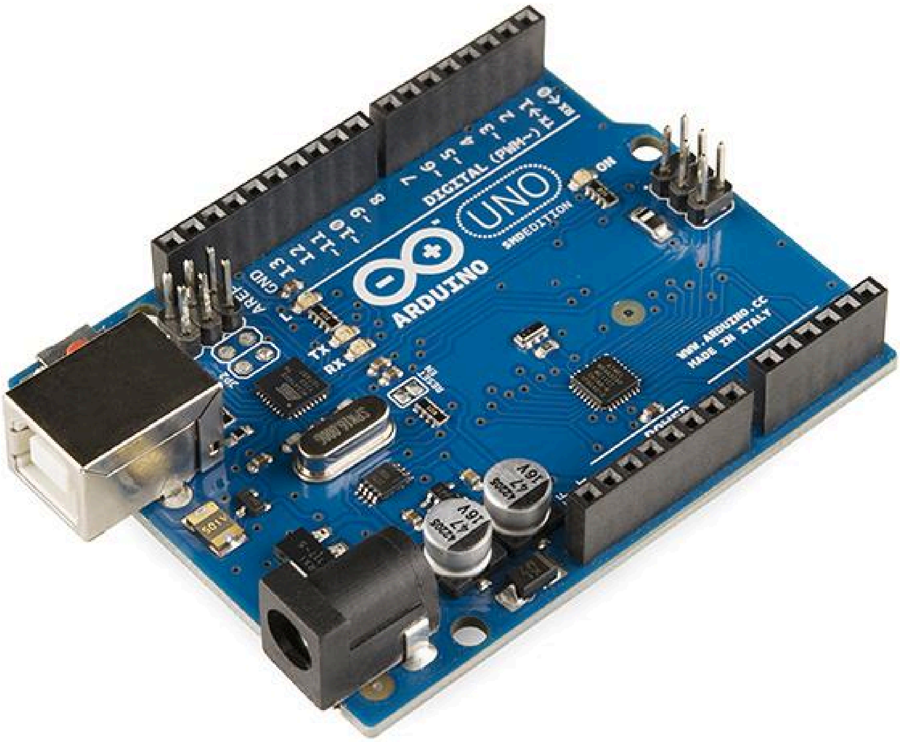
3. The Hub

The hub consists of all the sensors and electrical components of the diver including the ESCs, the Arduino UNO, receiver and battery. It is made up of an Acrylic Tube sealed shut by self designed aluminium caps which has slots for cables for communication and motors. The acrylic tube is loaded with specially designed 3D printed shelves. These shelves help in optimal organisation of internal electrical components.

4. ESCs

Electronic Speed Controllers or ESCs are the components that produces the three phase signal that is required to drive your DC motors. Each ESC at a time can only control one thruster. The ESC we are using is rated 40A.

5. Arduino UNO



Arduino UNOis like the flight controller of your underwater ROV. It is essentially the brain of the robot and where all the other components such as the sensors and ESCs will connect. The Arduino UNO is equipped with our self designed movement program which helps it read the receiver input, interpret it and run the thrusters accordingly

6. The Buoy

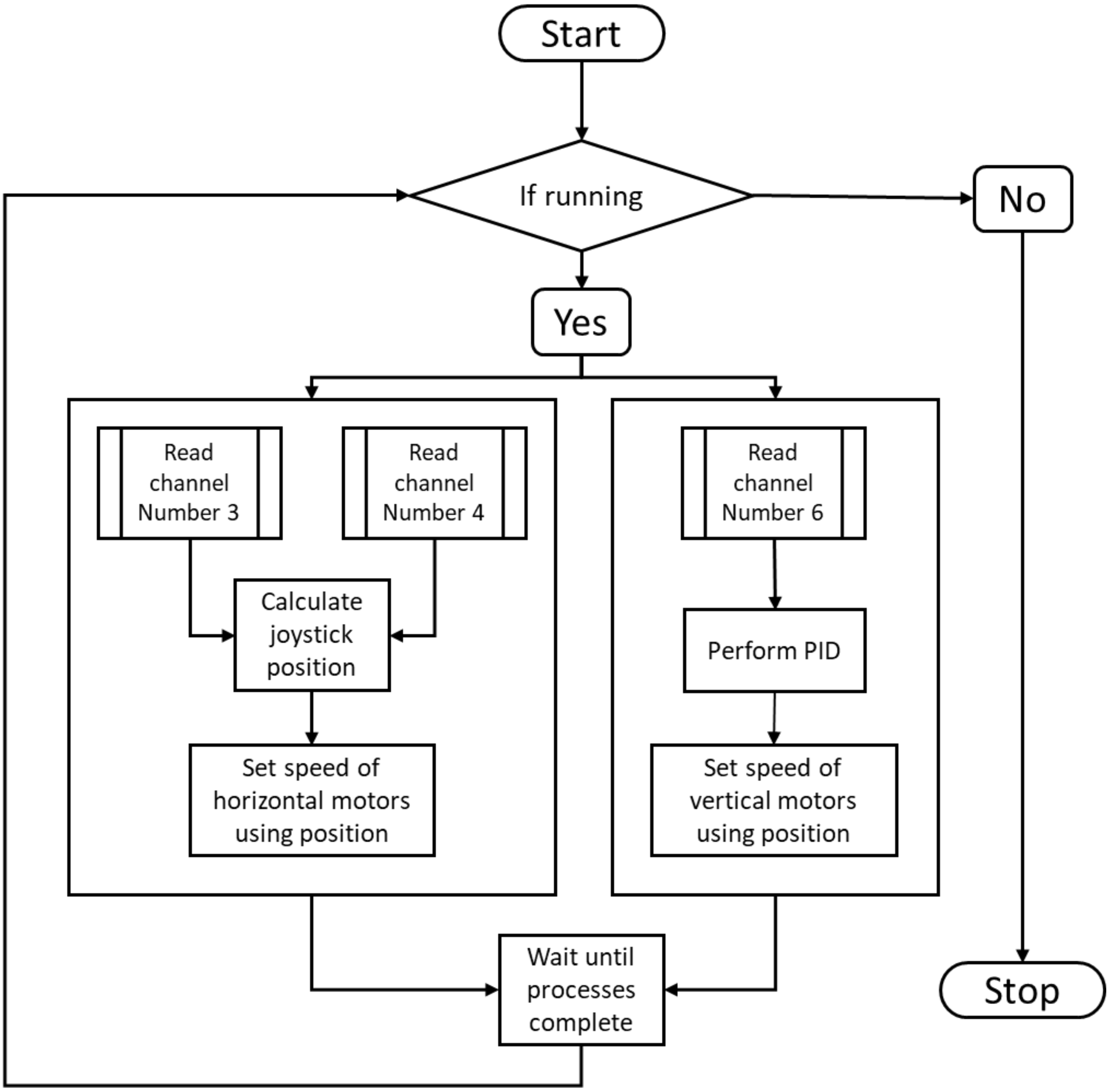
The Buoy is a disc-like 3D printed floating structure of the ROV. It consists of a receiver cable that links down to the diver using a cable. It can also be equipped with a GPS module to help the operator locate the ROV in case it is used in open sea.

Section 2.2: Inventory of Parts

- 12x Aluminium Extrusion Beam 15x15mm square profile 420mm
- 4x Aluminium Extrusion Beam 15x15mm square profile 225mm
- 2x Aluminium Extrusion Beam 15x15mm square profile 60mm
- 4x Brushless DC underwater thruster CW propeller
- 2x Brushless DC underwater thruster CCW propeller
- 6x Bidirectional Electronic Speed Controller
- 1x FS-CT6B Transmitter
- 1x FS-R6B Receiver
- 2x Arduino UNO
- 1x Acrylic Tube
- 2x Acrylic Tube Caps
- 1x MS5837 Pressure sensor
- 1x MPU6050 Gyroscope sensor
- Special 3D printed parts :
  Robomarine/3D-Parts: All design related Creo and stl files here (github.com)

Chapter 3: Program Design

Section 3.1: Overview of the Program



The above is a program flow diagram for the robot. There are two main parts of the program.

On the left, we read the signal from channels 3 and 4. Then we calculate the position of the joystick, which sets the speeds for the horizontal motors.

On the right, we read the value from channel 6 (the left knob of the CT6B transmitter). This is used to determine the depth. Then we perform a PID manoeuvre to ensure that the robot does not tile. Using the values from the PID, we set the speeds of the motors.

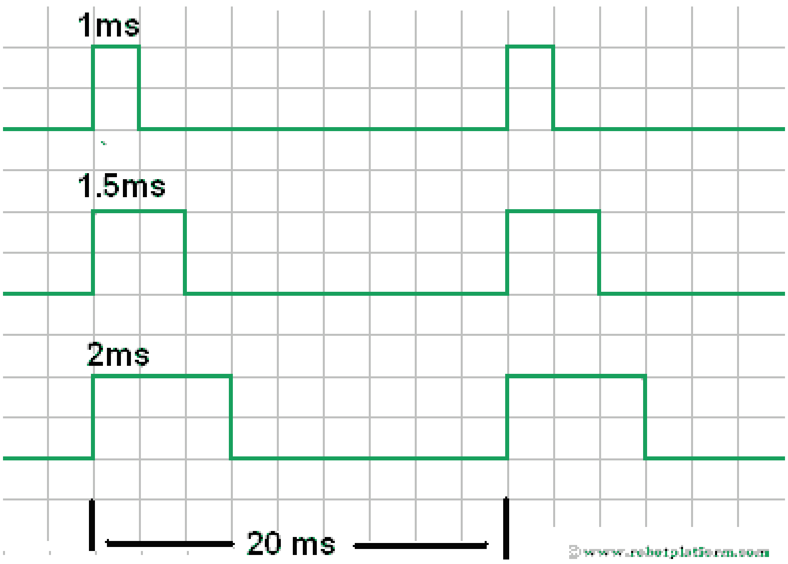
The reading of signals as well as the PID algorithms is explained in the following sections.

Section 3.2: PWM Signal operation guide

PWM refers to Pulse Width Modulation. It is a very common method of transmitting data, such as the position of a joystick or the speed of a motor. In our case, we are using PWM to read to radio signals from the receiver and to control the rotational speed of the thrusters.

PWM is an electronic signalling method used over electric wires. Every 20 milliseconds, the pulse is set to high. The pulse remains high for a minimum of 1 millisecond and a maximum of 2 milliseconds before switching to low. The time difference between the start of one pulse and the start of the next pulse is usually 20 milliseconds.

Attached below is a diagram of a sample PWM wave.



For the R6B receiver, a 1 ms pulse corresponds to a joystick in the full backward position while a 2 ms pulse corresponds to a joystick in the full forward position.

In the Arduino, the `pulseIn(PIN, HIGH)` is used to capture the time of the pulse. Using this, the position of the joystick is determined.

This is then used to generate a new PWM pulse to run the motor via the Electronic Speed Controller (ESC).

For a bi-directional ESC, a 1ms pulse indicates full-speed reverse. 1.5ms indicates the neutral point i.e. no speed. 2ms indicates full-speed forward. Depending on which way the robot needs to move, a pulse of different widths is sent to the various ESCs.

Section 3.3: Control algorithms

The current robot uses 3 important control algorithms: Deadzone, Linear speed, and Gyro PID.

Dead zone: A Dead Zone is a point where there is no movement. For example, the bidirectional ESCs have a dead zone at 1.5ms PWM. Similarly, the centre 20% of the transmitter's joystick is considered as a dead zone. Due to this, if the joystick is within 20% of the centre, then there will be no movement.

Linear speed: The Linear Speed algorithm ensures that the force output of the robot is proportional to how much the joystick is pushed. For example, when the joystick is pushed forward slightly, the robot will output a small force; when the joystick is pushed fully, the robot will output the full force.

Gyro PID: The Gyroscopic PID algorithm is used to keep the robot in a horizontal position at all times. It is perhaps the most complicated of the three control algorithms. It consists of two important parts: Correction and Prediction.

1. Correction: The Gyroscope measures the tilt of the robot in degrees. This value is then multiplied by a correction factor. Then it is sent to the vertical motors, which correct the tilt. This is referred to as the "Proportion" in PID.
2. Prediction: To predict the future tilt, we store the current tilt value in memory. Then we compare it to the old tilt value. Using this, we can get a prediction of what the future tilt value will be. This is referred to as the "Derivative" in PID.

More control algorithms will be added in the future; for example, a laser depth PID to control the diving depth.

Chapter 4: Usage and Calibration

Section 4.1: Binding the transmitter and receiver

During first-time use, it is possible that the transmitter is not linked to the receiver. In that case, one must manually bind them.

Step 1: Power off the robot. Ensure that the main switch is off and that no current is flowing. Power off the FS-CT6B transmitter.

Step 2: Remove the electronic components from the acrylic tube. Ensure that all parts are dry during removal and that none of the wires break.

Step 3: Locate the R6B receiver, which looks like this

Step 4: Using a female-female jumper cable, connect the top left pin (leftmost pin on `BAT` row) to any of the rightmost pins.

Step 5: DO NOT SWITCH ON THE TRANSMITTER. Power on the Arduino. A red light on the R6B should start flashing.

Step 6: On the transmitter, hold down the `BIND` button. With the button continuously held down, power on the transmitter.

Step 7: Upon successful binding of the receiver, the flashing red light should now be continuously on. The connection is successful; the robot and the transmitter can now be safely turned off.

Section 4.2: Calibrating the motors

Due to minor uncertainties in the ESCs, it is possible that the thrusters are not completely off when in the neutral position. To fix this, the motors must be calibrated.

Step 1: Download the latest release of the RoboMairne programs from
https://github.com/AdityaIyer2k7/robomarine-release/releases/latest

Step 2: Ensure that both the robot and the transmitter are powered.

Step 3: Carefully remove the electronic components from the acrylic tube. Remember, the robot is not powered off. Ensure that all parts are dry during removal and that none of the wires break.

Step 4: Using the Arduino connection cable, join the Arduino UNO to a PC device with the Arduino IDE installed.

Step 5: On the PC device, open the folder named `Calibration`. Inside, open the file `Calibration.ino` using the Arduino IDE.

Step 6: Inside the IDE, locate the following line:

```
/* Edit this line to choose the motor*/
#define MOTOR_TO_CALIBRATE MOTOR_FRONT_LEFT
```

Change `MOTOR_FRONT_LEFT` to whichever motor you want to calibrate. The options are:

- `MOTOR_FRONT_LEFT`
- `MOTOR_FRONT_RIGHT`
- `MOTOR_BACK_LEFT`
- `MOTOR_BACK_RIGHT`

Step 7: Ensure that the Arduino is connected to the device. Press the Upload button at the top left of the screen.

Step 8: The motor entered in Step 6 should start moving. Use the left knob on the CT6B to control its speed. Adjust it so that the motor is not spinning. Keep the knob in this position.

Step 8: On the PC device, open the Serial Plotter located at the top right of the screen. You should see a graph which is constantly updating with time. Identify the value displayed on the graph. It should usually be between 85 and 95.

Step 9: Exit the `Calibration` folder. Enter the folder `RoboMarineProgram`. Inside, open the `RoboMarineProgram.ino` program. Locate the following line:

```
#define MOTOR_YOUR_MOTOR_DZ 90
```

Where `YOUR_MOTOR` is the motor used in Step 6. Replace the value 90 with the value obtained in Step 8.

Step 10: After making the above change, press the upload button once more. In the neutral position, all motors should be off. This indicates a successful calibration.

Note that in the future, the ability to calibrate motors will be built into the robot itself. A PC device would not be required to run the calibration program.

Chapter 5: Testing procedures

Section 5.1: Preparing for the first dive

Checks to be performed before submersion:

1. Check whether the transmitter is bound to the receiver (Make sure to turn on the transmitter before the receiver).
2. Ensure that the ROV is sealed. Dip the robot in shallow water for a minute and check if there is any seepage inside the waterproof hub.
3. Arrows drawn on the metal rods point in the forward direction. For the initial test, place the robot so that it faces forward according to you. Then use the CT6B transmitter to test that each motor is running correctly.
4. Submerse the ROV to a depth of 1 metre (3.3 feet). Ensure that the robot still responds to signals from the CT6B.

Troubleshooting:

- If the R6B receiver is not linked to the CT6B transmitter, refer to Section 4.1 for details on binding
- If a seepage of water is found in Step 2, follow the following steps:
    1. Identify the point of leakage. If the water trail is not visible, you may use a dye to trace it.
    2. Drain the water from the robot. Let the electronic components dry.
    3. Apply a waterproof sealant and wait for it to dry. Please wait for it to dry completely to prevent it from dissolving in the water.
- If the battery is discharged/damaged, perform the following steps:
    1. Power off the robot. Ensure that the main switch is off and that no current is flowing. Power off the FS-CT6B transmitter.
    2. Remove the electronic components from the acrylic tube. Ensure that all parts are dry during removal and that none of the wires break.
    3. Identify the 3S Lithium Polymer battery. Disconnect the battery by removing the orange connector clip from the robot.
    4. Charge the battery using a certified 3S LiPo charger.

5. In case the battery is inflated, leaking, or damaged in some other way, carefully dispose of it as per the electronic waste disposal guidelines. Contact the RoboMarine team for help regarding battery replacement.
6. Once a charged/new battery is obtained, attach it to the robot using the orange connector clip.
7. Place the electronics back inside the Acrylic Tube Hub and Power On the robot.

Section 5.2: Challenges and Limitations

- Balance: Small inaccuracies in the parts could lead to an uneven weight distribution. At high speeds, this error could compound to cause the robot to flip over.

- Circuitry inside Hub: The robot uses a 3S LiPo battery, which is considerably heavy. If the battery moves inside the Acrylic Tube Hub, then there would be a change in the centre of mass and hence the balance. To account for this, the robot uses a 3D-printed shelf so that all of the electronic components are secured.

- Waterproof seals: Even though the wires of the ROV were sealed at the hub, it was discovered that water would seep into the robot through the gap between the three wires (inside the 3-wire rubber cable). The rubber cable was hence sealed, resolving this issue.

- Vertical motor orientation: During development, the team received a clockwise motor instead of anticlockwise. This incorrect motor was used for the vertical axis, where there is a lower risk. However, the unequal torque caused by this caused the robot to drift. This problem will not occur in production, since the incorrect CW motor would be replaced by a CCW motor. However, in case some drift does occur, the robot can be rotated using Channel 2 on the CT6B.

- Weight: The first prototype of the robot was not heavy enough to counter the buoyant force of water. Due to this, the vertical thrusters could not submerge the robot. The robot has been made heavier in the new version, so this is no longer an issue.

Chapter 6: Future Upgrades

Section 6.1: Upgrading Sensor Suite

Current sensors include an MS5837 Pressure sensor and an MPU6050 Gyroscopic sensor. The ROV will also include a camera to capture underwater footage.

In the future, the ROV will be able to communicate with many more sensors through an open I2C port and a modular design. The telemetry data will be recorded on an in-built Raspberry Pi 3B+ and transmitted to remote devices as well. Through this feature, RoboMarine will become a versatile Underwater Data Collection and Operation robot.

Section 6.2: Developing Autonomous Navigation

Developing autonomous navigation systems is a complex and multidisciplinary task. There are various approaches to building such a system, all of which utilise highly specialised spatial mapping and decision-making algorithms.

Autonomous Navigation has a variety of uses in the following scopes:

- Safety: Autonomous navigation reduces the need for human intervention. This decreases the risks associated with manned submarine missions. This is particularly important for deep-sea exploration and military applications.

- Efficiency: Autonomous submarines can operate continuously and efficiently for extended periods without human fatigue, making them suitable for tasks such as long-term environmental monitoring and data collection.

- Cost Reduction: Eliminating the need for a crew reduces operating costs significantly, making autonomous submarines a cost-effective choice for many missions.

Section 6.3: Using Virtual/Augmented Control

Augmented and Virtual Reality (AR/VR), generalised as Extended Reality (XR), offers a great solution for developing control systems for Underwater Remotely Operated Vehicles (ROVs).

The benefits of using XR Control include the following:

- Enhanced Situational Awareness: XR systems provide operators with a 3D visualisation of the underwater environment, allowing for a better understanding of surroundings, obstacles, and mission objectives.

- Immersive Experience: Operators can experience a more immersive and intuitive control interface, making it easier to manipulate the ROV's movements and interact with objects in the underwater environment.

- Precise Control: XR systems often offer more precise control over ROV movements, which can be crucial for delicate tasks such as inspections, sample collection, or manipulation of underwater equipment.

- Reduced Operation Fatigue: The immersive nature of XR can reduce operator fatigue compared to traditional control interfaces, as it allows operators to be more engaged and focused during long missions.

- Remote Operation: XR control enables operators to control ROVs from remote locations, which can be useful for missions in distant or hazardous underwater environments.

Chapter 7: Conclusion

Section 7.1: Recap of achievements

Maiden Test Dive of Underwater Submarine Robot in a Swimming Pool

The development of underwater robotics has opened up new possibilities in exploring and understanding the depths of our oceans. One significant milestone in this journey is the first test dive of an underwater submarine robot in a swimming pool. This write-up will provide an overview of this groundbreaking event, showcasing the significance of this technological achievement.

Before delving into the test dive, let's take a moment to understand the submarine robot itself. This sophisticated piece of engineering is equipped with state-of-the-art technology, including advanced sensors, cameras, and propellers, which enable it to navigate underwater environments with precision. It is designed for various purposes, including underwater exploration, research, and even underwater maintenance tasks.

Test Objectives:

The primary objectives of the first test dive in the swimming pool are as follows:

1. Assessing Mobility: We aimed to evaluate the robot's ability to move in water, control its depth, and navigate around obstacles.

2. Testing Communication: Ensuring that the robot's communication systems work flawlessly both underwater and on the surface.

3. Stability and Balance: Confirming that the robot maintains stability and balance in various water conditions.

4. Sensor Performance: Checking the accuracy and reliability of the robot's sensors, which are crucial for data collection and navigation.

The first test dive of the underwater submarine robot in a swimming pool represents a significant milestone in the field of underwater robotics. While it may seem like a small step, it marks the beginning of a journey that could lead to groundbreaking discoveries in ocean exploration, environmental monitoring, and underwater maintenance tasks. As we robotics enthusiasts continue to refine the robot's capabilities, we can look forward to a future where these machines play a vital role in understanding and preserving our planet's underwater ecosystems.

Section 7.2: Future Endeavours

- Adding Accessories such as grippers and open I2C and SPI ports for external sensors.
- Developing wireless underwater signal transmission technology.
- Developing custom 3D designs for thrusters and propellers.
- Adding device support for peripherals.
- Switching from Arduino UNO to bigger boards such as the GIGA, and possibly to custom-designed PCBs.
- Improving the attached camera for better footage.

Section 7.3: Future Scope

- Aquatic exploration and research: Our goal is to have our robot assist in exploring underwater structures and terrain. This is very useful for researching marine life such as fishes and corals, and also for uses like creating 3D maps of rivers and lakes. To accomplish this, the robot has an in-built Camera and Infrared lights which allow it to see even in low-light conditions.

- Water body analysis: Our goal is for our robot to be able to evaluate different characteristics of water bodies, such as the salt content of seas or the flow speed of rivers. This can allow geological teams to get data about the water bodies quickly. The robot can also check the amount of plastic present in these water bodies and can help in cleaning them. For this task, the robot has a modular interface which allows the user to connect different modules to the robot, such as the water quality sensors and plastic collecting grippers.

- Construction and maintenance: Our goal is to have our robot assist in the construction and repair of underwater structures like bridges and dams. For example, a majority of internet cables travel underwater and need expensive machinery to repair. Using our robot, these cables can be repaired in an inexpensive way. To perform underwater construction, the robot can be programmed to connect with other robots in the same network and coordinate the construction task.

- Wreck inspection: Our goal is to have our ROV play a crucial role in wreck inspection, particularly in underwater environments such as oceans, lakes, and rivers. They are used to explore, assess, and document the condition of wrecks, which can be ships, aircraft, submarines, or other submerged structures.

- Offshore wind farm inspection and monitoring: Our goal is to have our ROV help green energy companies monitor offshore wind farms, cutting the costs for multiple boat journeys, in turn making electricity cheaper

- Marine Defence: Our goal is to have our ROV help in marine defence and perform tasks like Surveillance and Reconnaissance, Mine Detection and Clearance and Search and Recovery which will help in the safety of our nation.

Appendix

Appendix A: Bill of Materials

Sr no	Product Description	Rate	Qty	Amt
1	Acrylic Tube	1800/-	1	1800/-
2	Acrylic Tube Cap	750/-	2	1500/-
3	Aluminium Extrusion Beam 15x15mm square profile 420mm	310/-	12	3720/-
4	Aluminium Extrusion Beam 15x15mm square profile 275mm	205/-	4	820/-
5	Aluminium Extrusion Beam 15x15mm square profile 60mm	45/-	2	90/-
6	Arduino UNO	900/-	1	900/-
7	*Brushless DC underwater thruster CW propeller	4500/-	4	18000/-
8	*Brushless DC underwater thruster CCW propeller	4500/-	2	9000/-
9	*Bidirectional Electronic Speed Controller	2200/-	6	13200/-
10	FS-CT6B Transmitter	3800/-	1	3800/-
11	FS-R6B Receiver	700/-	1	700/-
12	MPU6050 Gyroscope sensor	200/-	1	200/-
13	MS5837 Pressure sensor	2345/-	1	2345/-
	Grand Total			56975/-

Appendix B: Arduino Code Samples

All codes currently in use are available at the GitHub repository
https://github.com/AdityaIyer2k7/robomarine-release/releases/latest

Programs currently in development, as well as design documents, are stored in a private repository and are not accessible to the public. This may change if RoboMarine chooses to become open-source someday.

Appendix C: Troubleshooting guide

Known problems and errors:

- CT6B Transmitter is not linked to the robot: Refer to the Binding procedure outlined in Section 4.1
- Motors are rotating even when the transmitter is neutral: Refer to the Motor Calibration process outlined in Section 4.2
- Water is leaking into the Acrylic Tube Hub: Refer to the Troubleshooting section in Section 5.1
- Battery is discharged or damaged: Refer to the Troubleshooting section in Section 5.1
- Beam or Acrylic Tube is damaged: Contact the RoboMarine Team regarding help replacing damaged parts

Words of Inspiration