

**JOMO KENYATTA UNIVERSITY**

**OF**

**AGRICULTURE AND TECHNOLOGY**

**BSc MARINE ENGINEERING**

**PROJECT REPORT**

**DESIGN, FABRICATION AND TESTING OF A LOCALLY REMOTE OPERATED VEHICLE (ROV) FOR SHIP HULL INSPECTION**

|  |  |  |
| --- | --- | --- |
| **PROJECT CODE:** | **MR-PRO-24-04** |  |
| **STUDENTS:** | **DENNIS SAMMY** | **ENM 241-0117/2019** |
| **PEGGYOLANDA NYAMBURA** | **ENM 241-0118/2019** |
| **SUPERVISOR:** | **DR. ELIJAH MUNYAO** |  |

This project report has been submitted in partial fulfillment of the requirements for the award of degree of Bachelor of Science Marine Engineering

**DECLARATION**

We declare that this project is our original work. To the best of our knowledge, we also affirm that this project has not been presented to this university or any other university for examination, or for any other purpose.

|  |  |  |
| --- | --- | --- |
| Name: | **DENNIS SAMMY** | **ENM 241-0117/2019** |
| Signature: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |  |
| --- | --- | --- |
| Name: | **PEGGYOLANDA NYAMBURA** | **ENM 241-0118/2019** |
| Signature: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**CERTIFICATION**

I certify that the above-mentioned students carried out the work detailed in this project under my supervision.

|  |  |  |
| --- | --- | --- |
| Supervisor’s Name: | **DR. ELIJAH MUNYAO** |  |
| Signature: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Abstract

Hull inspection is an essential process for any sea going vessel to ensure safety of navigation. Over the years, hull inspection has been achieved through dry docking as well as deploying inspectors to carry out underwater inspections. However, this has had limitations due to the inadequate human capabilities underwater as well as the associated safety risks. Remotely operated vehicles eliminate the risks that divers are subjected to during hull inspections and eliminates expenditure on the entire hull inspection process in dry docks. The use of an ROV as an underwater inspection tool provides precise and efficient underwater inspections at reduced risks, associated costs, and docking time periods. The current work involved the design, fabrication, and testing of a remotely operated vehicle. It is worth noting that most vessels have greater draughts more than 7 m. Consequently, the design is based on a draught average of 10m which is greater than the draughts of majority small marine vessels such as Aylah 1 and Falcao.

**Table of Contents**

[Abstract 4](#_Toc172100618)

[List of Figures 7](#_Toc172100619)

[List of Tables 8](#_Toc172100620)

[List of Abbreviation 9](#_Toc172100621)

[1. Introduction 10](#_Toc172100622)

[1.1 Background 10](#_Toc172100623)

[1.2 Problem statement 11](#_Toc172100624)

[1.3 Objectives 12](#_Toc172100625)

[Specific objectives 12](#_Toc172100626)

[1.4 Scope 12](#_Toc172100627)

[2. Literature review 13](#_Toc172100628)

[2.1 Work Class ROV 15](#_Toc172100629)

[2.2 Observation Class ROV 15](#_Toc172100630)

[2.3 Micro ROV 15](#_Toc172100631)

[2.4 Work Class ROV 16](#_Toc172100632)

[2.5 Special Use ROVs 16](#_Toc172100633)

[3.0 Methodology 18](#_Toc172100634)

[3.1 Conceptual Design 18](#_Toc172100635)

[3.1.1 Mechanical & Structural design 18](#_Toc172100636)

[3.1.2 Propulsion mechanism Design 22](#_Toc172100637)

[3.1.3 Electrical and control design 26](#_Toc172100638)

[3.2 Fabrication and assembly 32](#_Toc172100639)

[3.3 Testing 32](#_Toc172100640)

[4. Results and Discussion 33](#_Toc172100641)

[Task 1: Research and selection of potential materials for key components. 33](#_Toc172100642)

[Task 2: Analysis of proposed materials 35](#_Toc172100643)

[Stress and Deformation of Steel on Depth Variation 39](#_Toc172100644)

[Stress and Deformation of Aluminium on Depth Variation 40](#_Toc172100645)

[Stress and Deformation of PVC on depth variation 41](#_Toc172100646)

[Stress and Deformation of Acrylic on depth variation 42](#_Toc172100647)

[Task 3: Determining the weight and displacement of the ROV. 48](#_Toc172100648)

[Task 4. Calculating the buoyant force, Hydrodynamic drag force, power of the motor, and the thrust Force. 49](#_Toc172100649)

[5. Conclusion 53](#_Toc172100650)

[6. Budget 54](#_Toc172100651)

[7. Time plan 55](#_Toc172100652)

[References 56](#_Toc172100653)

# List of Figures

[Figure 1 Rebikoff-niggeler Foundation First ROV 13](#_lnxbz9)

[Figure 2 A tethered ROV explores the ocean depth 14](#_35nkun2)

[Figure 3 Proposed design of a 3D ROV model 20](#_3whwml4)

[Figure 4 Acrylic cylindrical compartment 21](#_2bn6wsx)

[Figure 5 Nylon plastic to seal compartment 21](#_qsh70q)

[Figure 6 Kort Nozzle 24](#_1pxezwc)

[Figure 7 Orientation of thruster rotation 25](#_49x2ik5)

[Figure 8 Arduino nano 26](#_147n2zr)

[Figure 9 Raspberry pi 3 26](#_3o7alnk)

[Figure 10 Pressure sensors 27](#_23ckvvd)

[Figure 11 PID Controller 28](#_ihv636)

[Figure 12 Electronic Architecture 28](#_32hioqz)

[Figure 13 A ethernet cable used for transmission of data 29](#_1hmsyys)

# List of Tables

[Table 1 Estimation of the required deadweight of the ROV. 30](#_41mghml)

[Table 2 Estimated Budget 33](#_2u6wntf)

[Table 3 Time Plan 34](#_3tbugp1)

# List of Abbreviation

|  |  |
| --- | --- |
| Remotely operated vehicle | ROV |
| Polyvinyl chloride | PVC |
| Electronic Speed Controller | ESC30A |

# 1. Introduction

## Background

In the shipping industry, ships are required to be inspected over a given period as safety measures [1]. Currently, there are significant challenges associated with conducting hull inspections. These include the high cost, a time-consuming process involved in inspection of hulls in dry docks. In cases where underwater hull inspections are adopted, the inspectors are exposed to a myriad of safety risks as well as the inability to access some areas of the submerged hull. The hull is exposed to underwater processes such as biofouling and corrosion, making its inspection vital and therefore it should be conducted regularly, which is not feasible with the currently available method. Both Biofouling and corrosion have a great negative impact on the performance, condition and lifespan of the ship. Biofouling has also been identified as a contributor to increased fuel consumption for sea going vessels, consequently contributing to increased emission ozone depleting gasses - the greenhouse gasses.

Currently, the existing hull inspection methods include dry docking and use of underwater hull inspectors [2]. In dry docking, a ship is unsubmerged for a comprehensive inspection to be conducted on the hull, the rudder, propeller, and stabilizers. Dry docking has proven to be expensive and time consuming considering the required labor, resources and planning. The docking facilities are also not readily available resulting to delays in inspection. Another alternative involves utilizing underwater inspectors to evaluate the status of the hull. Even though this reduces the amount of resources and cost expended in the inspection, it has proven to be a difficult and risky endeavor. It is life threatening as it exposes the inspectors to hydrostatic pressures, low visibilities, marine predators, rough and harsh sea conditions [3]. Due to these limitations, there is a need to develop alternative reliable and safe methods to conduct hull inspection. One of the feasible approaches is to use a Remotely operated vehicle (ROV).

A Remotely operated vehicle (ROV) is a maneuverable underwater machine that can be used to explore and inspect ocean depths operated by personnel at the water surface. ROVs are versatile tools that find applications in oil and gas companies, marine construction works, military and law enforcements applications. They are also used in underwater research as well as in shipping industry, bringing forth numerous benefits in these areas of application [4]. ROVs allow the accessibility of inaccessible underwater areas. They also guarantee improved safety and operate with low maintenance. They have been used to recover torpedoes, bombs, and mines from the 1960s by the navy as well as to carry out missions that are very dangerous to humans such as mine breaking and mineral harvesting [5]. ROVs have also been used by the leading coating provider, Hempel, to perform remote hull inspections [6]. It is based on the above discussion that this project implements a locally made ROV for hull inspections.

## Problem statement

Inspection of ship hulls and offshore structures is a major challenge in the maritime sector. Conventional hull inspection methods such as docking, and use of underwater inspectors have proven to be costly and risky. An alternative way to overcome these challenges is to create a device which will fulfill the underwater duties. Remotely Operated Vehicles (ROV’s) have become vital tools in industries that are related to underwater. Deployment of ROVs has been proven to reduce the inspection cost from the previously existing methods but still costly for the local people. The existing ROVs are relatively expensive, and this is a prohibitive barrier to the entry into application, preventing widespread adoption of ROV for personal, research, and conservation applications. The current work therefore seeks to address the challenge by coming up with a low cost locally available Remotely Operated Vehicle without compromising its functionality.

## Objectives

The main objective of the current work is to design, fabricate and test a local Remotely Operated Vehicle (ROV) for hull inspection.

### Specific objectives

1. To select, analyze and propose materials for the ROV.
2. To design and test the structural integrity of the ROV frame.
3. To design, fabricate and test the control system of a Remotely operated vehicle (ROV).
4. To achieve maneuverable controls on the ROV’s main axes of motion, X, Y, Z and around an obstacle underwater.

## Scope

The scope of this project is to design, fabricate and test the control system of the Remotely Operated Vehicle. The ROV should maintain structural integrity as well as display maneuverable controls along the main axes of motion and around an underwater obstacle.

# 2. Literature review

Over time, humans have creatively built tools to improve lives and the ROV is no different. The human desire to reach places underwater that are too dangerous or too deep for a diver to venture into has seen the development of ROVs. ROVs are unoccupied machines designed for specific underwater missions. Typical ROVs are controlled from a vessel, land, or a floating platform.  The first tethered ROV referred to as POODLE was developed by DimitriRebikoff in 1953-1954. The United States navy later advanced the ROV technology to recover bombs, mines, and torpedoes in the 1960s that had been lost during tests. ROVs gained popularity when the Cable Controlled Underwater Recovery Vehicle systems were used by the US Navy to recover an atomic bomb lost off Palomares Spain and then saved pilots of a sunken submersible off Cork in Ireland with only minutes of air remaining [7].

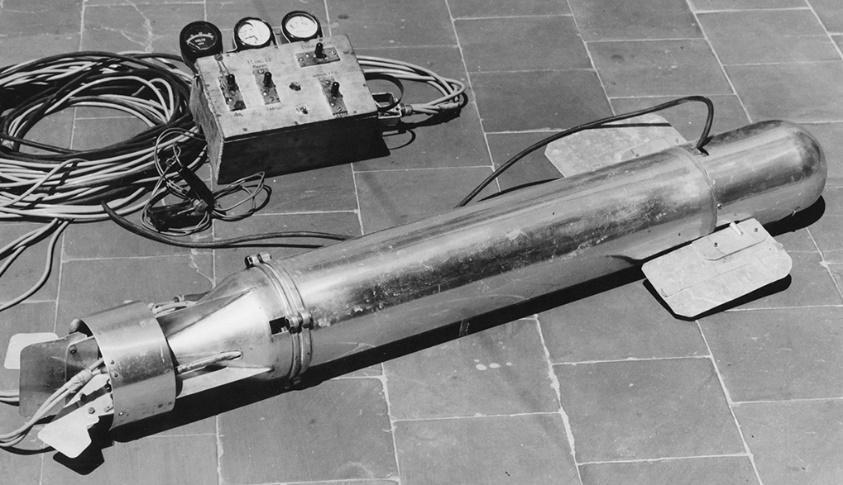


Figure 1 Rebikoff-niggeler Foundation First ROV

The ROVs were also used on rescue missions that exceeded the reach of human divers such as mine hunting and mine breaking. Since the 1980s, the offshore oil industry has successfully used ROVs to assist in offshore development, especially as waters became too deep for human divers. ROVs have become famous for discovering historic shipwrecks, including the famous RMS Titanic in 1985 and the Bismarck [8]. The first ROVs were initially referred to as Cable-Controlled Underwater Recovery Vehicles (CURV) since their primary function was to lead search and rescue operations in deep and dangerous waters.

ROVs consist of video cameras that transmit real time surveillance to scientists aboard vessels and this should be with low latency, sonar systems and a buoyancy pack that allow the vehicle to remain light and easy to maneuver when in the water. Thrusters are used to maneuver the vehicle, lights to provide illumination for the camera underwater, a frame that provides a structure to attach the ROV components, and a tether which carries electrical power and signals to the surface. ROVs use external sensors that are mounted on the vehicle to measure properties or parameters such as temperatures and depth. ROVs may be built with end effector such manipulators for collecting biological and geological samples.

Some ROVs are built with two bodies such as the NOAA Ocean Exploration’ vehicles [9]. The advantage of a two-body system is that the hovering ROV acts as an extra light source and camera giving pilots, scientists, and viewers an expanded view of the ocean. However, this has resulted in difficulties in control and transportation of single body counterparts. ROVs range in size, likewise the prices also vary a great vast too depending on the size and application of the ROV. They normally start at thousands of dollars and can go up to millions of dollars. The figure below shows an ROV using LED lights to explore the ocean floor.

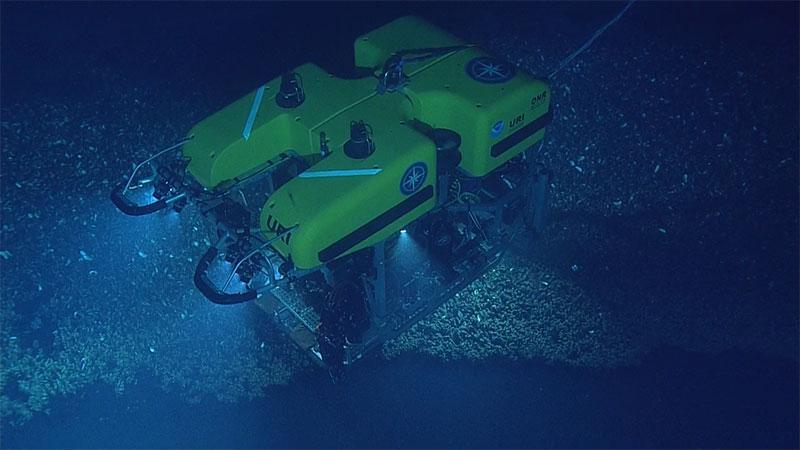
****

Figure 2 A tethered ROV explores the ocean depth.

Among the prominent ROVs is ROV Ventana. ROV Ventana was constructed by International Submarine Engineering in 1987 and launched by MBARI’s research vessel and has been in operation since 1988. It runs on hydraulic power, exploring waters at a depth of 1800 meters [10].

Underwater ROVs can be classified into various classes based on their weight, power, abilities and sizes. The common classes include the work class ROV, light work class, observation class ROV, and the micro ROV.

### 2.1 Work Class ROV

This class of ROV is used for ocean floor exploration and inspections at depths that divers are often unable to reach. Oil and gas exploration and production activities in the marine environment are increasing. In addition, the amount of global hydrocarbon reserves removed from below the seafloor is set to increase, with exploration heading into deeper, more remote waters, many of which are yet to be fully explored [11]. Work Class ROVs act as a safe alternative to divers and are often used in offshore energy projects and deep archaeological explorations.

### 2.2 Observation Class ROV

This class of ROV is small and is used to explore lakes, rivers, and coastal waters. They are often used to test water safety prior to a diver entering the water during missions and conduction inspections. They are equipped with sonar and custom sensors that make them versatile underwater vehicles.

### 2.3 Micro ROV

The ROVs in this class are the smallest and are often used to inspect hard to reach areas at shallow depths such as pipe systems and submerged infrastructure. They explore miniscule cavities and pipeline cracks which are physically impossible for divers to achieve.

### 2.4 Work Class ROV

This class of ROV is used for ocean floor exploration and inspections at depths that divers are often unable to reach. They act as a safe alternative to divers and are often used in offshore energy projects and deep archaeological investigations.

### 2.5 Special Use ROVs

They describe tethered underwater vehicles designed for specific purposes. An example of a special-use vehicle is a cable burial ROV system designed to plow the sea floor to bury telecommunications cables.

ROVs have great benefits that enable them to be a prospective candidate in providing a solution to underwater exploration, inspections, and adventures. They have been able to have quick deployment due to the compact design and easy-to-use inspection technology. This is highly beneficial in emergency situations and in areas that are narrow or difficult to access. ROVs are robust and rugged in design thus can withstand harsh water environments and require minimal maintenance. Depending on sea conditions and the type of operations, divers can only remain underwater for a limited amount of time; 30 min to an hour at a time. ROVs have overcome the challenge as they are designed to operate for long hours and in difficult conditions. One of the most remarkable benefits is the photo and video recording capabilities. ROVs can provide high resolution footage even in dark and murky waters. ROVs can be deployed to record and later be reviewed for documentation and results. In Addition, ROVs provide safe alternatives for exploration and cost effectiveness comparable to traditional methods [12].

Several industries have been able to benefit from the use of underwater ROVs technology. These industries include Military, Aquaculture, Municipal Infrastructure, Commercial and Salvage Diving, Ocean Science, Oil and energy, Shipping, Underwater Discovery.

In the current work, a ROV will be designed and fabricated from locally available materials for underwater hull inspection. The design will be based on a shallow draught landing craft with a draught of 0.75 m; therefore, a draught of 1 m is adopted. The ROV will be tested for maneuverable controls in the main directional axes as well as around an underwater obstacle.

# 3.0 Methodology

The current project is carried out systematically in three main steps: conceptual design, fabrication, and testing.

## 3.1 Conceptual Design

Conceptual design comprises the design of structural, electrical and control systems. The design aims to reduce the cost of the ROV by considering locally available materials and technologies without compromising the functionality of the ROV.

### Structural design

This design is an essential key part of the project as it helps in coming up with a sturdy and robust structure for the ROV. The design entailed:

* Determining the main dimensions; length, width, height.
* Determining weight of the ROV.

This provides the basis to create a 3D model that is essential in the successive sections of the design. A key part of the structural design is the selection of materials that are used to form the structure to withstand the high hydrostatic and hydrodynamic pressures that the ROV will be subjected to during its operation.

#### **Design constraints**

It is worth highlighting the major constraints that affect the design process.

1. Environmental constraints

The vehicle is to operate in sea waters, thus the parameters that will be experienced are maximum depth, density and salinity.

1. Density and salinity  
   The density of water is affected by temperature and salinity. Water is at its highest density at 4 degrees Celsius and the density decreases with an increase in saline concentration.
2. Working depth  
   The working depth defines the mechanical strength of all structural components. The current work suggests an operational depth of 10 m. This depth establishes a secure margin for the collapse resistance of the structural elements. Thus, the structural calculations are done at the design depth to guarantee the integrity of the ROV.

**Hydrostatic and hydrodynamic considerations**

A list of hydrostatic and hydrodynamic considerations is addressed by Allmendinger [13]. Such considerations include loads of interest, static equilibrium conditions, static and dynamic stability conditions and drag and thrust considerations.

* Load of interest

The loads that affect the ROV include weights, displacement force due to change in pressure, and drag and lift force as a result from the motion of the ROV in water,

* Static equilibrium

The buoyancy of the ROV is the difference between the displacement and its weight. The buoyancy is positive when the displacement is greater than weight and it is negative when the displacement is less than the weight. It is neutral when the displacement and the weight are equal. It is desired that the buoyancy be positive or neutral. This is achieved when the difference between the displacement and vehicle's weight be small and positive.

* Static and dynamic stability conditions

Static stability occurs when the heeling moment increases very slowly from zero to a maximum value while dynamic stability is when the heeling moment changes rapidly. Static stability is the most important measure of heeling behavior of a ship. A vessel submerged in water will always seek to assume a position in which all forces and moments acting upon it cancel each other out, which means that it is in equilibrium when its weight and buoyancy are equal but of opposite directions [14].

* Drag and thrust.

The thrust power must be able to overcome drag at an operational speed, and during acceleration intervals.

**Design Considerations and requirements**

The main parts of the ROV include the frame, the sealing compartment, thrusters, camera, lighting equipment, and buoyancy mechanism. The frame plays an important role of support and protection of the equipment. It is thus necessary the following factors should be considered in the design of the ROV:

• Withstand pressure at the designed depth.

• Cost effective.

• Lightweight to ensure a neutral or negative buoyancy

• Water corrosion resistance.

• Material Machinability.

• Robust and Sturdy.

**Pressure Calculation**

There have been previous attempts to build the ROVs as documented in [15]. With the desired operational depth of 10m, pressure can be calculated to determine whether the selected structural material for the frame is suitable.

|  |  |
| --- | --- |
|  | **1** |

Where P is the pressure, *p* is the density of water in Kg/m3, *g* is the gravity and *h* are the depth in meters. Taking *p* as density of sea water to be 1025 kg/m3.

Pressure is calculated using equation 1. Where P is hydrostatic pressure in MPa, h is 10m, g is 9.81m/s2 and density of seawater is 1025 kg/m3

|  |
| --- |
|  |

**Selection of materials**

The frame of the ROV, the sealing compartments and the end caps are the three key components of focus in material selection.

* **The Frame**

The frame is the structural architecture that will hold the components that will drive and execute the functions of the ROV. The materials proposed include aluminium, Stainless Steel, unplasticized PVC (Polyvinyl Chloride), and Acrylic- Plexiglass [reference]. The general properties of the material are shown in table 2 below.

**Table 2 pros and cons of proposed material**

|  |  |  |
| --- | --- | --- |
| Material | Pros | Cons |
| PVC | Light in weight  Excellent corrosion resistance  Cost effective.  Easy to obtain.  Durable | Susceptible to UV degradation  Lower strength in comparison to metals |
| Aluminium | Light in weight  Desirable weight-strength ratio  East to machine and fabricate.  Good corrosion resistance | Requires proper anodizing |
| Acrylic Plexiglass | Excellent strength -to weight ratio  Corrosion resistant  Ability to be moulded into complex shapes | Expensive  Complex manufacturing processes |
| Stainless steel | High strength and durable  Good corrosion resistance | Heavy  Expensive |

* **The Sealing compartment**

The sealing compartment is used to protect and house the electronic components of the ROV. The material used to make the sealing compartment of electronic components in Remotely Operated Vehicles (ROVs) is critical for ensuring the integrity and functionality of the electronics in underwater environments. The calculated pressure calculated for a 10m depth, the proposed material to be used for the body of the compartment is acrylic plexiglass that have a tensile strength of 55.16 MPa. Acrylic plexiglass can withstand the pressure at the designed depth. A clear acrylic tube is used with a view to observe all the different components mounted and to check the assembly.

* **The End Caps**

End caps are used to enclose the openings of the cylindrical sealing compartments. They are to be machined using Teflon, which is commonly used on bearings and low-friction bushings due to its: low friction, thermal resistance, and corrosion resistance. The design consists of 4 O-rings on the end caps as well as a wide flange nitrile gasket on the face of the dome and the fixed dome connecting the aluminium mount. A nylon ring is machined so that it presses the transparent dome against the front cap once sealed with silicone so that no water enters.

|  |  |
| --- | --- |
| **Figure 4 Acrylic cylindrical compartment** | **Figure 5 Nylon plastic to seal compartment** |

**Analysis of proposed materials**

Selection of the best material for the frame entails subjecting the proposed materials to a stress and deformation analysis test under load. The modelled design is subjected through a series of pressures at different depths of 10m, 50m, and 100m and the results are reviewed. Fusion360 is used to analyze the structure for stress and deformation. The mechanical properties, thickness, and allowable values of each material are then summarized.

### 3.1.2 Propulsion mechanism Design

The propulsion system allows the ROV to utilize underwater thrusters in a specific configuration to grant an operational range of motion converting input energy into useful thrust. The propulsion system consists of motors and propellers - a combination called a thruster. The size of the propeller is described by a pitch and diameter number. Rake is the degree that the blades of a propeller slant forward or backwards in relation to the hub which is the centre section of the propeller. The aim of the design of a ROV propulsion system is to have high thrust to physical size and power input ratios.

A propeller is selected with consideration for performance, speed, weight and the overall size of the ROV. Ducted propellers move more liquid, and they increase the efficiency depending on the flow rate. The selected propeller features two blades, a small diameter, a high rake angle and nylon polymer construction. Two bladed propellers are chosen because of the size of the ROV and the overall cost of the propeller. The smaller diameter propeller helps in avoiding collision with any of the frame components and to provide maximum thrust capabilities from the thrusters selected. A high rake angle allows for faster acceleration at lower rotations per minute thus allowing the ROV to respond rapidly to changes and reposition. Due to weight and cost restrictions, nylon plastic propellers are selected to ensure the best performance to cost ratio.

**Thrusters**

As the propeller rotates in ROV, high-pressure and low-pressure areas are formed between the blades. The thrust needed for the movement of the ROV is formed as the fluid moves from the high-pressure area to the low-pressure area. The design of the thrusters ought to allow for components to be replaced in future in case of a component upgrade or a part failure. The individual thrusters are to have identical propellers to allow for each propeller to have uniform performance during operation. This allows the operator to have more uniform control of the ROV despite external water conditions. As the propeller moves, there are losses at the tip of the propeller blades. To minimize the losses, a duct is designed around the propeller. The duct is designed with an aerodynamic structure to increase the efficiency of the design.

Brushless motor thrusters are chosen for the ROV since they are efficient and durable. The diameter of the propeller is chosen to be bigger than the motor diameter for enhanced thrust efficiency, reduced cavitation, lower power consumption, improved control and stability, optimized flow characteristics and hydrodynamic performance. The pitch of the blades depends on the diameter and the rotational speed of the motor in RPMs. The width of the blade determines the amount of water it pushes, thus light blades are used for higher speed applications. (Reference). They are designed to produce forward and reverse thrust. Each thruster is controlled using an electronic speed controller (ESC).

**Thruster configuration**

The ROV is designed with five thrusters. The thrusters mounted involve two thrusters moving in the horizontal motion along with three vertical thrusters to dive. Two thrusters are placed on the back of the ROV equidistant from the centre of momentum in the X-direction to minimize rotation. They provide translational motion in the X-direction. The remaining three thrusters are placed on the top of the ROV which provides the motion in the Z-direction. They are placed along the X-axis for the ROV to obtain pitch rotation by only firing the front thruster for pitch down and firing only the back thruster for pitch up. This configuration allows to actively control the system in six degrees of freedom. (Reference) The basis of this ROV system consists of 5 Blue Robotics T100 thrusters, which have the following technical characteristics: (Reference)

|  |  |
| --- | --- |
| Property | Description |
| Thrust | 14.7 N forward and 11.77N reverse |
| Dimensions | 72 by 92 mm |
| Mass | 0.075kg |
| Electric Power | Up to 150W when powered from 12V DC |
| Supply voltage | 12-24V |

**Analysis of upthrust, buoyancy, drag force, power of the motor and the thrust Force.**

**Calculation of Upthrust Force:**

Upthrust is the force exerted by a fluid on an object in an upward direction. It is equivalent to the weight of the liquid displaced by the submerged part of the body such that m is the mass of the body and g is the acceleration due to gravity.

|  |  |
| --- | --- |
|  |  |

**Calculation of buoyant force.**

The buoyant fore is the upward force exerted by the water on the ROV, which counteracts its weight. It is calculated using the Archimedes principle

|  |  |
| --- | --- |
|  | **2** |

where p is the pressure, g is the gravity and V is the volume of the ROV.

The ballast weights and syntactic foam are either added or subtracted to fine tune the ROV’s buoyancy and achieve a neutral buoyancy.

**Calculation of Drag force.**

The thrust required to achieve a speed of 2m/s depends on the hydrodynamic drag force acting on the ROV. The drag force on an ROV moving through water can be calculated using the drag equation (Caption)

|  |  |
| --- | --- |
|  | **13** |

Where:

ρ is the density of sea water= 1025kg/m3 and gravitational acceleration = 9.8m/s2

v is the desired velocity (2 m/s)

Cd​ is the drag coefficient

A is the characteristic cross-sectional area of the ROV

The Reynolds number is used to predict the flow patterns in fluid dynamics. It is a relation between density ρ, characteristic length L, velocity, U and fluid viscosity μ. The Reynolds number for the ROV in the water with ρ = 1025 kg/m³ and μ = 1.31 Cp =1.31 x 10³ kg/ms and moving at 2 m/s is calculated using equation (reference) and the actual value calculated in equation (reference)

The total drag of the ROV system is the vehicle drag and the tether drag. The coefficient of drag for tether cables used in Remotely Operated Vehicle (ROV) applications vary depending on the cable's surface characteristics, diameter, and flow conditions. The tether drag coefficient in a turbulent flow regime is approximated to be 1.2(Reference).

|  |  |
| --- | --- |
|  | **16** |

The total hydrodynamic drag force on the ROV system is thus the summation of the tether drag and the vehicle drag, and it is given by equation (Caption)

|  |  |
| --- | --- |
|  | **18** |

**Power of motor analysis**

To calculate the power of a motor needed to achieve a speed of 2m/s and overcome a drag force of the fundamental relationship between power, force and velocity is applied in equation:

|  |  |
| --- | --- |
|  | **19** |

The thrust force, FT, to achieve a desired speed is given by:

|  |  |
| --- | --- |
|  | **21** |

Where P is Power in watts, F is force in newtons, V is velocity in m/s,

|  |  |
| --- | --- |
|  | **22** |

### 3.1.3 Electrical and control design

The electronic circuitry is responsible for signalling to and from the control side and data acquisition during the inspection. The electronic equipment is securely sealed inside the electronic enclosure.

**Electrical devices**

 The electrical devices include microprocessors, microcontrollers, motors, relays, batteries, jumper cables, ethernet cable, camera, operator controller and sensors.

**Power system**

The power system describes how the ROV components are powered. This project is proposing batteries as the power source. The batteries are meant to be reusable hence rechargeable li-on batteries are used. Due to the presence of motors that are used to power the thrusters more power is drawn from the batteries. A proposed solution is to allocate a second power source coupled with relays and sensors to provide power to the motors when needed.

**Control system**

The control system manages the input from the operator controller at the surface and converts it into control motion of the ROV underwater. Measurable parameters such as voltage, battery life, depth, and position of the ROV are recorded by sensors and displayed to the operator.

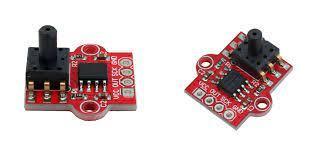
**Microprocessor and microcontroller**

The microprocessor and the microcontroller are responsible for executing parallel tasks such as motor control, measurement and recording of different sensors of the system, battery voltage monitoring for energy consumption and communication management with remote control. The Raspberry PI 3 is the selected microprocessor, and Arduino nano will be used as microcontroller. Figures (reference) show the microprocessor and the microcontroller.

|  |  |
| --- | --- |
| **Figure 8 Arduino nano** | **Figure 9 Raspberry pi 3** |

**Sensors**

Pressure and depth information is essential while the ROV is in operation. Pressure sensors, integrated with the code, can determine the information on depth. Figure illustrates a pressure sensor.



**Figure 10 Pressure sensors**

Electronic speed controllers (ESC) regulate the speed of motors. They include a pair of relays for each motor. The sensors required for the inspection are added as payload and integrated with the circuitry. Ethernet cable used as the tether for transmission of control signals, the camera feed, and the sensor data from the ROV**.**

Various sensors specific to the application are equipped to carry out detailed underwater monitoring. These sensors also serve as a health monitor to prevent its navigation beyond the operating range. The ROV is equipped with a temperature sensor with a usable temperature to monitor the temperature underwater for any alarming temperature variations. The pressure sensor was used to provide real-time pressure data that helps in determining the depth of the ROV. It also serves as an external aiding sensor for position tracking by providing depth. The data from these sensors are serially communicated to the microcontroller which is connected to the processor. They are powered by a 5V converter. These sensors are placed along the inner walls of the ROV frame and wired into the enclosure via a slotted acrylic flange.

The IMU contains a gyroscope and accelerometer for providing compensated incremental angle and velocity data for navigation and angular rates, linear accelerations, and magnetic fields for control through a digital serial interface bus. The IMU measures angular rates(rad/sec), linear acceleration(m/sec2), and magnetic fields(milli-gauss) in a body mounted strap down configuration along the three axes which are fed to a sensor fusion algorithm to obtain the attitude and heading of the ROV with respect to the earth’s frame. The navigation data involving the incremental angles and velocities along the three axes are primarily used for positioning the ROV while the control data is used in the determination of the orientation in Euler angles (roll, pitch, and yaw) and control of the vehicle. Since stand-alone gyroscope has no frame of reference in contrast to accelerometer with earth’s acceleration due to gravity as the reference and magnetometer with the earth’s magnetic north as a frame of reference, all the three sensors are fused to obtain the complete

orientation of the sensor with respect to a navigation frame. The IMU was powered with a 5V converter. It was mounted vertically on the flat tray at the centre of gravity of the vehicle such that the x-axis of the IMU coincides with the direction of the vehicle’s heading. Further calibration of the IMU after placing it at the COG was done to reduce the bias and obtain precise values.

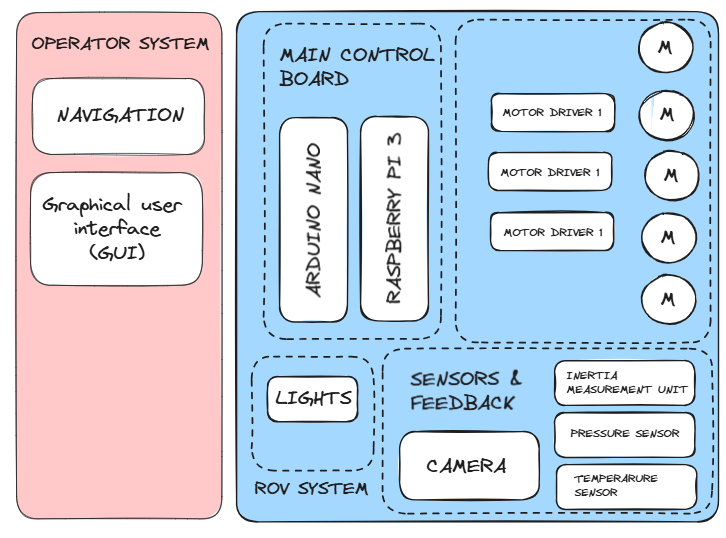
**Electronics architecture**

**ELECTRONIC SYSTEM ARCHITECTURE**

Figure (Caption) shows the block diagram of the proposed hardware that consists of two parts: the ROV vehicle system and the operator system. These two subsystems are communicated through an Ethernet Cable so that the user can control the ROV from a personal computer, laptop. The ROV hardware involves a Raspberry Pi 3 that is responsible for executing parallel tasks, such as:

* Acquisition of video by means of a digital camera in coordination with an Arduino Nano microcontroller.
* Measurement and recording of the different variables associated with the sensors of the system.
* Motors control.
* Battery voltage monitoring for energy consumption and internal temperature in coordination with an Arduino Nano microcontroller.
* Communication management with the remote control.

Power sources for the Raspberry Pi 3, an Arduino Nano microcontroller, and digital sensors, and the other one is a bank of five batteries of to power five thrusters’ motors. The navigation control shown in (Caption)consists of buttons series interconnected by an Ethernet network cable to onboard computer, from which the user controls the ROV through a graphical user interface (GUI).



The thrusters are controlled by a L293D motor drivers, with the speed control having as an input a pulse width modulation (PWM) signal generated by the raspberry pi 3 microcontroller. In this project, we're using python scripts run on a Raspberry Pi 3 to set GPIO outputs to an L293D motor controller IC and run thruster in either direction at any speed. The Raspberry Pi has 40 open GPIO pins. GPIO stands for "General Purpose Input/Output", which means these pins can either send electrical signals to drive hardware or receive them and read sensor data. We're using them as outputs, to send signals to the L293D IC Chip, which is just a chip used to control thrusters. The way we'll control the speed of the motor is by using a python module called PWM. PWM controls the amount of time a voltage is on by flipping between high and low for a set amount of time. The amount of time the voltage is high is called the 'duty' or 'duty cycle', and whatever percentage that is will be the percentage of power the motor runs on. The L293D motor IC uses two pins referred to as inputs to sense the desired direction of the output, and another pin called Enable to sense on/Off. So, in our code, with the Enable pin On, if we want the motor to spin forward, we'll set input 1 to 'True' or 'HIGH', and input 2 to 'False' or 'LOW'. And if we want it to spin backwards, we'll set input 1 to 'False' or 'LOW' and input 2 to 'True' or 'HIGH'. If both inputs are True or both are False, the motor will not run.

|  |
| --- |
| A circuit board with text and numbers  Description automatically generated |

**Tether management system**

For communication with the operator, ROVs use tether, high-bandwidth communication cable. The fiber optic tether enables the transfer of high-definition videos.



**Figure 13 A ethernet cable used for transmission of data.**

**Code implementation**

Python is the base of modern programming, and it is an excellent language to program microprocessors such as Arduino.

## 3.2 Fabrication and assembly

Implementing the ROV design will follow a series of processes which involves the acquisition of materials, parts, electrical components and integration of the software and hardware, as well as assembly of the ROV. PVC will be used for construction of the main hull in view of space required by the internal components while keeping the size minimal. The motor mounts will be made using ‘L’ clamps and at the front, a clear acrylic dome will be installed to enable the survey without exposing the camera to water and eliminating the need for special waterproofing. LEDs will be installed for illumination. The immersion motors are to be mounted inside a pipe using aluminum mountings and three holes drilled in the back cover for the tether and propulsion motor wires. The propulsion motor mountings are to be made so as to minimize the effective width of the ROV while maintaining a safe distance of the motors from the tether to prevent entangling with the propellers. To prevent water leakage into the ROV and keep the electronic components safe, an O-ring will be used to seal off the electrical compartment.

## Testing

Testing is a critical phase in the design of a ROV, just like any other mechanical system to ensure that it functions reliably and safely in various underwater environments. The tests are aimed to verify that the water does not leak inside the ROV capsule and prove its buoyancy in the water. The ROV is expected to achieve maneuverability in the main axes of motion and around an obstacle. The deigned ROV should attain a depth of 1 meter. In addition, an obstacle will be added in the controlled test environment to monitor the maneuverability of the ROV.

# Results and Discussion

## Task 1: Research and selection of potential materials for key components.

When designing an underwater Remotely Operated Vehicle, careful selection of materials for key components is crucial to ensure reliability, performance, cost effectiveness, and longevity. The ROV is to conduct hull inspection on a vessel and thus the material selected should withstand the pressure experienced at the drought of the vessel. Settling on an average depth of 10m, which is greater than the draughts of majority small marine vessels such as Aylah 1 and Falcao, the pressure is calculated.

**Calculation of Pressure**

Pressure is calculated using equation (Caption)

Where P is hydrostatic pressure in MPa, h is 10m, g is 9.81m/s2 and density of seawater is 1025 kg/m3

|  |  |
| --- | --- |
|  | **10** |

**Selection of materials**

The frame of the ROV, the sealing compartments and the end caps are the three key components of focus in material selection.

1. **The Frame**

The frame is the structural architecture that will hold the components that will drive and execute the functions of the ROV. The structure is to meet the following requirements:

* Withstand pressure at the designed depth
* Cost effective
* Light in weight to ensure a neutral or negative buoyancy
* Resistive to water corrosion.
* Easily machinable to form the desired plan.
* Easily available.
* Sturdy and robust to provide support for the internal components
* Environmentally friendly
* Low magnetic properties to reduce the interference of the output of the sensors.

The materials proposed include aluminium, Stainless Steel, unplasticized PVC (Polyvinyl Chloride), and Acrylic- Plexiglass. The general properties of the material are shown in the table (References)

**Table 2 pros and cons of proposed material**

|  |  |  |  |
| --- | --- | --- | --- |
| Material | Properties | Pros | Cons |
| PVC | Density: 1.38 - 1.41 g/cm³  Young’s modulus: 2.4 - 4 GPa  Poisson’s ratio: 0.38 - 0.42  Yield strength.: 50 - 55 MPa  Ultimate Tensile Strength: 55 - 65 MPa  Thermal Conductivity: 0.14 - 0.28 W/(m·K)  Thermal Expansion Coefficient: 5 x 10⁻⁵ - 7 x 10⁻⁵  Specific Heat: 900 - 1000 J/(kg·K) | Light in weight  Excellent corrosion resistance  Cost effective.  Easy to obtain.  Durable | Susceptible to UV degradation  Lower strength in comparison to metals |
| Aluminium | Density: 2.70 g/cm³  Young's Modulus: 69  Poisson's Ratio: 0.33  Yield Strength 275.00 MPa  Ultimate Tensile Strength :310.00 MPa  Thermal Conductivity: 273 W/(m·K)  Thermal Expansion Coefficient: 2.3 x 10⁻⁵ /°C  Specific Heat: 897 J/(kg·K) | Light in weight  Desirable weight-strength ratio  East to machine and fabricate.  Good corrosion resistance | Requires proper anodizing |
| Acrylic Plexiglass | Density: 1.18 g/cm³  Young’s modulus: 2740.00Mpa  Poisson's ratio: 0.355  Yield strength: 48.90MPa  Ultimate Tensile Strength: 79.00MPa  Thermal conductivity 2.400E-04W/mmC  Thermal expansion coefficient: 5 x 10⁻⁵ /°C  Specific heat: 2437.00J/kg. K | Excellent strength -to weight ratio  Corrosion resistant  Ability to be moulded into complex shapes | Expensive  Complex manufacturing processes |
| Stainless steel | Density: 8.00 g/cm³  Young's Modulus: 193000.00 MPa  Poisson's Ratio: 0.30  Yield Strength: 250.00 MPa  Ultimate Tensile Strength: 540.00 MPa  Thermal Conductivity: 0.016 W / (mm C)  Thermal Expansion Coefficient: 1.040E-05 / C  Specific Heat: 477.00 J / (kg.K ) | High strength and durable  Good corrosion resistance | Heavy  Expensive |

## Task 2: Analysis of proposed materials

**Generating the model**

The model development study was carried out on the existing ROV of the dimensions given in table (Caption) and it was modelled to the design plan. (Reference). To maximize the resulting velocity from a given propulsive force, the ROV ought to displace the least amount of water possible along its movement. Thus, there should be an absolute minimum of surface area that obstructs water flow through the ROV yet retain structural integrity to withstand the propulsive forces without any deformation. A smaller frame that still has minimal solid obstructions is ideal, yet still places the thrusters in a manner to ensure manoeuvrability in the respective degree of freedom

  ROV Principal Dimension and Parts Plate Thickness

Principal Dimension

|  |  |  |
| --- | --- | --- |
| No. | Dimension | Overall Size (mm) |
| 1. | Length | 550 |
| 2. | Width | 550 |
| 3. | Height | 211.13 |

Plate Thickness

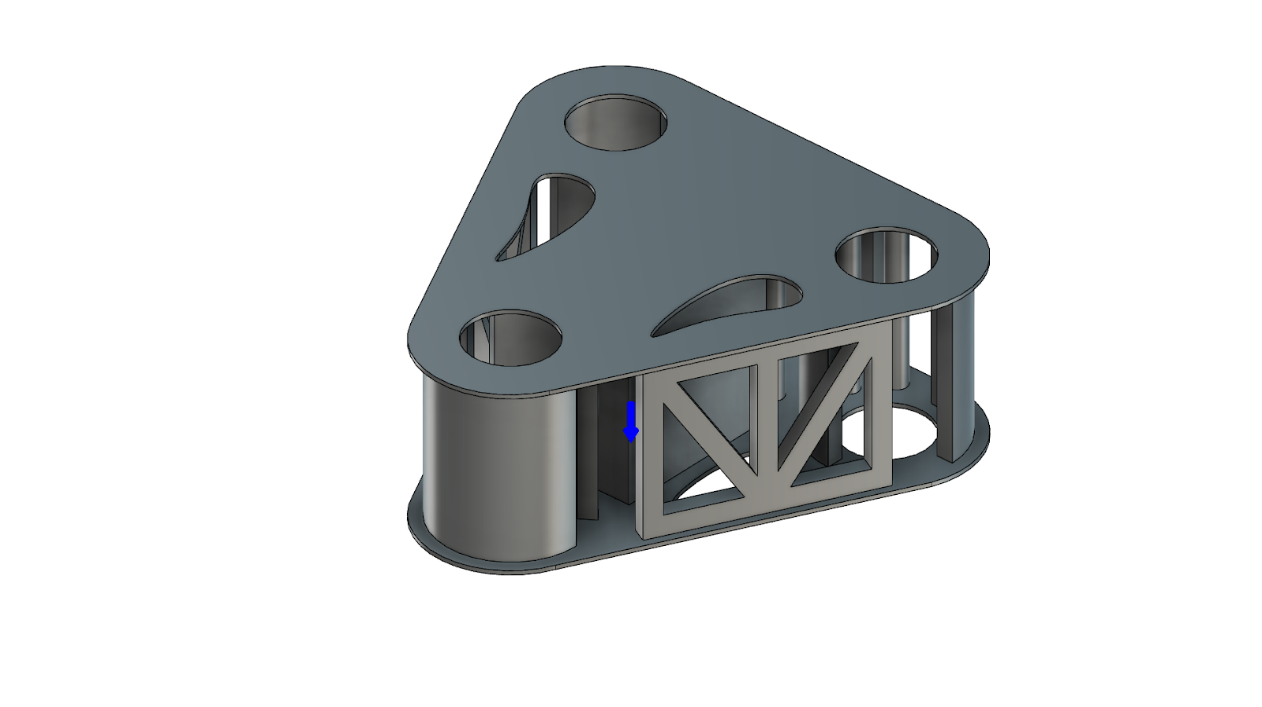
|  |  |  |
| --- | --- | --- |
| No. | Name of part | Thickness (mm) |
| 1 | Top frame | 5 |
| 2 | Bottom frame | 5 |

The design considerations for the frame are:

 • Simplicity and rigidity

 • Symmetry in thrusters’ distribution to minimize undesired reaction moments.

• The vehicle should be nearly neutrally buoyant, and the centre of buoyancy should be placed above the centre of gravity to generate righting moment.



**Finite Element Analysis**

Selection of the best material for the frame entails subjecting the proposed materials to a stress and deformation analysis test under load. During the test, a series of pressures of 101043 Pa, 505215 Pa, and 1,010,430 Pa was subjected to the model design of each of the materials for the depths of 10m, 50m, and 100m respectively and the results were reviewed.

The analysis uses a pressure test in Fusion360 to analyse the mechanical structure for maximum stress and deformation. The analysis involved adjusting the different materials according to the variations to be analysed. The mechanical properties, thickness, and allowable values of each material are summarized in table (Caption) below.

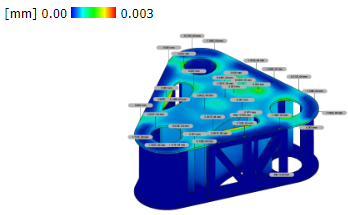
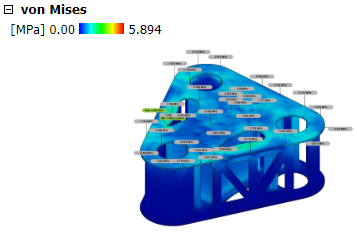
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Properties | Steel | Aluminium | PVC | Acrylic |
| Density | 7.850E-06 kg / mm^3 | 2.700E-06 kg / mm^3 | 1.300E-06 kg / mm^3 | 1.188E-06 kg / mm^3 |
| Young’s Modulus | 210000.00 MPa | 68900.00 MPa | 6.97 MPa | 2740.00 MPa |
| Poisson’s Ratio | 0.30 | 0.33 | 0.39 | 0.355 |
| Yield Strength | 207.00 MPa | 275.00 MPa | 11.00 MPa | 48.90 MPa |
| Ultimate Tensile Strength | 345.00 MPa | 310.00 MPa | 11.00 MPa | 79.80 MPa |

The analysis was taken through a series of 3 depth variations i.e. 10m, 50m, and 100m. Table (Caption) below shows the hydrostatic load used.

|  |  |  |
| --- | --- | --- |
| No | Depth | Load (Pa) |
| 1 | 10 | 101,043 |
| 2 | 50 | 505,215 |
| 3 | 100 | 1,010,430 |

**Results and discussion**

The first analysis was carried out on each material to establish variation effect to the model with the load from 10m water depth.



Von Mises Stress and Deformation Using Steel

A blue object with white text

Description automatically generatedA blue object with white text

Description automatically generated

Von Mises Stress and Deformation Using aluminium

A blue object with white text

Description automatically generatedA blue object with white text

Description automatically generated

Von Mises Stress and Deformation using PVC.

A blue object with white text

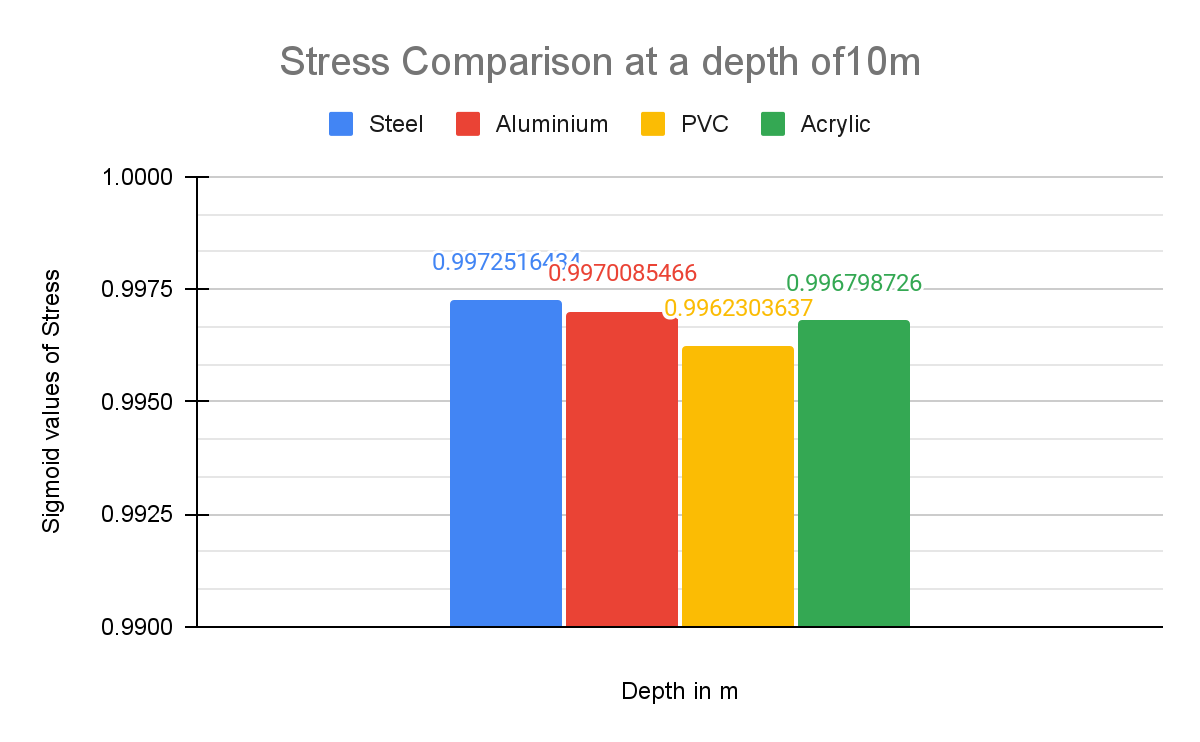
Description automatically generatedA blue object with white labels

Description automatically generated

 Von Mises Stress and Deformation Using Acrylic

The results are tabulated in table (Caption) below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Material | Deformation(mm) | Deformation (sigmoid function) | Stress (MPa) | Stress (sigmoid function) |
| 1 | Steel | 0.003 | 0.5007499994 | 5.894 | 0.9972516434 |
| 2 | Aluminium | 0.008 | 0.5019999893 | 5.809 | 0.9970085466 |
| 3 | PVC | 0.758 | 0.6809193547 | 5.577 | 0.9962303637 |
| 4 | Acrylic | 0.208 | 0.5518133302 | 5.741 | 0.996798726 |



A graph of different colored bars

Description automatically generated

## Stress and Deformation of Steel on Depth Variation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STAINLESS STEEL | | | | | | |
| No | Depth(m) | Load (Mpa) | Deformation(mm) | Deformation(sigmoid) | Stress (Mpa) | Stress(sigmoid) |
| 1 | 10 | 0.101043 | 3.00E-03 | 5.007E-01 | 5.894 | 0.997252 |
| 2 | 50 | 0.505215 | 1.50E-02 | 5.037E-01 | 29.433 | 1.000000 |
| 3 | 100 | 1.01043 | 3.10E-02 | 5.077E-01 | 58.83 | 1.000000 |

|  |  |  |
| --- | --- | --- |
| Depth (m) |  |  |
| 10 |  |  |
| 50 |  |  |
| 100 |  |  |

## Stress and Deformation of Aluminum on Depth Variation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ALUMINIUM | | | | | | |
| No | Depth(m) | Load (Mpa) | Deformation(mm) | Deformation(sigmoid) | Stress (Mpa) | Stress(sigmoid) |
| 1 | 10 | 0.101043 | 8.00E-03 | 5.020E-01 | 5.809 | 0.997009 |
| 2 | 50 | 0.505215 | 4.20E-02 | 5.105E-01 | 29.042 | 1.000000 |
| 3 | 100 | 1.01043 | 8.40E-02 | 5.210E-01 | 58.057 | 1.000000 |

|  |  |  |
| --- | --- | --- |
| Depth |  |  |
| 10 |  |  |
| 50 |  |  |
| 100 |  |  |

## Stress and Deformation of PVC on depth variation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PVC UNPLASTICIZED | | | | | | |
| No | Depth(m) | Load (Mpa) | Deformation(mm) | Deformation(sigmoid) | Stress (Mpa) | Stress(sigmoid) |
| 1 | 10 | 0.101042 | 0.758 | 6.809E-01 | 5.577 | 0.996230 |
| 2 | 50 | 0.505215 | 3.788 | 9.779E-01 | 27.889 | 1.000000 |
| 3 | 100 | 1.01043 | 7.572 | 9.995E-01 | 55.754 | 1.000000 |

|  |  |  |
| --- | --- | --- |
| Depth |  |  |
| 10 |  |  |
| 50 |  |  |
| 100 |  |  |

## Stress and Deformation of Acrylic on depth variation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ACRYLIC | | | | | | |
| No | Depth(m) | Load (Mpa) | Deformation(mm) | Deformation(sigmoid) | Stress (Mpa) | Stress(sigmoid) |
| 1 | 10 | 0.101042 | 2.08E-01 | 5.518E-01 | 5.741 | 0.996799 |
| 2 | 50 | 0.505215 | 1.042 | 7.392E-01 | 28.709 | 1.000000 |
| 3 | 100 | 1.01043 | 2.083 | 8.892E-01 | 57.394 | 1.000000 |

|  |  |  |
| --- | --- | --- |
| Depth |  |  |
| 10 |  |  |
| 50 |  |  |
| 100 |  |  |

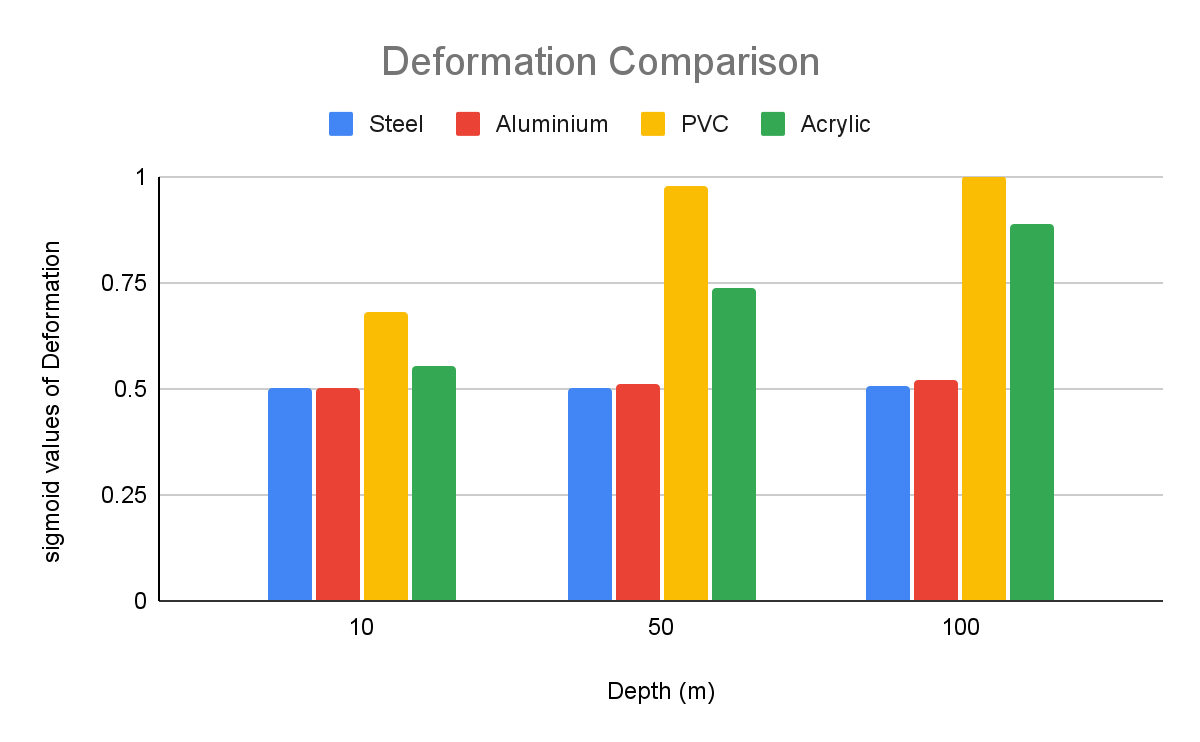
Stress and deformation limitations are calculated to determine the value of stress and deformation permitted to the material installed. Structure stress and deformation that overvalue the allowable limit are not recommended to be used in the design.

Allowable stress can be expressed by:

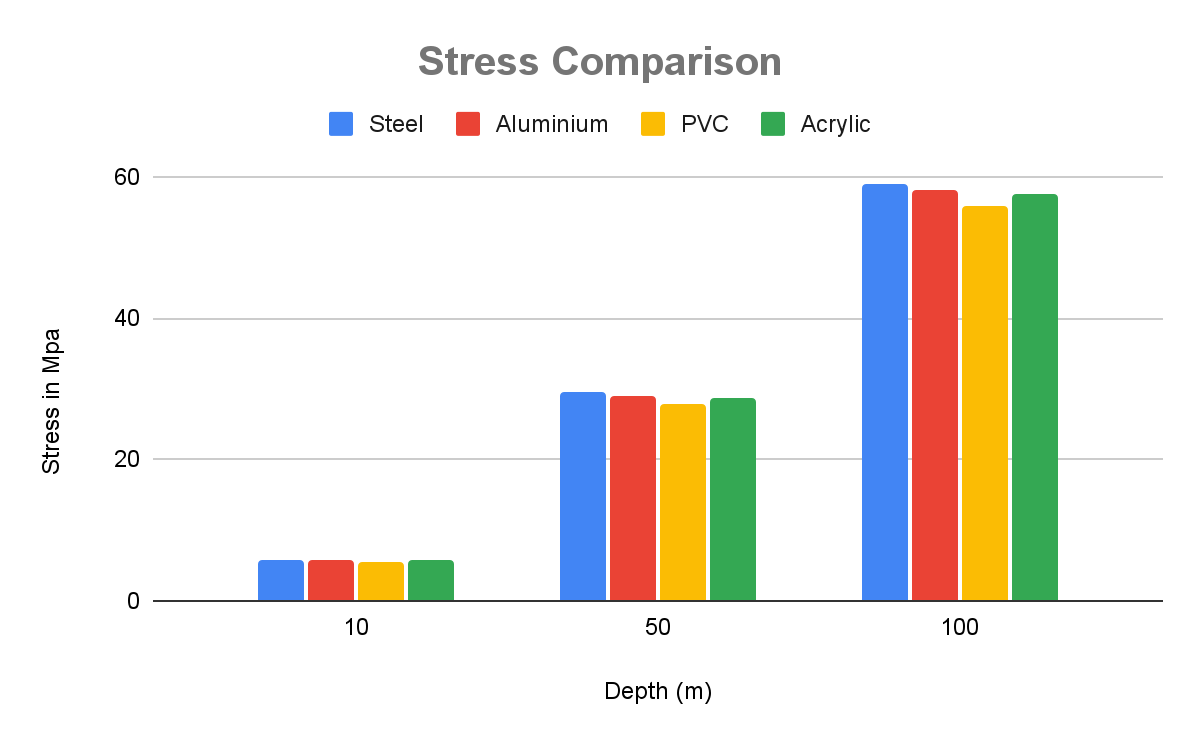
|  |  |
| --- | --- |
| {"id":"2","backgroundColor":"#ffffff","font":{"color":"#000000","family":"Times New Roman","size":12},"code":"$$\\sigma\\,=\\,\\,\\frac{Ultimate\\,Tensile\\,Strength}{Safety\\,Factor}$$","type":"$$","aid":null,"ts":1717741607408,"cs":"0S/6dSnNmAOB6eN0DuoKzA==","size":{"width":230,"height":36}} |  |

|  |  |  |
| --- | --- | --- |
| No | Material | Stress (MPa) |
| 1 | Steel | 115 |
| 2 | Aluminium | 77.5 |
| 3 | PVC | 18.75 |
| 4 | Acrylic | 10.4 |

Deformation comparison of each material at different depths



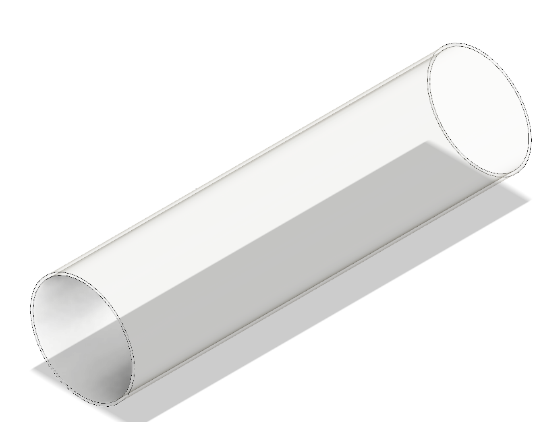
Stress comparison of each material at different depths

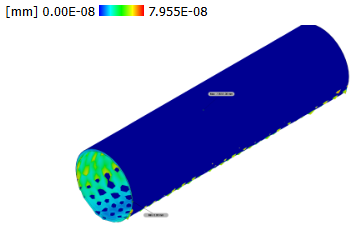
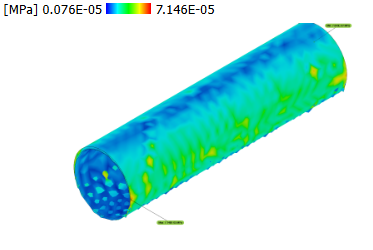


The analysis shows that the stress value of the ROV with every material variation is within allowable values in all operational conditions. The differences in stress and deformation are caused by the values of yield strength and modulus of elasticity that affect the material. However, aluminium is susceptible to corrosion in marine environment and require treatments such as anodizing which add to the cost and maintenance requirements. Further, aluminium requires techniques for welding and precise machining. Acrylic offers superior UV resistance than PVC preventing degradation from sunlight exposure. It is also lightweight in comparison to PVC and has a polished appearance.

**The Sealing Compartment and End Caps**

The sealing compartment is to protect and house the electronic components of the ROV. The material used to make the sealing compartment of electronic components in Remotely Operated Vehicles (ROVs) is critical for ensuring the integrity and functionality of the electronics in underwater environments. The proposed material is acrylic plexiglass as it can withstand the pressure at the ROV-designed depth. A clear acrylic tube is used with a view to observe all the different components mounted and to check the assembly. End caps are used to enclose the openings of the cylindrical sealing compartments. They are to be machined using Teflon, which is commonly used on bearings and low-friction bushings due to its: low friction, thermal resistance, and corrosion resistance. The design consists of 4 O-rings on the end caps as well as a wide flange nitrile gasket on the face of the dome and the fixed dome connecting the aluminium mount.  A nylon ring is machined so that it presses the transparent dome against the front cap once sealed with silicone so that no water enters. The acrylic tube is a cylinder of 400mm length and an exterior diameter of 110mm. The acrylic tube's inner diameter is about 105mm. This value is the first one that provides the main dimension of the caps. Then the cap needs to be 105mm as much. (Reference). The closed body of the ROV is the area experiencing hydrostatic pressure





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Depth(m) | Load (Mpa) | Deformation(mm) | Stress (Mpa) |
| 1 | 100 | 1.010430 | 7.955E-08 | 7.146E-05 |

From the analysis, we can observe that the acrylic cylindrical compartment holds under the high pressure experienced at a depth of 100m.

**c) Propulsion system**

The propulsion system allows the ROV to utilize underwater thrusters in a specific configuration to grant an operational range of motion converting input energy into useful thrust. The propulsion system consists of motors and propellers - a combination called a thruster. The size of the propeller is described by a pitch and diameter number. Rake is the degree that the blades of a propeller slant forward or backwards in relation to the hub which is the centre section of the propeller. The aim of the design of a ROV propulsion system is to have high thrust to physical size and power input ratios.

A propeller is selected with consideration for performance, speed, weight and the overall size of the ROV. Ducted propellers move more liquid, and they increase the efficiency depending on the flow rate. The selected propeller features two blades, a small diameter, a high rake angle and nylon polymer construction. Two bladed propellers are chosen because of the size of the ROV and the overall cost of the propeller. The smaller diameter propeller helps in avoiding collision with any of the frame components and to provide maximum thrust capabilities from the thrusters selected. A high rake angle allows for faster acceleration at lower rotations per minute thus allowing the ROV to respond rapidly to changes and reposition. Due to weight and cost restrictions, nylon plastic propellers are selected to ensure the best performance to cost ratio.

**Thruster design**

 As the propeller rotates in ROV, high-pressure and low-pressure areas are formed between the blades. The thrust needed for the movement of the ROV is formed as the fluid moves from the high-pressure area to the low-pressure area. The design of the thrusters ought to allow for components to be replaced in future in case of a component upgrade or a part failure. The individual thrusters are to have identical propellers to allow for each propeller to have uniform performance during operation. This allows the operator to have more uniform control of the ROV despite external water conditions. As the propeller moves, there are losses at the tip of the propeller blades. To minimize the losses, a duct is designed around the propeller. The duct is designed with an aerodynamic structure to increase the efficiency of the design.

Brushless motor thrusters are chosen for the ROV since they are efficient and durable. The diameter of the propeller is chosen to be bigger than the motor diameter for enhanced thrust efficiency, reduced cavitation, lower power consumption, improved control and stability, optimized flow characteristics and hydrodynamic performance. The pitch of the blades depends on the diameter and the rotational speed of the motor in RPMs. The width of the blade determines the amount of water it pushes, thus light blades are used for higher speed applications. 3D-printed nozzles are developed to maximize thrust and efficiency according to the **Rice nozzle principles(reference).** Nozzles also minimize cavitation. They are designed to produce forward and reverse thrust. Each thruster is controlled using an electronic speed controller (ESC)

**Thruster configuration**

 The ROV is designed with five thrusters. The thrusters mounted involve two thrusters moving in the horizontal motion along with three vertical thrusters to dive. Two thrusters are placed on the back of the ROV equidistant from the centre of momentum in the X-direction to minimize rotation. They provide translational motion in the X-direction. The remaining three thrusters are placed on the top of the ROV which provides the motion in the Z-direction. They are placed along the X-axis for the ROV to obtain pitch rotation by only firing the front thruster for pitch down and firing only the back thruster for pitch up. This configuration allows to actively control the system in six degrees of freedom. (Reference) The basis of this ROV system consists of 5 Blue Robotics T100 thrusters, which have the following technical characteristics: (Reference)

|  |  |
| --- | --- |
| Property | Description |
| Thrust | 14.7 N forward and 11.77N reverse |
| Dimensions | 72 by 92 mm |
| Mass | 0.075kg |
| Electric Power | Up to 150W when powered from 12V DC |
| Supply voltage | 12-24V |

## Task 3: Determining the weight and displacement of the ROV.

Determining the total weight of a Remotely Operated Vehicle (ROV) involves listing all its components, determining the individual weights, summing the weights and considering any assembly materials. The information on the weight of the ROV is crucial for designing the buoyancy system and ensuring that the ROV operates effectively under water. This information is crucial for designing the buoyancy system and ensuring the ROV operates effectively underwater. Table (Caption) below shows the ROV components and their corresponding mass. It also shows the total weight of the ROV.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Qty | Mass [ Kg] | Size (mm) | Total mass [ Kg] |
| Frame | | | | |
| Acrylic Plexiglass | 1 | 0.45 | 400 x D110 | 0.45 |
| O-ring seals | 6 | 0.086 | - | 0.516 |
| Frame | 1 | 13.39325 | 550 x 550 x 210.11 | 13.39325 |
| Buoyant | - | - |  | - |
| Ballast | - | - |  | - |
| Electrical | | | | |
| Battery for motors | 6 | 0.2 | 94 x 94 | 1.2 |
| Battery for electrical components | 1 | 0.2 |  | 0.2 |
| Raspberry Pi 3 | 1 | 0.1 | 8.5× 5.6× 2 | 0.1 |
| Heatsinks for raspberry Pi 3 | 3 | 0.05 | 14 x 14 x 6 | 0.15 |
| Relays for motors | 6 | 0.015 | 70 × 50 × 20 | 0.09 |
| ESC30A | 6 | 0.03 | 139 x 28 | 0.18 |
| Brushless Motor | 6 | 0.06 | 50 x 30 x 30 | 0.36 |
| Camera | 1 | 0.063 | 70 x 50.8 x 43.9 | 0.063 |
| LED lights | 2 | 0.05 |  | 0.1 |
| Led driver | 2 | 0.004 | 25.3 x 25.3 x 10.3 | 0.08 |
| waterproof LED Lens | 2 | 0.025 |  | 0.05 |
| SD card | 1 | 0.002 | 32.0 × 24.0 × 2.1 | 0.002 |
| USB-type A connector | 1 | 0.04 |  | 0.04 |
| MOSFET IRF1404 | 1 | 0.02 | ‎5 x 2.76 x 0.31 | 0.02 |
| Pressure sensor | 1 | 0.02 |  | 0.02 |
| Nylon plastic | 1 | 0.5 |  | 0.5 |
| Pixhawk controller | 1 | 0.05 |  | 0.05 |
| Motor driver | 2 | 0.06 | 56 x 57 | 0.12 |
| Pulse width modulation | 6 | 0.09 | 100 × 80 × 40 | 0.54 |
|  |  |  |  |  |
| Propulsion | | | | |
| Thrusters | 6 | 0.075 | ‎72 x 92 | 0.45 |
|  | | Total Mass |  | 18.67425 |
|  | | Total weight |  | 183.1943925 |

From table (Caption),

The estimated mass of the ROV is 18.67425kg, estimated weight, taking gravitational pull as 9.81m/s2, is 183.1943925N, and the volume is 1.127E+7mm3.

## Task 4. Calculating the buoyant force, Hydrodynamic drag force, power of the motor, and the thrust Force.

To select a suitable motor for a Remotely Operated Vehicle (ROV) to achieve a designed desired speed of 2 m/s, the factors considered include the hydrodynamic drag, thrust required and power needed. The steps towards achieving these specifications are systematically discussed below:

**Calculation of Upthrust Force:**

To find the force of upthrust experienced by the ROV, equation 2 is used

|  |  |
| --- | --- |
|  | **11** |

**Calculation of force of buoyancy:**

To find the force of buoyancy experienced by the ROV, equation 3 is used

|  |  |
| --- | --- |
|  | **12** |

**Calculate the hydrodynamic drag**

The thrust required to achieve a speed of 2m/s depends on the hydrodynamic drag force acting on the ROV. The drag force on an ROV moving through water can be calculated using the drag equation (Caption)

|  |  |
| --- | --- |
|  | **13** |

Where:

 ρ is the density of sea water= 1025kg/m3 and gravitational acceleration = 9.8m/s2

 v is the desired velocity (2 m/s)

 Cd​ is the drag coefficient

A is the characteristic cross-sectional area of the ROV

The Reynolds number is used to predict the flow patterns in fluid dynamics. It is a relation between density ρ, characteristic length L, velocity, U and fluid viscosity μ. The Reynolds number for the ROV in the water with ρ = 1025 kg/m³ and μ = 1.31 Cp =1.31 x 10³ kg/ms and moving at 2 m/s is calculated using equation (Citation) and the actual value calculated in equation (Citation)

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |



The Reynolds number exceed 4000 signifying the flow to be turbulent. For turbulent flow around a smooth sphere, the coefficient of drag levels off to an approximately constant value as the Reynolds number increases, particularly beyond Re≈105. This levelled-off value is typically around 0.47. (Reference). The force of drag on the ROV vehicle is thus calculated by taking the values below:

Designed ROV speed = 2m/s

Density of sea water =1025kg/m3

The characteristic cross-sectional area of the ROV= 0.3015m2

Coefficient of drag = 0.47

|  |  |
| --- | --- |
|  | **15** |

However, the total drag of the ROV system is the vehicle drag and the tether drag. The coefficient of drag for tether cables used in Remotely Operated Vehicle (ROV) applications vary depending on the cable's surface characteristics, diameter, and flow conditions. The tether drag coefficient in a turbulent flow regime is approximated to be 1.2(Reference). With a tether cable of 0.01m and a length of 10m, the characteristic area of the tether cable is obtained as 0.004712388m3. The drag force on the tether cable is thus calculated in the equation:

|  |  |
| --- | --- |
|  | **16** |

|  |  |
| --- | --- |
|  | **17** |

The total hydrodynamic drag force on the ROV system is thus the summation of the tether drag and the vehicle drag, and it is given by equation (Caption)

|  |  |
| --- | --- |
|  | **18** |

**Power of motor analysis**

To calculate the power of a motor needed to achieve a speed of 2m/s and overcome a drag force of 302.08772N, the fundamental relationship between power, force and velocity is applied in equation:

|  |  |
| --- | --- |
|  | **19** |

The required power output for the motor is the rate of energy change needed to move the system.

|  |  |
| --- | --- |
|  | **20** |

The thrust force, FT, to achieve a desired speed will be given by:

|  |  |
| --- | --- |
|  | **21** |

Where.

P=Power in watts

F=Force in newtons

V=velocity in m/s

|  |  |
| --- | --- |
|  | **22** |

|  |  |
| --- | --- |
| W | **23** |

By including a design factor of safety of 1.5, the maximum power of the motor is obtained from equation (Caption).

|  |  |
| --- | --- |
| \*  W | **24** |

The motors need to produce 1132.82895W, whish is equivalent to 1.519352 HP to maintain a maximum speed of 2.5m/s while overcoming the ROV system drag force of 302.08772N. It is assumed that the drag force remains constant at the designed speed.

**TASK 5: THE ELECTRICAL DESIGN**

The electronic circuitry is responsible for signalling to and from the control side and data acquisition during the inspection. The electronic equipment is securely sealed inside the electronic enclosure. The sensors required for the inspection are added as payload and integrated with the circuitry. Ethernet cable used as the tether for transmission of control signals, the camera feed, and the sensor data from the ROV**.**

Sensors are vital for any underwater inspection and survey to happen effectively. Various sensors

specific to the application are equipped to carry out detailed underwater monitoring. These sensors also serve as a health monitor to prevent its navigation beyond the operating range. The ROV was equipped with a temperature sensor with a usable temperature to monitor the temperature underwater for any alarming temperature variations. The pressure sensor was used to provide real-time pressure data which helps in determining the depth of the ROV. It also serves as an external aiding sensor for position tracking by providing depth. The data from these sensors are serially communicated to the microcontroller which is connected to the processor. They are powered by a 5V converter. These sensors were placed along the inner walls of the ROV frame and wired into the enclosure via a slotted acrylic flange.

The IMU contains a gyroscope and accelerometer for providing compensated incremental angle and velocity data for navigation and angular rates, linear accelerations, and magnetic fields for control through a digital serial interface bus. The IMU measures angular rates(rad/sec), linear acceleration(m/sec2), and magnetic fields(milli-gauss) in a body mounted strap down configuration along the three axes which are fed to a sensor fusion algorithm to obtain the attitude and heading of the ROV with respect to the earth’s frame. The navigation data involving the incremental angles and velocities along the three axes are primarily used for positioning the ROV while the control data is used in the determination of the orientation in Euler angles (roll, pitch, and yaw) and control of the vehicle. Since stand-alone gyroscope has no frame of reference in contrast to accelerometer with earth’s acceleration due to gravity as the reference and magnetometer with the earth’s magnetic north as a frame of reference, all the three sensors are fused to obtain the complete

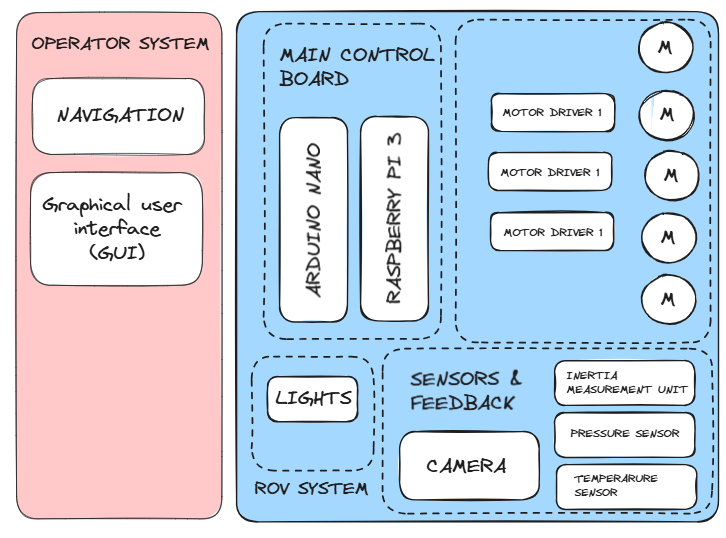
orientation of the sensor with respect to a navigation frame. The IMU was powered with a 5V converter. It was mounted vertically on the flat tray at the centre of gravity of the vehicle such that the x-axis of the IMU coincides with the direction of the vehicle’s heading. Further calibration of the IMU after placing it at the COG was done to reduce the bias and obtain precise values.

**ELECTRONIC SYSTEM**

Figure (Caption) shows the block diagram of the proposed hardware that consists of two parts: the ROV vehicle system and the operator system. These two subsystems are communicated through an Ethernet Cable so that the user can control the ROV from a personal computer, laptop. The ROV hardware involves a Raspberry Pi 3 that is responsible for executing parallel tasks, such as:

* 1. Acquisition of video by means of a digital camera in coordination with an Arduino Nano microcontroller.
  2. Measurement and recording of the different variables associated with the sensors of the system.
  3. Motors control\*
  4. Battery voltage monitoring for energy consumption and internal temperature in coordination with an Arduino Nano microcontroller.
  5. Communication management with the remote control. Power sources for the Raspberry Pi 3, an Arduino Nano microcontroller, and digital sensors, and the other one is a bank of six batteries of to power five thrusters’ motors.

The navigation control shown in (Caption)consists of buttons series interconnected by an Ethernet network cable to onboard computer, from which the user controls the ROV through a graphical user interface (GUI).



The thrusters are controlled by a L293D motor drivers, with the speed control having as an input a pulse width modulation (PWM) signal generated by the raspberry pi 3 microcontroller. In this project, we're using python scripts run on a Raspberry Pi 3 to set GPIO outputs to an L293D motor controller IC and run thruster in either direction at any speed. The Raspberry Pi has 40 open GPIO pins. GPIO stands for "General Purpose Input/Output", which means these pins can either send electrical signals to drive hardware or receive them and read sensor data. We're using them as outputs, to send signals to the L293D IC Chip, which is just a chip used to control thrusters. The way we'll control the speed of the motor is by using a python module called PWM. PWM controls the amount of time a voltage is on by flipping between high and low for a set amount of time. The amount of time the voltage is high is called the 'duty' or 'duty cycle', and whatever percentage that is will be the percentage of power the motor runs on. The L293D motor IC uses two pins referred to as inputs to sense the desired direction of the output, and another pin called Enable to sense on/Off. So, in our code, with the Enable pin On, if we want the motor to spin forward, we'll set input 1 to 'True' or 'HIGH', and input 2 to 'False' or 'LOW'. And if we want it to spin backwards, we'll set input 1 to 'False' or 'LOW' and input 2 to 'True' or 'HIGH'. If both inputs are True or both are False, the motor will not run.

A circuit board with text and numbers

Description automatically generated

|  |
| --- |
| # !/usr/bin/env python3 # Modules from goto import with\_goto from stddef import \* import var import pio import resource from datetime import datetime import RPi.GPIO as GPIO import time import cpu import FileStore import VFP  def peripheral\_setup () : # Peripheral Constructors  pio.cpu=cpu.CPU ()  pio.storage=FileStore.FileStore ()  pio.server=VFP.VfpServer ()  pio.storage.begin ()  pio.server.begin (0) # Install interrupt handlers  def peripheral\_loop () :  pio.server.poll ()   class MotorController:     def \_\_init\_\_(self, pwm\_pin, in1\_pin, in2\_pin, in3\_pin=None, in4\_pin=None, in5\_pin=None, in6\_pin=None):         self.pwm\_pin = pwm\_pin         self.in1\_pin = in1\_pin         self.in2\_pin = in2\_pin         self.in3\_pin = in3\_pin         self.in4\_pin = in4\_pin         self.in5\_pin = in5\_pin         self.in6\_pin = in6\_pin          GPIO.setup(self.pwm\_pin, GPIO.OUT)         if self.in1\_pin is not None:             GPIO.setup(self.in1\_pin, GPIO.OUT)         if self.in2\_pin is not None:             GPIO.setup(self.in2\_pin, GPIO.OUT)         if self.in3\_pin is not None:             GPIO.setup(self.in3\_pin, GPIO.OUT)         if self.in4\_pin is not None:             GPIO.setup(self.in4\_pin, GPIO.OUT)         if self.in5\_pin is not None:             GPIO.setup(self.in5\_pin, GPIO.OUT)         if self.in6\_pin is not None:             GPIO.setup(self.in6\_pin, GPIO.OUT)          self.pwm = GPIO.PWM(self.pwm\_pin, 100)         self.pwm.start(0)      def move(self, speed=100):         if self.in1\_pin is not None:             GPIO.output(self.in1\_pin, GPIO.HIGH)         if self.in2\_pin is not None:             GPIO.output(self.in2\_pin, GPIO.LOW)         if self.in3\_pin is not None:             GPIO.output(self.in3\_pin, GPIO.HIGH)         if self.in4\_pin is not None:             GPIO.output(self.in4\_pin, GPIO.LOW)         if self.in5\_pin is not None:             GPIO.output(self.in5\_pin, GPIO.HIGH)         if self.in6\_pin is not None:             GPIO.output(self.in6\_pin, GPIO.LOW)                  self.pwm.ChangeDutyCycle(speed)      def stop(self):         self.pwm.stop()         GPIO.cleanup()  class MotorSystem:     def \_\_init\_\_(self):         self.ZN\_AXIS\_MOTORS = MotorController(26, 22, 23, 25, 24)         self.YN\_AXIS\_MOTORS = MotorController(21, 18, 27, 4, 17)         self.XN\_AXIS\_MOTORS = MotorController(21, 27, 18)                  self.ZP\_AXIS\_MOTORS = MotorController(26, 23, 22, 24, 25)         self.YP\_AXIS\_MOTORS = MotorController(21, 27, 18, 17, 4)         self.XP\_AXIS\_MOTORS = MotorController(21, 17, 4)                  self.UP\_DIRECTION\_PIN = 6  # GO UP         self.DOWN\_DIRECTION\_PIN = 5   # GO DOWN         self.LEFT\_DIRECTION\_PIN = 20   # GO LEFT         self.RIGHT\_DIRECTION\_PIN = 19  # GO RIGHT         self.FORWARD\_DIRECTION\_PIN = 16  # GO FORWARD         self.BACK\_DIRECTION\_PIN = 13  # REVERSE          GPIO.setup(self.UP\_DIRECTION\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)         GPIO.setup(self.DOWN\_DIRECTION\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)         GPIO.setup(self.LEFT\_DIRECTION\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)         GPIO.setup(self.RIGHT\_DIRECTION\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)         GPIO.setup(self.FORWARD\_DIRECTION\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)         GPIO.setup(self.BACK\_DIRECTION\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)          GPIO.add\_event\_detect(self.UP\_DIRECTION\_PIN, GPIO.BOTH, callback=self.UP\_DIRECTION , bouncetime=200)         GPIO.add\_event\_detect(self.DOWN\_DIRECTION\_PIN, GPIO.BOTH, callback=self.DOWN\_DIRECTION, bouncetime=200)         GPIO.add\_event\_detect(self.LEFT\_DIRECTION\_PIN, GPIO.BOTH, callback=self.LEFT\_DIRECTION, bouncetime=200)         GPIO.add\_event\_detect(self.RIGHT\_DIRECTION\_PIN, GPIO.BOTH, callback=self.RIGHT\_DIRECTION , bouncetime=200)         GPIO.add\_event\_detect(self.FORWARD\_DIRECTION\_PIN, GPIO.BOTH, callback=self.FORWARD\_DIRECTION, bouncetime=200)         GPIO.add\_event\_detect(self.BACK\_DIRECTION\_PIN, GPIO.BOTH, callback=self.BACK\_DIRECTION, bouncetime=200)      def UP\_DIRECTION(self):         if GPIO.input(self.UP\_DIRECTION\_PIN) == GPIO.LOW:             self.ZP\_AXIS\_MOTORS.move()         else:             self.ZP\_AXIS\_MOTORS.stop()          def DOWN\_DIRECTION(self):         if GPIO.input(self.DOWN\_DIRECTION\_PIN) == GPIO.LOW:             self.ZN\_AXIS\_MOTORS.move()         else:             self.ZN\_AXIS\_MOTORS.stop()      def LEFT\_DIRECTION(self):         if GPIO.input(self.LEFT\_DIRECTION\_PIN) == GPIO.LOW:             self.XN\_AXIS\_MOTORS.move()         else:             self.XN\_AXIS\_MOTORS.stop()                  def RIGHT\_DIRECTION(self):         if GPIO.input(self.RIGHT\_DIRECTION\_PIN) == GPIO.LOW:             self.XP\_AXIS\_MOTORS.move()         else:             self.XP\_AXIS\_MOTORS.stop()                  def FORWARD\_DIRECTION(self):         if GPIO.input(self.FORWARD\_DIRECTION\_PIN) == GPIO.LOW:             self.YN\_AXIS\_MOTORS.move()         else:             self.YN\_AXIS\_MOTORS.stop()      def BACK\_DIRECTION(self):         if GPIO.input(self.BACK\_DIRECTION\_PIN) == GPIO.LOW:             self.YP\_AXIS\_MOTORS.move()         else:             self.YP\_AXIS\_MOTORS.stop()  def main():     print("Press the button to start the motor.")     # SETUP      peripheral\_setup()     motor\_system = MotorSystem()     try:         while True:             time.sleep(1)  # Wait for button press      except KeyboardInterrupt:         pass     finally:         motor\_system.ZN\_AXIS\_MOTORS.stop()         motor\_system.XN\_AXIS\_MOTORS.stop()         motor\_system.YN\_AXIS\_MOTORS.stop()         motor\_system.ZP\_AXIS\_MOTORS.stop()         motor\_system.XP\_AXIS\_MOTORS.stop()         motor\_system.YP\_AXIS\_MOTORS.stop()   # Command line execution if \_\_name\_\_ == '\_\_main\_\_' :    main() |

A screenshot of a computer program

Description automatically generated

A diagram of a motor driver

Description automatically generated

A diagram of a camera module

Description automatically generated

|  |
| --- |
| #include <stdint.h> #include <avr/io.h> #include <util/twi.h> #include <util/delay.h> #include <avr/pgmspace.h>  #define F\_CPU 16000000UL #define vga   0 #define qvga  1 #define qqvga   2 #define yuv422  0 #define rgb565  1 #define bayerRGB  2 #define camAddr\_WR  0x42 #define camAddr\_RD  0x43  /\* Registers \*/ #define REG\_GAIN    0x00  /\* Gain lower 8 bits (rest in vref) \*/ #define REG\_BLUE    0x01  /\* blue gain \*/ #define REG\_RED       0x02  /\* red gain \*/ #define REG\_VREF    0x03  /\* Pieces of GAIN, VSTART, VSTOP \*/ #define REG\_COM1    0x04  /\* Control 1 \*/ #define COM1\_CCIR656  0x40    /\* CCIR656 enable \*/ #define REG\_BAVE    0x05  /\* U/B Average level \*/ #define REG\_GbAVE   0x06  /\* Y/Gb Average level \*/ #define REG\_AECHH   0x07  /\* AEC MS 5 bits \*/ #define REG\_RAVE    0x08  /\* V/R Average level \*/ #define REG\_COM2    0x09  /\* Control 2 \*/ #define COM2\_SSLEEP         0x10  /\* Soft sleep mode \*/ #define REG\_PID           0x0a  /\* Product ID MSB \*/ #define REG\_VER           0x0b  /\* Product ID LSB \*/ #define REG\_COM3    0x0c  /\* Control 3 \*/ #define COM3\_SWAP         0x40  /\* Byte swap \*/ #define COM3\_SCALEEN          0x08  /\* Enable scaling \*/ #define COM3\_DCWEN          0x04  /\* Enable downsamp/crop/window \*/ #define REG\_COM4    0x0d  /\* Control 4 \*/ #define REG\_COM5    0x0e  /\* All "reserved" \*/ #define REG\_COM6    0x0f  /\* Control 6 \*/ #define REG\_AECH    0x10  /\* More bits of AEC value \*/ #define REG\_CLKRC   0x11  /\* Clocl control \*/ #define CLK\_EXT           0x40  /\* Use external clock directly \*/ #define CLK\_SCALE   0x3f  /\* Mask for internal clock scale \*/ #define REG\_COM7    0x12  /\* Control 7 \*/ //REG mean address. #define COM7\_RESET          0x80  /\* Register reset \*/ #define COM7\_FMT\_MASK         0x38 #define COM7\_FMT\_VGA          0x00 #define COM7\_FMT\_CIF          0x20  /\* CIF format \*/ #define COM7\_FMT\_QVGA         0x10  /\* QVGA format \*/ #define COM7\_FMT\_QCIF         0x08  /\* QCIF format \*/ #define COM7\_RGB          0x04  /\* bits 0 and 2 - RGB format \*/ #define COM7\_YUV          0x00  /\* YUV \*/ #define COM7\_BAYER          0x01  /\* Bayer format \*/ #define COM7\_PBAYER         0x05  /\* "Processed bayer" \*/ #define REG\_COM8    0x13  /\* Control 8 \*/ #define COM8\_FASTAEC          0x80  /\* Enable fast AGC/AEC \*/ #define COM8\_AECSTEP          0x40  /\* Unlimited AEC step size \*/ #define COM8\_BFILT    0x20  /\* Band filter enable \*/ #define COM8\_AGC    0x04  /\* Auto gain enable \*/ #define COM8\_AWB    0x02  /\* White balance enable \*/ #define COM8\_AEC    0x01  /\* Auto exposure enable \*/ #define REG\_COM9    0x14  /\* Control 9- gain ceiling \*/ #define REG\_COM10   0x15  /\* Control 10 \*/ #define COM10\_HSYNC         0x40  /\* HSYNC instead of HREF \*/ #define COM10\_PCLK\_HB         0x20  /\* Suppress PCLK on horiz blank \*/ #define COM10\_HREF\_REV          0x08  /\* Reverse HREF \*/ #define COM10\_VS\_LEAD         0x04  /\* VSYNC on clock leading edge \*/ #define COM10\_VS\_NEG          0x02  /\* VSYNC negative \*/ #define COM10\_HS\_NEG          0x01  /\* HSYNC negative \*/ #define REG\_HSTART    0x17  /\* Horiz start high bits \*/ #define REG\_HSTOP   0x18  /\* Horiz stop high bits \*/ #define REG\_VSTART    0x19  /\* Vert start high bits \*/ #define REG\_VSTOP   0x1a  /\* Vert stop high bits \*/ #define REG\_PSHFT   0x1b  /\* Pixel delay after HREF \*/ #define REG\_MIDH    0x1c  /\* Manuf. ID high \*/ #define REG\_MIDL    0x1d  /\* Manuf. ID low \*/ #define REG\_MVFP    0x1e  /\* Mirror / vflip \*/ #define MVFP\_MIRROR         0x20  /\* Mirror image \*/ #define MVFP\_FLIP   0x10  /\* Vertical flip \*/ #define REG\_AEW           0x24  /\* AGC upper limit \*/ #define REG\_AEB           0x25    /\* AGC lower limit \*/ #define REG\_VPT           0x26  /\* AGC/AEC fast mode op region \*/ #define REG\_HSYST   0x30  /\* HSYNC rising edge delay \*/ #define REG\_HSYEN   0x31  /\* HSYNC falling edge delay \*/ #define REG\_HREF    0x32  /\* HREF pieces \*/ #define REG\_TSLB    0x3a  /\* lots of stuff \*/ #define TSLB\_YLAST    0x04  /\* UYVY or VYUY - see com13 \*/ #define REG\_COM11   0x3b  /\* Control 11 \*/ #define COM11\_NIGHT         0x80  /\* NIght mode enable \*/ #define COM11\_NMFR          0x60  /\* Two bit NM frame rate \*/ #define COM11\_HZAUTO          0x10  /\* Auto detect 50/60 Hz \*/ #define COM11\_50HZ          0x08  /\* Manual 50Hz select \*/ #define COM11\_EXP   0x02 #define REG\_COM12   0x3c  /\* Control 12 \*/ #define COM12\_HREF          0x80  /\* HREF always \*/ #define REG\_COM13   0x3d  /\* Control 13 \*/ #define COM13\_GAMMA         0x80  /\* Gamma enable \*/ #define COM13\_UVSAT         0x40  /\* UV saturation auto adjustment \*/ #define COM13\_UVSWAP          0x01  /\* V before U - w/TSLB \*/ #define REG\_COM14   0x3e  /\* Control 14 \*/ #define COM14\_DCWEN         0x10  /\* DCW/PCLK-scale enable \*/ #define REG\_EDGE    0x3f  /\* Edge enhancement factor \*/ #define REG\_COM15   0x40  /\* Control 15 \*/ #define COM15\_R10F0         0x00  /\* Data range 10 to F0 \*/ #define COM15\_R01FE         0x80  /\*      01 to FE \*/ #define COM15\_R00FF         0xc0  /\*      00 to FF \*/ #define COM15\_RGB565          0x10  /\* RGB565 output \*/ #define COM15\_RGB555          0x30  /\* RGB555 output \*/ #define REG\_COM16   0x41  /\* Control 16 \*/ #define COM16\_AWBGAIN         0x08  /\* AWB gain enable \*/ #define REG\_COM17   0x42  /\* Control 17 \*/ #define COM17\_AECWIN          0xc0  /\* AEC window - must match COM4 \*/ #define COM17\_CBAR          0x08  /\* DSP Color bar \*/ /\* \* This matrix defines how the colors are generated, must be \* tweaked to adjust hue and saturation. \* Order: v-red, v-green, v-blue, u-red, u-green, u-blue \* They are nine-bit signed quantities, with the sign bit \* stored in0x58.Sign for v-red is bit 0, and up from there. \*/  #define REG\_CMATRIX\_BASE  0x4f #define CMATRIX\_LEN           6 #define REG\_CMATRIX\_SIGN  0x58 #define REG\_BRIGHT    0x55  /\* Brightness \*/ #define REG\_CONTRAS         0x56  /\* Contrast control \*/ #define REG\_GFIX    0x69  /\* Fix gain control \*/ #define REG\_REG76   0x76  /\* OV's name \*/ #define R76\_BLKPCOR         0x80  /\* Black pixel correction enable \*/ #define R76\_WHTPCOR         0x40  /\* White pixel correction enable \*/ #define REG\_RGB444          0x8c  /\* RGB 444 control \*/ #define R444\_ENABLE         0x02  /\* Turn on RGB444, overrides 5x5 \*/ #define R444\_RGBX   0x01  /\* Empty nibble at end \*/ #define REG\_HAECC1    0x9f  /\* Hist AEC/AGC control 1 \*/ #define REG\_HAECC2    0xa0  /\* Hist AEC/AGC control 2 \*/ #define REG\_BD50MAX         0xa5  /\* 50hz banding step limit \*/ #define REG\_HAECC3    0xa6  /\* Hist AEC/AGC control 3 \*/ #define REG\_HAECC4    0xa7  /\* Hist AEC/AGC control 4 \*/ #define REG\_HAECC5    0xa8  /\* Hist AEC/AGC control 5 \*/ #define REG\_HAECC6    0xa9  /\* Hist AEC/AGC control 6 \*/ #define REG\_HAECC7    0xaa  /\* Hist AEC/AGC control 7 \*/ #define REG\_BD60MAX         0xab  /\* 60hz banding step limit \*/ #define REG\_GAIN    0x00  /\* Gain lower 8 bits (rest in vref) \*/ #define REG\_BLUE    0x01  /\* blue gain \*/ #define REG\_RED           0x02  /\* red gain \*/ #define REG\_VREF    0x03  /\* Pieces of GAIN, VSTART, VSTOP \*/ #define REG\_COM1    0x04  /\* Control 1 \*/ #define COM1\_CCIR656          0x40  /\* CCIR656 enable \*/ #define REG\_BAVE    0x05  /\* U/B Average level \*/ #define REG\_GbAVE   0x06  /\* Y/Gb Average level \*/ #define REG\_AECHH   0x07  /\* AEC MS 5 bits \*/ #define REG\_RAVE    0x08  /\* V/R Average level \*/ #define REG\_COM2    0x09  /\* Control 2 \*/ #define COM2\_SSLEEP         0x10  /\* Soft sleep mode \*/ #define REG\_PID           0x0a  /\* Product ID MSB \*/ #define REG\_VER           0x0b  /\* Product ID LSB \*/ #define REG\_COM3    0x0c  /\* Control 3 \*/ #define COM3\_SWAP         0x40  /\* Byte swap \*/ #define COM3\_SCALEEN          0x08  /\* Enable scaling \*/ #define COM3\_DCWEN          0x04  /\* Enable downsamp/crop/window \*/ #define REG\_COM4    0x0d  /\* Control 4 \*/ #define REG\_COM5    0x0e  /\* All "reserved" \*/ #define REG\_COM6    0x0f  /\* Control 6 \*/ #define REG\_AECH    0x10  /\* More bits of AEC value \*/ #define REG\_CLKRC   0x11  /\* Clocl control \*/ #define CLK\_EXT           0x40  /\* Use external clock directly \*/ #define CLK\_SCALE   0x3f  /\* Mask for internal clock scale \*/ #define REG\_COM7    0x12  /\* Control 7 \*/ #define COM7\_RESET          0x80  /\* Register reset \*/ #define COM7\_FMT\_MASK         0x38 #define COM7\_FMT\_VGA          0x00 #define COM7\_FMT\_CIF          0x20  /\* CIF format \*/ #define COM7\_FMT\_QVGA         0x10  /\* QVGA format \*/ #define COM7\_FMT\_QCIF         0x08  /\* QCIF format \*/ #define COM7\_RGB    0x04  /\* bits 0 and 2 - RGB format \*/ #define COM7\_YUV    0x00  /\* YUV \*/ #define COM7\_BAYER          0x01  /\* Bayer format \*/ #define COM7\_PBAYER         0x05  /\* "Processed bayer" \*/ #define REG\_COM8    0x13  /\* Control 8 \*/ #define COM8\_FASTAEC          0x80  /\* Enable fast AGC/AEC \*/ #define COM8\_AECSTEP          0x40  /\* Unlimited AEC step size \*/ #define COM8\_BFILT    0x20  /\* Band filter enable \*/ #define COM8\_AGC    0x04  /\* Auto gain enable \*/ #define COM8\_AWB    0x02  /\* White balance enable \*/ #define COM8\_AEC    0x01  /\* Auto exposure enable \*/ #define REG\_COM9    0x14  /\* Control 9- gain ceiling \*/ #define REG\_COM10   0x15  /\* Control 10 \*/ #define COM10\_HSYNC         0x40  /\* HSYNC instead of HREF \*/ #define COM10\_PCLK\_HB         0x20  /\* Suppress PCLK on horiz blank \*/ #define COM10\_HREF\_REV          0x08  /\* Reverse HREF \*/ #define COM10\_VS\_LEAD           0x04  /\* VSYNC on clock leading edge \*/ #define COM10\_VS\_NEG          0x02  /\* VSYNC negative \*/ #define COM10\_HS\_NEG          0x01  /\* HSYNC negative \*/ #define REG\_HSTART    0x17  /\* Horiz start high bits \*/ #define REG\_HSTOP   0x18  /\* Horiz stop high bits \*/ #define REG\_VSTART    0x19  /\* Vert start high bits \*/ #define REG\_VSTOP   0x1a  /\* Vert stop high bits \*/ #define REG\_PSHFT   0x1b  /\* Pixel delay after HREF \*/ #define REG\_MIDH    0x1c  /\* Manuf. ID high \*/ #define REG\_MIDL    0x1d  /\* Manuf. ID low \*/ #define REG\_MVFP    0x1e  /\* Mirror / vflip \*/ #define MVFP\_MIRROR         0x20  /\* Mirror image \*/ #define MVFP\_FLIP   0x10  /\* Vertical flip \*/ #define REG\_AEW           0x24  /\* AGC upper limit \*/ #define REG\_AEB           0x25  /\* AGC lower limit \*/ #define REG\_VPT           0x26  /\* AGC/AEC fast mode op region \*/ #define REG\_HSYST   0x30  /\* HSYNC rising edge delay \*/ #define REG\_HSYEN   0x31  /\* HSYNC falling edge delay \*/ #define REG\_HREF    0x32  /\* HREF pieces \*/ #define REG\_TSLB    0x3a  /\* lots of stuff \*/ #define TSLB\_YLAST    0x04  /\* UYVY or VYUY - see com13 \*/ #define REG\_COM11   0x3b  /\* Control 11 \*/ #define COM11\_NIGHT         0x80  /\* NIght mode enable \*/ #define COM11\_NMFR          0x60  /\* Two bit NM frame rate \*/ #define COM11\_HZAUTO          0x10  /\* Auto detect 50/60 Hz \*/ #define COM11\_50HZ          0x08  /\* Manual 50Hz select \*/ #define COM11\_EXP   0x02 #define REG\_COM12   0x3c  /\* Control 12 \*/ #define COM12\_HREF          0x80  /\* HREF always \*/ #define REG\_COM13   0x3d  /\* Control 13 \*/ #define COM13\_GAMMA         0x80  /\* Gamma enable \*/ #define COM13\_UVSAT         0x40  /\* UV saturation auto adjustment \*/ #define COM13\_UVSWAP          0x01  /\* V before U - w/TSLB \*/ #define REG\_COM14   0x3e  /\* Control 14 \*/ #define COM14\_DCWEN         0x10  /\* DCW/PCLK-scale enable \*/ #define REG\_EDGE    0x3f  /\* Edge enhancement factor \*/ #define REG\_COM15   0x40  /\* Control 15 \*/ #define COM15\_R10F0         0x00  /\* Data range 10 to F0 \*/ #define COM15\_R01FE         0x80  /\*      01 to FE \*/ #define COM15\_R00FF         0xc0  /\*      00 to FF \*/ #define COM15\_RGB565          0x10  /\* RGB565 output \*/ #define COM15\_RGB555          0x30  /\* RGB555 output \*/ #define REG\_COM16   0x41  /\* Control 16 \*/ #define COM16\_AWBGAIN         0x08  /\* AWB gain enable \*/ #define REG\_COM17   0x42  /\* Control 17 \*/ #define COM17\_AECWIN          0xc0  /\* AEC window - must match COM4 \*/ #define COM17\_CBAR          0x08  /\* DSP Color bar \*/ #define CMATRIX\_LEN             6 #define REG\_BRIGHT    0x55  /\* Brightness \*/ #define REG\_REG76   0x76  /\* OV's name \*/ #define R76\_BLKPCOR         0x80  /\* Black pixel correction enable \*/ #define R76\_WHTPCOR         0x40  /\* White pixel correction enable \*/ #define REG\_RGB444          0x8c  /\* RGB 444 control \*/ #define R444\_ENABLE         0x02  /\* Turn on RGB444, overrides 5x5 \*/ #define R444\_RGBX   0x01  /\* Empty nibble at end \*/ #define REG\_HAECC1    0x9f  /\* Hist AEC/AGC control 1 \*/ #define REG\_HAECC2    0xa0  /\* Hist AEC/AGC control 2 \*/ #define REG\_BD50MAX         0xa5  /\* 50hz banding step limit \*/ #define REG\_HAECC3    0xa6  /\* Hist AEC/AGC control 3 \*/ #define REG\_HAECC4    0xa7  /\* Hist AEC/AGC control 4 \*/ #define REG\_HAECC5    0xa8  /\* Hist AEC/AGC control 5 \*/ #define REG\_HAECC6    0xa9  /\* Hist AEC/AGC control 6 \*/ #define REG\_HAECC7    0xaa  /\* Hist AEC/AGC control 7 \*/ #define REG\_BD60MAX         0xab  /\* 60hz banding step limit \*/ #define MTX1            0x4f  /\* Matrix Coefficient 1 \*/ #define MTX2            0x50  /\* Matrix Coefficient 2 \*/ #define MTX3            0x51  /\* Matrix Coefficient 3 \*/ #define MTX4            0x52  /\* Matrix Coefficient 4 \*/ #define MTX5            0x53  /\* Matrix Coefficient 5 \*/ #define MTX6            0x54  /\* Matrix Coefficient 6 \*/ #define REG\_CONTRAS         0x56  /\* Contrast control \*/ #define MTXS            0x58  /\* Matrix Coefficient Sign \*/ #define AWBC7           0x59  /\* AWB Control 7 \*/ #define AWBC8           0x5a  /\* AWB Control 8 \*/ #define AWBC9           0x5b  /\* AWB Control 9 \*/ #define AWBC10            0x5c  /\* AWB Control 10 \*/ #define AWBC11            0x5d  /\* AWB Control 11 \*/ #define AWBC12            0x5e  /\* AWB Control 12 \*/ #define REG\_GFI           0x69  /\* Fix gain control \*/ #define GGAIN           0x6a  /\* G Channel AWB Gain \*/ #define DBLV            0x6b   #define AWBCTR3           0x6c  /\* AWB Control 3 \*/ #define AWBCTR2           0x6d  /\* AWB Control 2 \*/ #define AWBCTR1           0x6e  /\* AWB Control 1 \*/ #define AWBCTR0           0x6f  /\* AWB Control 0 \*/  struct regval\_list{   uint8\_t reg\_num;   uint16\_t value; };  const struct regval\_list qvga\_ov7670[] PROGMEM = {   { REG\_COM14, 0x19 },   { 0x72, 0x11 },   { 0x73, 0xf1 },   { REG\_HSTART, 0x16 },   { REG\_HSTOP, 0x04 },   { REG\_HREF, 0xa4 },   { REG\_VSTART, 0x02 },   { REG\_VSTOP, 0x7a },   { REG\_VREF, 0x0a },   { 0xff, 0xff }, /\* END MARKER \*/ };  const struct regval\_list yuv422\_ov7670[] PROGMEM = {   { REG\_COM7, 0x0 },  /\* Selects YUV mode \*/   { REG\_RGB444, 0 },  /\* No RGB444 please \*/   { REG\_COM1, 0 },   { REG\_COM15, COM15\_R00FF },   { REG\_COM9, 0x6A }, /\* 128x gain ceiling; 0x8 is reserved bit \*/   { 0x4f, 0x80 },   /\* "matrix coefficient 1" \*/   { 0x50, 0x80 },   /\* "matrix coefficient 2" \*/   { 0x51, 0 },    /\* vb \*/   { 0x52, 0x22 },   /\* "matrix coefficient 4" \*/   { 0x53, 0x5e },   /\* "matrix coefficient 5" \*/   { 0x54, 0x80 },   /\* "matrix coefficient 6" \*/   { REG\_COM13, COM13\_UVSAT },   { 0xff, 0xff },   /\* END MARKER \*/ };  const struct regval\_list ov7670\_default\_regs[] PROGMEM = {//from the linux driver   { REG\_COM7, COM7\_RESET },   { REG\_TSLB, 0x04 }, /\* OV \*/   { REG\_COM7, 0 },  /\* VGA \*/   /\*   \* Set the hardware window.  These values from OV don't entirely   \* make sense - hstop is less than hstart.  But they work...   \*/   { REG\_HSTART, 0x13 }, { REG\_HSTOP, 0x01 },   { REG\_HREF, 0xb6 }, { REG\_VSTART, 0x02 },   { REG\_VSTOP, 0x7a }, { REG\_VREF, 0x0a },   { REG\_COM3, 0 }, { REG\_COM14, 0 },   /\* Mystery scaling numbers \*/   { 0x70, 0x3a }, { 0x71, 0x35 },   { 0x72, 0x11 }, { 0x73, 0xf0 },   { 0xa2,/\* 0x02 changed to 1\*/1 }, { REG\_COM10, 0x0 },   /\* Gamma curve values \*/   { 0x7a, 0x20 }, { 0x7b, 0x10 },   { 0x7c, 0x1e }, { 0x7d, 0x35 },   { 0x7e, 0x5a }, { 0x7f, 0x69 },   { 0x80, 0x76 }, { 0x81, 0x80 },   { 0x82, 0x88 }, { 0x83, 0x8f },   { 0x84, 0x96 }, { 0x85, 0xa3 },   { 0x86, 0xaf }, { 0x87, 0xc4 },   { 0x88, 0xd7 }, { 0x89, 0xe8 },   /\* AGC and AEC parameters.  Note we start by disabling those features,   then turn them only after tweaking the values. \*/   { REG\_COM8, COM8\_FASTAEC | COM8\_AECSTEP },   { REG\_GAIN, 0 }, { REG\_AECH, 0 },   { REG\_COM4, 0x40 }, /\* magic reserved bit \*/   { REG\_COM9, 0x18 }, /\* 4x gain + magic rsvd bit \*/   { REG\_BD50MAX, 0x05 }, { REG\_BD60MAX, 0x07 },   { REG\_AEW, 0x95 }, { REG\_AEB, 0x33 },   { REG\_VPT, 0xe3 }, { REG\_HAECC1, 0x78 },   { REG\_HAECC2, 0x68 }, { 0xa1, 0x03 }, /\* magic \*/   { REG\_HAECC3, 0xd8 }, { REG\_HAECC4, 0xd8 },   { REG\_HAECC5, 0xf0 }, { REG\_HAECC6, 0x90 },   { REG\_HAECC7, 0x94 },   { REG\_COM8, COM8\_FASTAEC | COM8\_AECSTEP | COM8\_AGC | COM8\_AEC },   { 0x30, 0 }, { 0x31, 0 },//disable some delays   /\* Almost all of these are magic "reserved" values.  \*/   { REG\_COM5, 0x61 }, { REG\_COM6, 0x4b },   { 0x16, 0x02 }, { REG\_MVFP, 0x07 },   { 0x21, 0x02 }, { 0x22, 0x91 },   { 0x29, 0x07 }, { 0x33, 0x0b },   { 0x35, 0x0b }, { 0x37, 0x1d },   { 0x38, 0x71 }, { 0x39, 0x2a },   { REG\_COM12, 0x78 }, { 0x4d, 0x40 },   { 0x4e, 0x20 }, { REG\_GFIX, 0 },   /\*{0x6b, 0x4a},\*/{ 0x74, 0x10 },   { 0x8d, 0x4f }, { 0x8e, 0 },   { 0x8f, 0 }, { 0x90, 0 },   { 0x91, 0 }, { 0x96, 0 },   { 0x9a, 0 }, { 0xb0, 0x84 },   { 0xb1, 0x0c }, { 0xb2, 0x0e },   { 0xb3, 0x82 }, { 0xb8, 0x0a },   /\* More reserved magic, some of which tweaks white balance \*/   { 0x43, 0x0a }, { 0x44, 0xf0 },   { 0x45, 0x34 }, { 0x46, 0x58 },   { 0x47, 0x28 }, { 0x48, 0x3a },   { 0x59, 0x88 }, { 0x5a, 0x88 },   { 0x5b, 0x44 }, { 0x5c, 0x67 },   { 0x5d, 0x49 }, { 0x5e, 0x0e },   { 0x6c, 0x0a }, { 0x6d, 0x55 },   { 0x6e, 0x11 }, { 0x6f, 0x9e }, /\* it was 0x9F "9e for advance AWB" \*/   { 0x6a, 0x40 }, { REG\_BLUE, 0x40 },   { REG\_RED, 0x60 },   { REG\_COM8, COM8\_FASTAEC | COM8\_AECSTEP | COM8\_AGC | COM8\_AEC | COM8\_AWB },   /\* Matrix coefficients \*/   { 0x4f, 0x80 }, { 0x50, 0x80 },   { 0x51, 0 },    { 0x52, 0x22 },   { 0x53, 0x5e }, { 0x54, 0x80 },   { 0x58, 0x9e },   { REG\_COM16, COM16\_AWBGAIN }, { REG\_EDGE, 0 },   { 0x75, 0x05 }, { REG\_REG76, 0xe1 },   { 0x4c, 0 },     { 0x77, 0x01 },   { REG\_COM13, /\*0xc3\*/0x48 }, { 0x4b, 0x09 },   { 0xc9, 0x60 },   /\*{REG\_COM16, 0x38},\*/   { 0x56, 0x40 },    { 0x34, 0x11 }, { REG\_COM11, COM11\_EXP | COM11\_HZAUTO },   { 0xa4, 0x82/\*Was 0x88\*/ }, { 0x96, 0 },   { 0x97, 0x30 }, { 0x98, 0x20 },   { 0x99, 0x30 }, { 0x9a, 0x84 },   { 0x9b, 0x29 }, { 0x9c, 0x03 },   { 0x9d, 0x4c }, { 0x9e, 0x3f },   { 0x78, 0x04 },   /\* Extra-weird stuff.  Some sort of multiplexor register \*/   { 0x79, 0x01 }, { 0xc8, 0xf0 },   { 0x79, 0x0f }, { 0xc8, 0x00 },   { 0x79, 0x10 }, { 0xc8, 0x7e },   { 0x79, 0x0a }, { 0xc8, 0x80 },   { 0x79, 0x0b }, { 0xc8, 0x01 },   { 0x79, 0x0c }, { 0xc8, 0x0f },   { 0x79, 0x0d }, { 0xc8, 0x20 },   { 0x79, 0x09 }, { 0xc8, 0x80 },   { 0x79, 0x02 }, { 0xc8, 0xc0 },   { 0x79, 0x03 }, { 0xc8, 0x40 },   { 0x79, 0x05 }, { 0xc8, 0x30 },   { 0x79, 0x26 },   { 0xff, 0xff }, /\* END MARKER \*/ };  void error\_led(void){   DDRB |= 32;//make sure led is output   while (1){//wait for reset     PORTB ^= 32;// toggle led     \_delay\_ms(100);   } }  void twiStart(void){   TWCR = \_BV(TWINT) | \_BV(TWSTA) | \_BV(TWEN);//send start   while (!(TWCR & (1 << TWINT)));//wait for start to be transmitted   if ((TWSR & 0xF8) != TW\_START)     error\_led(); }  void twiWriteByte(uint8\_t DATA, uint8\_t type){   TWDR = DATA;   TWCR = \_BV(TWINT) | \_BV(TWEN);   while (!(TWCR & (1 << TWINT))) {}   if ((TWSR & 0xF8) != type)     error\_led(); }  void twiAddr(uint8\_t addr, uint8\_t typeTWI){   TWDR = addr;//send address   TWCR = \_BV(TWINT) | \_BV(TWEN);    /\* clear interrupt to start transmission \*/   while ((TWCR & \_BV(TWINT)) == 0); /\* wait for transmission \*/   if ((TWSR & 0xF8) != typeTWI)     error\_led(); }  voidwriteReg(uint8\_t reg, uint8\_t dat){   //send start condition   twiStart();   twiAddr(camAddr\_WR, TW\_MT\_SLA\_ACK);   twiWriteByte(reg, TW\_MT\_DATA\_ACK);   twiWriteByte(dat, TW\_MT\_DATA\_ACK);   TWCR = (1 << TWINT) | (1 << TWEN) | (1 << TWSTO);//send stop   \_delay\_ms(1); }  static uint8\_t twiRd(uint8\_t nack){   if (nack){     TWCR = \_BV(TWINT) | \_BV(TWEN);     while ((TWCR & \_BV(TWINT)) == 0); /\* wait for transmission \*/     if ((TWSR & 0xF8) != TW\_MR\_DATA\_NACK)       error\_led();     return TWDR;   }   else{     TWCR = \_BV(TWINT) | \_BV(TWEN) | \_BV(TWEA);     while ((TWCR & \_BV(TWINT)) == 0); /\* wait for transmission \*/     if ((TWSR & 0xF8) != TW\_MR\_DATA\_ACK)       error\_led();     return TWDR;   } }  uint8\_t rdReg(uint8\_t reg){   uint8\_t dat;   twiStart();   twiAddr(camAddr\_WR, TW\_MT\_SLA\_ACK);   twiWriteByte(reg, TW\_MT\_DATA\_ACK);   TWCR = (1 << TWINT) | (1 << TWEN) | (1 << TWSTO);//send stop   \_delay\_ms(1);   twiStart();   twiAddr(camAddr\_RD, TW\_MR\_SLA\_ACK);   dat = twiRd(1);   TWCR = (1 << TWINT) | (1 << TWEN) | (1 << TWSTO);//send stop   \_delay\_ms(1);   return dat; }  void wrSensorRegs8\_8(const struct regval\_list reglist[]){   uint8\_t reg\_addr, reg\_val;   const struct regval\_list \*next = reglist;   while ((reg\_addr != 0xff) | (reg\_val != 0xff)){     reg\_addr = pgm\_read\_byte(&next->reg\_num);     reg\_val = pgm\_read\_byte(&next->value);    writeReg(reg\_addr, reg\_val);     next++;   } }  void setColor(void){   wrSensorRegs8\_8(yuv422\_ov7670);  // wrSensorRegs8\_8(qvga\_ov7670); }  void setResolution(void){  writeReg(REG\_COM3, 4); // REG\_COM3 enable scaling   wrSensorRegs8\_8(qvga\_ov7670); }  void camInit(void){  writeReg(0x12, 0x80);   \_delay\_ms(100);   wrSensorRegs8\_8(ov7670\_default\_regs);  writeReg(REG\_COM10, 32);//PCLK does not toggle on HBLANK. }  void arduinoUnoInut(void) {   cli();//disable interrupts     /\* Setup the 8mhz PWM clock   \* This will be on pin 11\*/   DDRB |= (1 << 3);//pin 11   ASSR &= ~(\_BV(EXCLK) | \_BV(AS2));   TCCR2A = (1 << COM2A0) | (1 << WGM21) | (1 << WGM20);   TCCR2B = (1 << WGM22) | (1 << CS20);   OCR2A = 0;//(F\_CPU)/(2\*(X+1))   DDRC &= ~15;//low d0-d3 camera   DDRD &= ~252;//d7-d4 and interrupt pins   \_delay\_ms(3000);     //set up twi for 100khz   TWSR &= ~3;//disable prescaler for TWI   TWBR = 72;//set to 100khz     //enable serial   UBRR0H = 0;   UBRR0L = 1;//0 = 2M baud rate. 1 = 1M baud. 3 = 0.5M. 7 = 250k 207 is 9600 baud rate.   UCSR0A |= 2;//double speed aysnc   UCSR0B = (1 << RXEN0) | (1 << TXEN0);//Enable receiver and transmitter   UCSR0C = 6;//async 1 stop bit 8bit char no parity bits }  void StringPgm(const char \* str){   do{       while (!(UCSR0A & (1 << UDRE0)));//wait for byte to transmit       UDR0 = pgm\_read\_byte\_near(str);       while (!(UCSR0A & (1 << UDRE0)));//wait for byte to transmit   } while (pgm\_read\_byte\_near(++str)); }  static void captureImg(uint16\_t wg, uint16\_t hg){   uint16\_t y, x;   StringPgm(PSTR("\*RDY\*"));   while (!(PIND & 8));//wait for high   while ((PIND & 8));//wait for low     y = hg;   while (y--){         x = wg;       //while (!(PIND & 256));//wait for high     while (x--){       while ((PIND & 4));//wait for low             UDR0 = (PINC & 15) | (PIND & 240);           while (!(UCSR0A & (1 << UDRE0)));//wait for byte to transmit       while (!(PIND & 4));//wait for high       while ((PIND & 4));//wait for low       while (!(PIND & 4));//wait for high     }     //  while ((PIND & 256));//wait for low   }     \_delay\_ms(100); }  void setup(){   arduinoUnoInut();   camInit();   setResolution();   setColor();  writeReg(0x11, 10); //Earlier it had the value:writeReg(0x11, 12); New version works better for me :) !!!! }  void loop(){   captureImg(320, 240); } |

# Conclusion

There are challenges experienced in hull inspections of marine vessels. These challenges include costly and time-consuming dry-docking operations, as well as safety risks exposed to the underwater inspectors. The use of ROVs for underwater inspection exercises is deemed a potential solution to these challenges. This report proposes specific objectives which include the design, fabrication and testing of an underwater ROV. This being the first phase of this undertaking, it is proposed that the ROV design be based on a shallow landing craft with a draught of 0.75 m and therefore a design depth of 1 m is adopted for the ROV in this report. There are three categories of design in this work, namely mechanical, propulsion and electrical design works, which are clearly outlined under the methodology chapter. At the end of the project, it is expected that a ROV with maneuverable controls along the main axes of motion as well as around an obstacle underwater will be realized.

# 6. Budget

**Table 2 Estimated Budget**

|  |  |  |  |
| --- | --- | --- | --- |
| ITEM | NUMBER OF ITEMS | COST PER ITEM  (KSH) | TOTAL COST  (KSH) |
| PVC | 4 | 545 | 2,180 |
| Battery for motors | 6 | 549 | 3,294 |
| Battery for electrical components | 1 | 549 | 549 |
| Raspberry Pi 3 | 1 | 8,000 | 8,000 |
| Arduino nano | 1 | 1,800 | 1,800 |
| Relays for motors | 6 | 150 | 900 |
| ESC30A | 6 | 550 | 3,300 |
| Camera | 1 | 1,800 | 1,800 |
| Nylon plastic | 4 | 250 | 1,000 |
| Ethernet cable | 1 | 550 | 550 |
| Acrylic Plexiglass | 1 | 2,300 | 2,300 |
| Jumper cable | 1 | 313 | 313 |
| Thrusters | 6 | 5,000 | 30,000 |
| Operator Controller | 1 | 2,000 | 2,000 |
|  | | **Total** | **57,989** |

# 7. Time plan

**Table 3 Time Plan**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | May | Jun | Jul | Aug | Sept | Oct | Nov | Dec |
| Project Proposal |  |  |  |  |  |  |  |  |
| Literature Review |  |  |  |  |  |  |  |  |
| Mechanical & Structural design |  |  |  |  |  |  |  |  |
| Propulsion design |  |  |  |  |  |  |  |  |
| Electronic design |  |  |  |  |  |  |  |  |
| Control subsystem implementation |  |  |  |  |  |  |  |  |
| Manufacture of the structural and mechanical design |  |  |  |  |  |  |  |  |
| Assembly |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |
| Final Presentation |  |  |  |  |  |  |  |  |

# 

# References

[1] M. a. C. Agency, "In-water Surveys," *Marine Guidance Note - MGN 546(M)*.

[2] B. Lin and X. Dong, "Ship hull inspection: A survey," *Ocean Engineering,* vol. 289, p. 116281, 2023/12/01/ 2023, doi: <https://doi.org/10.1016/j.oceaneng.2023.116281>.

[3] B. Lin and X. Dong, "Ship Hull Inspection: A Survey," *Ocean Engineering,* 11/05 2023.

[4] N. Lahkar and S. Gautam, "Applications of Remotely Operated Vehicles (ROVs) in Offshore Operations: A Review on Recent Developments," 09/15 2023.

[5] H. PARSONS, "An insight into the history of ROVs."

[6] B. Jacques, "Hempel launches underwater hull inspections using ROVs," S. Maritime, Ed., ed, Sept 2020.

[7] Y. Ahmed, O. Yaakob, and B. Sun, "Design of a new low cost ROV vehicle," *Jurnal Teknologi,* vol. 69, 07/20 2014, doi: 10.11113/jt.v69.3262.

[8] P. I. Macreadie *et al.*, "Eyes in the sea: unlocking the mysteries of the ocean using industrial, remotely operated vehicles (ROVs)," *Science of the Total Environment,* vol. 634, pp. 1077-1091, 2018.

[9] NOAA, "Remotely Operated Vehicle," *Ocean Engineering*.

[10] A. Wylie, "ROV Ventana completes 4500 Dives Deep Sea Dives."

[11] I. Hudson, D. Jones, and D. Wigham, "A review of the uses of work-class ROVs for the benefits of science: lessons learned from the SERPENT project," *Underwater Technology,* vol. 26, no. 3, pp. 83-88, 2005.

[12] J. Lever, G. Phetteplace, and J. Weale, "Benefits of Using Remotely Operated Vehicles to Inspect USACE Navigation Structures," p. 28, 03/01 2007.

[13] E. Allmendinger, *Submersible vehicle systems de- sign. The society of naval architects and marine engineers, Jersey City (NJ).* 1990.

[14] M. D. Patryk Zając, Adam Olejnik, "STABILITY TESTING METHOD FOR UNDERWATER ROVS," no. POLISH HYPERBARIC RESEARCH 1(74)2021, 2021.

[15] O. Aguirre-Castro *et al.*, "Design and Construction of an ROV for Underwater Exploration," *Sensors,* vol. 19, p. 5387, 12/06 2019, doi: 10.3390/s19245387.