ROVEN Sealion: Designing an Underwater Remotely Operated Vehicle for Deep Sea Energy Wells

A Major Qualifying Project Report
Submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the degree of Bachelor of Science

by

Brad Bailey
Mechanical Engineering

Gimo Barrera
Mechanical Engineering

Shannon Ketcham
Electrical & Computer Engineering

Greg Bailey
Mechanical Engineering

Aren Johansen
Mechanical Engineering

Kevin Malhorn
Robotics Engineering

Date: April 25, 2013

Approved:

Professor Mustapha S. Fofana, Major Advisor
Mechanical and Robotics Engineering Departments

Professor Fred J. Looft, Co-Advisor
Electrical & Computer Engineering Department
Abstract

Extracting energy from offshore oil wells presents enormous engineering challenges in ensuring that the energy is harvested safely and efficiently. Recent events have shown that the failure to monitor and control the various systems used during production can lead to environmental disasters. Blowout preventers are large stacks of hydraulically operated rams located on the sea floor and are the last line of defense against oil spills. It is nearly impossible for humans to work on such structures without special equipment. Remotely operated vehicles (ROVs) are a common method of maintaining the systems on the blowout preventers and allow humans to safely monitor the structure from the oil platform or a nearby boat. Without adequate visual and environmental feedback, operators of these machines cannot perform their task safely. The purpose of this project was to design and build an ROV capable of monitoring the condition of underwater structures. After studying current industry standards for deep sea observational ROVs, the team designed a prototype which can perform similar tasks in shallow waters. This prototype ROV is able to provide feedback from several sensors, including a video camera, a temperature sensor, pressure sensors, and a compass. With regard to future work, the ROV was designed as an open frame with adjustable thruster and battery locations. The lightweight design of the ROV also allows for the addition of a payload, which could include robotic manipulators and additional sensors and thrusters.
Table of Contents

Abstract............................................................................................................................................ i

Table of Contents ............................................................................................................................ ii

List of Figures ................................................................................................................................... v

List of Tables ................................................................................................................................... viii

Acknowledgements ........................................................................................................................ ix

CHAPTER 1: Introduction ..................................................................................................................... 1

CHAPTER 2: Background Research .................................................................................................. 2

2.1 Introduction................................................................................................................................ 2

2.2 Structural Health Monitoring .................................................................................................. 2

2.2.1 Offshore Oil Platforms ........................................................................................................ 3

2.2.2 The Deepwater Horizon ...................................................................................................... 3

2.3 Underwater Remotely Operated Vehicles ............................................................................ 7

2.3.1 ROV Classes ....................................................................................................................... 7

2.3.2 ROV Usage in Structural Health Monitoring .................................................................. 10

2.3.3 Inspection and Monitoring Using ROVs ....................................................................... 11

2.4 ROV System ............................................................................................................................ 14

2.4.1 ROV Components Overview ......................................................................................... 15

2.4.1.1 Umbilical, TMS, and Tether ..................................................................................... 15

2.4.1.2 Subsea Cage ............................................................................................................... 17

2.4.1.3 Propulsion (Thrusters) ............................................................................................ 17

2.4.1.4 Cameras .................................................................................................................... 18

2.4.1.5 Manipulators ........................................................................................................... 18

2.4.2 Interface Requirements .................................................................................................... 19

2.5 Case Studies ............................................................................................................................ 20

2.5.1 Spectrum ROV .................................................................................................................. 20

2.5.2 Sea Maxx Satellite ROV ................................................................................................. 22

2.5.3 SeaBotix LBV300-5 ....................................................................................................... 23

2.5.4 OpenROV ....................................................................................................................... 25

2.6 ROV Design ........................................................................................................................... 27

2.6.1 Thrusters .......................................................................................................................... 27

2.6.2 Buoyancy ......................................................................................................................... 30

2.6.3 Materials .......................................................................................................................... 31

2.6.3.1 Frame ...................................................................................................................... 31
List of Figures

Figure 1: Extreme Loads from Storms for Marine Environment .................................................. 2
Figure 2: Types of Oil Platforms .................................................................................................. 3
Figure 3: Shear Ram (left) and Annular Preventer (right) ............................................................ 4
Figure 4: BOP stack and housing ............................................................................................... 5
Figure 5: Broken Riser Pipe ........................................................................................................ 6
Figure 6: Class 1 ROV ................................................................................................................ 8
Figure 7: Class 2 ROV ................................................................................................................ 9
Figure 8: Class 3 ROV ................................................................................................................ 9
Figure 9: ROV Installation Assistance ....................................................................................... 10
Figure 10: Seabed Environment ................................................................................................. 11
Figure 11: (a) Cleaning Brush (b) Pressure Washer ................................................................. 12
Figure 12: Cathodic protection (CP) probe ............................................................................... 13
Figure 13: ROV system .............................................................................................................. 14
Figure 14: Umbilical cross-section ........................................................................................... 16
Figure 15: Tether cross-section ............................................................................................... 16
Figure 16: Tether Management System (TMS) ...................................................................... 17
Figure 17: ROV cage .................................................................................................................. 17
Figure 18: Spectrum ROV ....................................................................................................... 21
Figure 19: Spectrum ROV and work class ROV .................................................................. 22
Figure 20: Sea Maxx Satellite ROV ...................................................................................... 23
Figure 21: SeaBotix LBV300-5 ROV .................................................................................... 24
Figure 22: LBV300-5 controller ............................................................................................ 25
Figure 23: OpenROV ................................................................................................................ 25
Figure 24: OpenROV parts ..................................................................................................... 26
Figure 25: Possible ROV thruster arrangement .................................................................... 27
Figure 26: Vehicle degrees of freedom .................................................................................... 28
Figure 27: Commutator and brushes ...................................................................................... 29
Figure 28: Complete thruster diagram .................................................................................... 29
Figure 29: A PVC frame (left) and a polypropylene frame (right) ............................................ 32
Figure 30: Closed-cell and open-cell foams .......................................................................... 34
Figure 31: Floatation components ........................................................................................... 35
Figure 32: Backscatter ............................................................................................................. 36
Figure 33: Lighting layout ........................................................................................................ 36
Figure 34: Rotundity CREE MR16 LED .................................................................................. 37
Figure 35: DS18B20 thermistor ............................................................................................. 40
Figure 36: The Mini Gyro ....................................................................................................... 42
Figure 37: A USBL positioning system .................................................................................... 42
Figure 38: Tilt sensor ............................................................................................................... 43
Figure 39: Cross-sectional view of ROV tether ..................................................................... 44
Figure 40: Dead reckoning ..................................................................................................... 46
Figure 41: Board camera ......................................................................................................... 47
Figure 42: GoPro camera ......................................................................................................... 47
Figure 43: Webcam .................................................................................................................. 48
Figure 44: System diagram ...................................................................................................... 49
Figure 45: Final design of the ROVEN Sealion ..................................................................... 51
Figure 92: Pulse-width modulated signals .............................................................. 112
Figure 93: Program flow for the ROVEN Sealion .................................................. 114
Figure 94: Controller input configuration .............................................................. 115
Figure 95: Processing code used to map user inputs to bytes ............................... 117
Figure 96: The graphical user interface for the ROVEN Sealion ............................ 118
Figure 97: ROV descending with turbulent region attached ................................. 120
Figure 98: ROV descending as turbulent region becomes detached ................... 121
List of Tables

Table 1: ROV Details ...................................................................................................................... 8
Table 2: Possible materials for an ROV frame ............................................................................. 33
Table 3: Material Properties for some polymer foams ................................................................. 35
Table 4: Common sensor types for ROVs .................................................................................... 38
Table 5: Acoustic positioning maximum signal range by frequency ........................................... 45
Table 6: Comparison between deep-sea ROV and this project's design ....................................... 52
Table 7: Comparison of motors .................................................................................................... 56
Table 8: Current draw for each motor speed ................................................................................. 59
Table 9: Average thrust for each motor speed .............................................................................. 60
Table 10: Material comparison for the ROV frame ...................................................................... 64
Table 11: Buoyancy calculations .................................................................................................. 68
Table 12: Drag forces in each direction ......................................................................................... 85
Table 13: Drag coefficient and frontal area of each movement direction .................................... 89
Table 14: Specifications for the Logitech C210 webcam ............................................................. 97
Table 15: Specifications for the GE NPC-1220 pressure sensor ................................................ 100
Table 16: Specifications for the Honeywell HMC6352 magnetic compass module ................. 102
Table 17: Specifications for the DS18B20 temperature sensor .................................................. 104
Table 18: AWG wire properties .................................................................................................. 106
Table 19: Specifications for the UB12180 battery ..................................................................... 109
Table 20: Specifications for the PRB2314 electronic speed controller ...................................... 111
Table 21: Pulse width settings for the PRB2314 electronic speed controller ............................. 113
Acknowledgements

This project group would like to thank Worcester Polytechnic Institute for supplying testing facilities, providing storage space, and providing reimbursement for the budget that was spent on the project.

Additional thanks go to our advisors Professor Mustapha S. Fofana and Professor Fred J. Looft for their support and guidance.
CHAPTER 1: Introduction

Maintenance on oil platforms is of utmost importance due to the potential for widespread disaster. Offshore oil platforms are difficult to maintain, especially in deeper waters. There is a need for structural health monitoring and damage detection on these underwater structures in order to prevent oil spills. British Petroleum’s (BP) oil platform, Deepwater Horizon, encountered a malfunction in its blowout preventer, resulting in the 2010 oil spill in the Gulf of Mexico. Events like these bring more attention to the safety and maintenance procedures used on offshore oil platforms. In response to oil spills, submersible remotely operated vehicles (ROVs) are frequently used to monitor the spill at the source and manually engage the blowout preventer (BOP), which is a device used to prevent oil spills by stopping the flow of oil in an emergency situation. The use of ROVs has become more prevalent in deep-sea applications because of their versatility and ability to operate in harsh conditions.

This Major Qualifying Project (MQP) focused on designing and constructing the ROVEN Sealion, an observational remotely operated underwater vehicle that mimics the abilities of professional-grade deep-sea observational ROVs. The ROVEN Sealion was designed to be capable of basic structural health monitoring and damage detection, while maintaining a budget suited to a college project. Many deep-sea observational ROVs monitor the outer parts of blowout preventers on offshore oil platforms. Instead, the ROVEN Sealion is a shallow-water observational ROV with some basic payload capability designed for use in the oil industry.

Chapter 2 contains information about structural health monitoring, blowout preventer functionality, and overall ROV design concepts. All of the research needed about ROVs for this project can be found in this chapter. Chapter 3 focuses on the design, methodology, construction, and troubleshooting of the ROVEN Sealion. This chapter includes detailed analyses of the forces and stresses the ROV will experience. This chapter also goes into detail on flow analysis, and experiments that were performed during the design process. Component selection, programming procedures, and electrical systems of the design also are found in this chapter. The end of Chapter 3 discusses the results and the troubleshooting for this project. Chapter 4 discusses the constraints and limitations of the project. Chapter 4 also serves as a conclusion for the final report, emphasizing the impact of the ROVEN Sealion and outlining possible future work on the project.
CHAPTER 2: Background Research

2.1 Introduction

To successfully build an ROV, there are many different elements that must be considered. While it is always best to consider each individual component, it is also helpful to review previous work in the same field. In this section, previous work in the field of ROVs is discussed. In addition, each individual component and consideration is reviewed in depth. This chapter outlines the foundation of knowledge upon which the ROVEN Sealion is built.

2.2 Structural Health Monitoring

Structural Health Monitoring (SHM) is the process of implementing damage detection and maintenance on an engineering structure [1]. SHM increases human safety, environmental safety, and reduces maintenance cost. Practical implementations of engineering techniques meet several difficulties in marine conditions. Offshore structures are frequently exposed to harsh marine environments. As seen in Figure 1, storms can create heavy loads on offshore structures, including long-term cyclic loadings, which occur from continuously acting sea waves, and short-term extreme loads that occur from severe storms, seaquakes, or collisions. Animal damages to the structures are common, and due to the long exposure to sea water, structures suffer from fast corrosion, erosion, and scour processes. This environment creates damages and accelerates existing ones as well [2]. The data interpretation and processing techniques in SHM are essential to properly collect information and take proper action [3].

Figure 1: Extreme Loads from Storms for Marine Environment
2.2.1 Offshore Oil Platforms

Structural Health Monitoring is most commonly seen in offshore oil platforms. These are structures with facilities capable of drilling in order to extract oil or natural gases. These structures also temporarily store product until it is brought to shore for refining and marketing. The most common of these platforms are the ones that are fixed to the ocean floor (fixed platforms), and semi-submersible floating platforms, otherwise known as artificial islands. Figure 2 shows the different types of offshore oil platforms, all of which use blowout preventers (BOP). Starting from the left: 1 & 2) conventional fixed platforms; 3) compliant tower; 4 & 5) vertically moored tension leg and mini-tension leg platform; 6) Spar; 7 & 8) Semi-Submersibles; 9) Floating production, storage, and offloading facility; 10) sub-sea completion and tie-back to host facility [4].

![Figure 2: Types of Oil Platforms](image)

2.2.2 The Deepwater Horizon

On April 20th, 2010, the offshore drilling platform, the Deepwater Horizon, exploded in the Gulf of Mexico. This event caused the deaths of 11 men as well as the release of millions of gallons of oil into the Gulf Stream. While the explosion itself was the result of a number of problems, the release of the oil was the result of a failure of one system, the blowout preventer. A blowout preventer is placed on the top of an offshore oil well as a method to prevent the escape of oil into the water by using rams to cut and seal the pipe connecting the platform to the well. The blowout preventer plays a crucial role in the impacts felt by an oil drilling disaster.
The release of such a large amount of oil into the ocean has a tremendous impact on the surrounding ecosystem, the coastal communities and their economies, as well as the offshore energy business.

Because of the BOP’s importance in the event of a disaster, it is imperative that the BOP can be relied on to function in a number of different situations. The BOP already has an extensive array of redundant systems to ensure the completion of its task. In certain situations, even those systems are not enough.

A blowout preventer is a large device weighing as much as 400 tons and measuring 50-60 ft tall. The BOP is placed directly on top of the surface of the well and surrounds the pipe which delivers oil to the platform above. Should a blowout ever occur, the BOP is meant to stop the flow of oil as soon as possible. It does this by completely severing or crushing the oil pipe via a system of blind shear rams and annular preventers, seen in Figure 3. The rams are hydraulically operated and consist of a metal plate affixed to a piston. When activated, the piston is pushed via hydraulics and the plate slices through the pipe and seals off the oil from the upper sections of the BOP.

![Figure 3: Shear Ram (left) and Annular Preventer (right)](image)

There are usually multiple shear rams lined up along the length of the BOP in case one fails to completely seal off the pipe. The system of multiple rams and annular preventers makes up the majority of the mass of the BOP and can be referred to as the BOP stack. The BOP stack used on the Macondo Well, the oil well on which the *Deepwater Horizon* was stationed, used a series of 5 blind shear rams and 2 annular preventers. Figure 4 shows a BOP stack; the shear rams are in blue and the annular preventers are in red.
The control of a blowout preventer comes from the oil platform. The first control system is an automatic kick detection system which activates the rams when control of the oil flow rate is lost. The second control system, should the kick detection fail, is activated by the crew on the platform and is called the Emergency Disconnect System. Both of these systems rely on cables to send the signal from the platform to the blowout preventer. If these cables are compromised, a third system, called “the Deadman System” or Automatic Mode Function (AMF), is activated as soon as communication to the platform or the hydraulics on the BOP is lost and is generally considered to be a failsafe. In the case that this does not succeed in activating the rams, the remaining option is to activate them manually using the controls on the BOP itself as it sits on the bottom of the ocean. The only way of doing this is by an underwater remotely operated vehicle (ROV). The ROV uses onboard manipulators to turn valves which close the BOP [5].

During the Deepwater Horizon disaster, each of these methods of closing the BOP failed, including the ROV intervention system. When the oil platform exploded, the connection between the BOP and the control room was severed. This caused the kick detection and the emergency-disconnect system to fail. At this point, the AMF should have activated the rams. Upon later inspection of the BOP, however, one of the hydraulic operators was found to have a defective solenoid and did not activate. Some of the other rams had depleted batteries and could not provide enough power to finish the AMF sequence. The Deepwater Horizon did not have a subsea electronic module (SEM) to monitor the charge of the batteries even though this was an option when the BOP was being installed [6].
The first ROVs reached the BOP at about 6pm on April 21\textsuperscript{st} and immediately attempted to close the BOP manual using the ROV intervention panel. Until this time, oil was still being released and fueling the fires from the explosion. In order to manually close the BOP, an ROV has to be able to cut a poppet valve. The ROV was not able to do so. An attempt to activate the blind shear ram by hot stabbing was also unsuccessful. Before the BOP was lowered into the ocean, Transocean, the company managing the \textit{Deepwater Horizon}, switched the variable bore ram on the BOP stack to a test ram. This was not known until after the first hot stabbing attempt. Figure 5 shows the oil escaping from the broken riser pipe.

![Figure 5: Broken Riser Pipe](image)

“From the outset of the ROV interventions, Transocean could not produce the redline (as built) drawings to know how the BOP components worked as were configured. Throughout the intervention, drawings had to be updated with what was observed from ROV footage. According to Transocean, they kept the redline drawings on the \textit{Deepwater Horizon} and the documents went down with the rig.” [6]

This means that the ability of the ROVs to obtain clear images and video of the BOP was extremely important. All of the BOP plans had to be updated based on the ROV’s feedback. Initial footage from the ROVs gave the operators the impression that the BOP was successful in cutting off the flow of oil. The ROVs did not obtain footage of the broken riser pipe until the
23rd when an ROV found oil flowing from the end of the pipe. The likelihood that the rams could close decreased each day due to the corrosion that was caused by the oil and sand moving through the BOP. In May, the BOP was still not closed and oil had been flowing into the gulf the entire time. Around this point, Transocean began to focus on capping the well instead of sealing off the BOP. Capping the oil well also required the use of ROVs which had to cut the broken riser pipe and guide the cap onto the top of the well [5].

### 2.3 Underwater Remotely Operated Vehicles

In this paper, the term remotely operated vehicle (ROV) refers to tethered underwater robots in the oil and gas industry. ROVs are unmanned, and are remotely operated by personnel aboard a vessel, such as an oil platform, drillship, or exploration ship.

ROVs are popular in the oil and gas industry for performing tasks such as platform and pipeline inspection. For these tasks, an ROV is equipped with various instruments to monitor corrosion and fouling, to locate structural cracks, to estimate biological fouling, to check for leaks, to determine overall health, and to ensure acceptable installations. Before installation of an oil platform, ROVs are employed to take topographical surveys through both visual and acoustic means. ROVs can also accomplish tasks as simple as debris removal or more complex tasks such as drilling and construction support. With power tools and manipulators, ROVs are capable of performing the very complex tasks involved in installing, repairing, and inspecting blowout preventers [7].

#### 2.3.1 ROV Classes

There are multiple classes of ROVs which are based on their capabilities; some of the most popular types are detailed in Table 1. There are actually two other ROV classes in addition to those shown in Table 1. All five ROV classes are described in this section.
<table>
<thead>
<tr>
<th>Class</th>
<th>Purpose</th>
<th>Weight (lbs)</th>
<th>Size (sq ft)</th>
<th>Uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Observational</td>
<td>30 - 250</td>
<td>1 - 60</td>
<td>Visual inspection</td>
</tr>
<tr>
<td>2</td>
<td>Obs. with Payload</td>
<td>60 - 300</td>
<td>6 - 60</td>
<td>Visual inspection, Environmental sensing, Minor manipulation</td>
</tr>
<tr>
<td>3</td>
<td>Work Class</td>
<td>600 - 1000+</td>
<td>120 - 500</td>
<td>Visual inspection, Environmental sensing, Manipulation, Installation, Repair</td>
</tr>
</tbody>
</table>

Class 1: Observational ROVs

Figure 6 shows a Class 1 observational ROV. These ROVs are small in size and only include camera, lights, and sonar. Their main purpose is observation only. Some are able to handle one additional sensor, such as a video camera or cathodic protection probe.

![Figure 6: Class 1 ROV](image)

Class 2: Observational ROVs with Payload Option

Figure 7 shows a Class 2 observational ROV with payload option. These ROVs typically have two simultaneously viewable camera and sonar feeds, and are capable of handling several sensors. These ROVs may also have basic manipulative capabilities. They should be able to operate without loss of original function while carrying two additional sensors or manipulators.
Class 3: Work Class Vehicles

Figure 8 shows a Class 3 work class vehicle. These ROVs commonly have multiplexing capabilities that allows for additional sensors and tools to operate without being connected to the umbilical system (discussed in section 2.4.1.1 Umbilical, TMS, and Tether). These ROVs are larger and more powerful than observational ROVs (Class 1 and Class 2). Due to this large size, they are capable of carrying additional sensors and manipulators.

Class 4: Towed and Bottom-Crawling Vehicles

Class 4 vehicles are pulled through the water by a surface craft or winch. Because of this, they have limited maneuverability. These are typically large and heavy, and are designed for specific tasks such as burying cables. These ROVs use a wheel or track system to move across the seafloor. Some of these are able to travel limited distances off of the seafloor.
Class 5: Prototype or Development Vehicles

Special-purpose vehicles that do not fit into any of the other classes are assigned to Class 5. This class also includes autonomous underwater vehicles (AUVs). Prototype ROVs that are still being developed are also assigned to this Class [8].

2.3.2 ROV Usage in Structural Health Monitoring

In the installation of offshore structures, the first issues to appear are those related to finding the position and location for the base frame, BOP, drill hole, and pipe lines, as seen in Figure 9. The installation process can be divided into two parts: subsea equipment installation, and pipeline installation. One of the methods used to familiarize the engineers with the environment is through seabed mapping. This process can be accomplished in multiple ways, one of which is precise bathymetry, in which seabed information is gathered by a reference system with differential pressure sensors and acoustic data transmission, and then transmitted to a nearby ROV. ROVs are also capable of carrying multibeam echosounders (MBE), side-scan sonar (SSS), and sub-bottom profilers (SBP), which can map the seabed and assess its quality for offshore foundation installation [9].

![Figure 9: ROV Installation Assistance](image)

When setting up the frame and the drill, ROV structure position assistance may be required. ROVs performing this task focus mainly on pulling and pushing in order to properly place the structure into its required location. This can also include guiding wire deployment, recovery, and cutting [9]. During installation, ROVs are used for observation and verification of
proper component placement and for engagement and release of guide wires and hooks. The ROV may also be used to retrieve components that are only needed during installation. It is possible for an ROV to independently install smaller parts and components. Figure 10 shows the environment of the seabed where the ROVs work.

![Figure 10: Seabed Environment](image)

### 2.3.3 Inspection and Monitoring Using ROVs

Inspection is needed on a routine basis; as explained previously, under these harsh conditions, structures are expected to deteriorate from flowline vibrations, internal erosion, corrosion, and drastic loads. There are multiple techniques for inspection. The most simplistic one is general visual inspection, which includes cathodic protection measurements and marine growth measurements.

Close visual inspection, used in order to find cracks and deformation, requires physical cleaning. This cleaning allows for cathodic protection (CP) measurements, which test the quality of the corrosion-resistant coating on the structure. Magnetic particle inspection (MPI), eddy-current testing, and alternating current field measurement (ACFM) are methods which test for cracks or deformations within the structure [9].
These tests are known as nondestructive testing (NDT) methods. The visual inspection is in order to detect cracks and deformations in the structure. The ROVs capable of performing this task are equipped with lights and high-definition (HD) cameras. Monitoring done by observational class ROVs is used in drilling assistance, frame construction by checking the orientation, and monitoring the structure as it is lowered. Visual observation is occasionally accompanied by scanning sonar for supplemental acoustic observation [9].

Different sensors allow ROVs to monitor flow temperatures as well as pressures both in the environment and in the structure [9]. Other tools include brushing tools or high-pressure wet jets, as seen in Figure 11. A clean surface not only allows for visual inspection, but also for CP and MPI testing. Structures are also cleaned so they are not deteriorated by the marine environment. When installing a structure’s cement, cleaning is necessary for placing a base.

![Figure 11: (a) Cleaning Brush (b) Pressure Washer](image)

CP potential measurements are accomplished through the use of a CP probe. These probes are usually placed at the end of a manipulator and attached to a work class ROV, since it only needs to touch the surface to run tests. The CP probe checks the quality of the corrosion protection on the surface of the inspected area. This type of protection used on piping and metal frames, among other components. Figure 12 shows an example of a CP probe.
A detailed inspection checks for cracks underneath the surface, wall thickness measurements, and flooded member detection. These are done through MPI tests and eddy-current tests. MPI tests are usually conducted by divers in shallower areas of the structures. For deeper environments, working-class ROVs check the pipes and frames using similar tests. MPIs work on ferroelectric materials where the part in question is magnetized and then checked for magnetic flux leaks, which would point out deformations or cracks underneath the surface. Usually small metal particles are released and travel to the section with flux leakage, showing where there might be deformations in the material [10]. CP and MPI measurements are both nondestructive testing processes.

Visual inspection and monitoring serve to identify necessary maintenance and repairs. Many of these have to be done by divers and at high depths the item that needs repairs needs to be brought up to the surface. Some maintenance tasks can be replicated by a work class ROV. Maintenance on an underwater structure includes repair or replacement of modules that are subject to wear. This module is taken out and brought to the surface, and replaced with a functional one [9].
2.4 ROV System

Figure 13 shows a typical arrangement for an ROV system. In this figure, the surface power supply unit is a motor generator that provides power with 200 kVA capacity. These generators are usually self-contained diesel-powered units, or a motor/alternator which depends on an external power source. The control unit, usually referred to as the control van/container, is used by the operator during ROV subsea operations. The entire operation interface, including cage and ROV control, takes place here. These control rooms are usually equipped with a climate control system, lighting with intensity adjustments, electrical outlets, fire alarms, and fire extinguishers [8].

The surface handling involves many subcomponents. The first being the main lift winch, which is made up of a hydraulically driven winch drum, a hydraulically driven level wind assembly, a transformer box, a stationary and rotating fiber optic cable, and other small components. The A-frame launch skid allows for the cage and ROV to be positioned on the side
of the vessel for deployment and retrieval, using hydraulic cylinders to swing outboard and inboard. Lastly, the hydraulic power unit (HPU) provides the power for the A-frame and lift winch. The HPU is powered by electric motors which directly connect to the vessel’s power supply. The HPU has its own interface [8].

The Launch and Recovery System (LARS) is the main crane used to lower and return the cage/ROV to the support vessel or send it to the ocean floor.

Overall, the ROV system can be divided into the following sections: power source, control room, workover room, deck handling, and umbilical (to the cage and the ROV) [9].

2.4.1 ROV Components Overview

Section 2.4 explains common components used in industrial class ROVs. These components include the umbilical, tether, tether management system, subsea cage, thrusters, cameras, and manipulators. This section will also discuss the interface requirements for these components.

2.4.1.1 Umbilical, TMS, and Tether

The umbilical is located between the support vessel and the ROV’s Tether Management System (TMS). It is used to transport hydraulic and electronic power from the vessel to the ROV. The umbilical also transfers information gathered by the ROV to the surface and eventually into the control room. One of the main factors when designing an umbilical is the diameter. The aim is to keep the diameter small in order to reduce the drag force due to the waves and current. Having a small diameter and low weight also lowers the force required when returning the ROV back to the surface. Umbilicals are usually negatively buoyant, so that they do not float unpredictably. In some instances, the umbilical has different buoyancies throughout its structure in order to avoid entanglement when running operations close to the surface [9]. Typically, a steel shielding is constructed around a central cable, which contains smaller power and communication cables, as shown in Figure 14 [8]. Unlike an autonomous underwater vehicle (AUV), ROVs have a tether [9].
The tether, similar to the umbilical, serves as a medium through which signals, power, and information are transmitted to and from the ROV. The tether is from the cage to the ROV (Figure 15), while the umbilical is from the cage to the surface (Figure 14). Tethers are usually neutrally buoyant [8]. The length of the tether varies from 600 to 2000 ft, depending on the ROV’s intended use.

The tether management system (TMS) sits on top of the ROV cage (Figure 16). This ensures there is minimum drag on the ocean floor. It can also be used for storing samples collected by the ROV. The TMS ensures the tether does not get entangled, and it retracts the tether when needed.
2.4.1.2 Subsea Cage

The cage, as seen in Figure 17, serves as a support vessel for the ROV during launch and recovery. The cage serves as a carrier for tooling and sample storage. It consists of a metal structure, which protects the ROV during launch and recovery [8].

2.4.1.3 Propulsion (Thrusters)

The propulsion system of an ROV consists of a power source, a controller for electronic motor or a servo-valve pack for a hydraulic motor, and thrusters. The main component of the propulsion system is the thrusters. The thrusters adjust the vehicle conditions and also allow the operator to effectively maneuver and control the position of the vehicle. This includes propelling
the ROV from the TMS to the work site, or vice versa. The main factors when taking into account which thruster to use are power, efficiency, pressure, flow, weight, size, and forward/reverse characteristics. The thrust needs to be able to counter the forces acting on the ROV, including hydrodynamic and workload forces. Thrusters range from electrically powered, mainly used for observational/small class ROVs, to hydraulically powered, used in work class ROVs [9].

2.4.1.4 Cameras

Cameras are usually used for navigation, inspection, and monitoring purposes. Cameras have image sensors which could include the following:

- Low-light silicon intensified targets (SITs)
- Charge-coupled devices (CCDs)
- Hole accumulation diode (HAD) devices used for HD images

Cameras also utilize light emitting diode (LED) lights to provide illumination for close-up inspection. This eliminates the need for separate lighting. The camera transmits video signals through the tether and umbilical to a video capture device on the surface. [9]

2.4.1.5 Manipulators

A manipulator in this section refers to a robotic arm, which is customized to carry out various subsea functions [8]. The two most common manipulators used in ROVs are position stabilization (normally with a five-function arm) and one for intervention tasks (seven-function arm). Depending on the class of ROV, the manipulator will vary in size, load rating, reach, functionality, and controllability. Work class ROVs have the greatest need for manipulators, since they have the best capacity to carry and properly control one. The manipulator hand usually consists of a two- to three-finger mechanism that grasps handles, objects, and structural members. The hand usually includes grippers to perform tasks or to help stabilize the ROV [9].

A linear valve override tool (LVOT), commonly used on the BOP, allows an ROV to manually operate a particular type of subsea valve. During installation it can help remotely operate subsea manifolds and trees to test contingency plans in case there is a control failure on the surface.
2.4.2 Interface Requirements

ROV interfaces, such as those listed below, are produced for ROVs to help in inspection, maintenance, and repair (IMR) including installation assistance and drilling assistance [9].

- Hydraulic work package and docking frame for the ROV
- Hydraulic connector, valve override tool, and adaptor for the ROV manipulator
- Manifold valves fitted with a guidance system
- Modules and tools equipped with a local control panel

Stabilization is achieved through:

- Landing on a working platform that is connected to the subsea structure
- Suction cups, usually at end of manipulators (arm)
- Grasping
- Docking; used with single or twin docking tool deployment unit (TDU)

Interfaces have the following specifications for stabilization purposes:

- Working platforms should be free from obstruction.
- Subsea structure's flat surfaces should be broadly adjacent to the task area for suction cup attachment for stabilization purposes.
- The grasping interfaces should withstand a minimum force of 2.2 kN and a gripping force of 2.2 kN from any direction.
- Docking should have fail-safe release and overload limitation features [9].

Handle interfaces are located between tools or hot stabs and the ROV manipulators or ROV mounted tools. A hot stab is a tool that allows fluid transfer from one point to another with minimal leakage into the subsea environment [8]. This interface should follow the following specifications:

- The stem of handles should resist maximum operational forces, both linear and rotational forces.
- Out-of-line forces generated by operator in linear applications should be considered by the compliancy between the handle and the attachment to the tool.
- Handle should be marked to display direction of movement of the handles. (reduces probability of damage) [9].
The rotary torque tool is a bi-directional tool that fits into a subsea bucket with a receiving valve system. The main purpose of this tool is to apply exact torque values to the valve; it also counts how many times the valve has been turned. The torque tool allows verifiable precision actuation of subsea valves, thus preventing damages to valves [8].

2.5 Case Studies

Underwater remotely operated vehicles are a good way to deal with and interact with structures that sit on the ocean floor, but there are several ways to design ROVs. As discussed earlier, there are several different classes of ROVs that differ by the role for which the ROV was made. This project focuses on Class 2 ROVs (observation with payload), but even amongst this class of ROV, there are a variety of ways in which ROVs can be designed. Case studies were performed to look for some of the common practices that go into making an observational ROV.

Oceaneering is the leading company involved in the development and use of deep sea ROVs. Oceaneering focuses mainly on building products to aid offshore oil platforms by helping in deep water environments [11]. Some of the tasks that an Oceaneering ROV would be built to fulfill include assessing the integrity of undersea structures, inspecting and running non-destructive tests on the structures, and installing devices onto structures [11]. A good portion of the ROVs that Oceaneering specializes in are work class submarines, but they develop observational ROVs as well.

2.5.1 Spectrum ROV

One such observational ROV made by Oceaneering includes the Spectrum ROV, as seen in Figure 18. The Spectrum is a medium-sized ROV; it is not as big as work class ROVs, but is larger than most other observational ROVs, being 55.1x35.4x33.4 inches [12]. The ROV weighs 639 lbs and can reach depths of about 3000 meters [12].
The Spectrum is unique in that it is a combination of work class ROVs and observational ROVs. This sort of ROV exhibits the characteristics of a Class 2 ROV, but technically, the Spectrum is classified as a work class ROV. If the Spectrum is a work class ROV, then it is a very light work class ROV. The main objective of the Spectrum ROV is to aid larger work class ROVs and fulfill tasks that would require a smaller-bodied work class ROV. The Spectrum can aid oil or gas drilling, construction, and production activities [12]. Figure 19 shows how the Spectrum can be used in tandem with other work class ROVs. The Oceaneering Power and Control system (known as OPAC) is made specifically to cater to its main objectives [12]. The control system is also interchangeable with all of Oceaneering’s other work class ROVs. This is beneficial since it allows for fewer problems and better synergy between all of the ROVs.
The Spectrum can be lowered into the ocean by a cage-deployment system and is connected to the cage by a tether. The cage is connected to the ROV control station by an umbilical cord. The Spectrum can be fitted with a skid-deployed five-function manipulator which can be used to operate some lightweight tools including AX/VX ring tools, trash pumps, CP probes, and light survey sensors [12]. With the ability to use small tools and the ability to observe its work zone, the Spectrum highlights many important characteristics of a Class 2 ROV.

2.5.2 Sea Maxx Satellite ROV

The Sea Maxx Satellite, seen in Figure 20, is another Oceaneering ROV, and meets observational class specifications more closely than the Spectrum. The Sea Maxx Satellite ROV was actually built by a subsidiary of Oceaneering called Deep Sea Systems International Incorporated (DSSI), which is more focused on designing products to aid in deep sea exploration [13]. The Sea Maxx Satellite ROV, or SAT-ROV for short, works in tandem with other work class ROVs from Oceaneering.
The SAT-ROV is much smaller than other Oceaneering ROVs with its dimensions of 30x24x18 inches [13]. It weighs 230 lbs and can dive to depths of up to 4400 meters. The SAT-ROV is deployed from a cage, a common method among Oceaneering ROVs, but the SAT-ROV actually has its own separate housing that sits beneath the cage from which it is deployed. It is of interest to note that the SAT-ROV is positively buoyant in water, while most other observational ROVs tend to be neutrally buoyant. The SAT-ROV has four thrusters which include two horizontal thrusters, one vertical, and one lateral. The camera that is used is a wide angle HDTV camera module. Also equipped to the ROV are two variable intensity 40 watt LED lights and a 450 foot tether [13]. The SAT-ROV is mainly used to help observe when other ROVs are working, and can inspect areas that other work class ROVs cannot reach due to its small size.

2.5.3 SeaBotix LBV300-5

SeaBotix is another company that is devoted to making ROVs that vary in abilities, but focuses more on smaller ROV systems. Among the many interesting ROVs that SeaBotix makes, the LBV300-5, as seen in Figure 21, was chosen as an ROV to analyze.
The LBV300-5 is a small lightweight ROV that is capable of performing tasks with its arm manipulator which exemplifies the characteristics of a Class 2 ROV. The LBV300-5 is 20.5x17.5x10.2 inches. It is 28.7 lbs and can reach a depth of 300 meters [14]. The main reason the LBV300-5 was selected for examination was because of its unusually high number of thrusters: five as opposed to the more common four or less. According to SeaBotix, the five thrusters give the ROV more stability and versatility [14]. There are two vertical thrusters, two forward thrusters, and one lateral thruster. The vertical thrusters give the ROV its maneuverability. The ROV includes a three jaw grabber which allows the ROV to pick up small and large objects in addition to just observing [15]. It can only do very simple tasks, but if the grabber was more complicated, the control system and power system would become more complex, and the size of the ROV would most likely increase. The control system, as seen in Figure 22, is intuitive and ergonomic [14]. The video system includes a wide range color camera that is able to rotate about an axis to get better viewing angles.
2.5.4 OpenROV

OpenROV is a small company that specializes in one ROV of the same name, which can be seen in Figure 23. OpenROV is an open-source ROV that is available for public use and is designed to be a "do it yourself" ROV that can be made by its customers. OpenROV specializes in exploration and education.

The most interesting piece of information about the OpenROV is that it is very small. It may be the smallest observational ROV on the market. It is purely observational, primarily because other features and functionality would add to the size and weight of the ROV. The
dimensions are 11.8x7.8x5.9 inches and it weighs only 5.5 lbs [16]. The ROV is neutrally buoyant and is powered by three thrusters: two horizontal and one vertical thruster. OpenROV can only reach a depth of 100 meters, but the main objective of this ROV is to explore small areas such as underwater caves [16]. OpenROV is powered by eight on-board C batteries that fit in the two cylinders on the bottom of the ROV. The weight of the batteries counteracts the thrusters and acts as a ballast [16]. The video camera provides live video from a webcam. The tether can connect to an Ethernet cable which can then send the live video to any computer. The simple, low-cost design of the OpenROV is something this project hopes to emulate.

Figure 24: OpenROV parts
2.6 ROV Design

Section 2.6 explains the basic components and design of an industrial ROV. This section will discuss in depth the materials and components used, the function of each of these components, and how each interacts with the environment and the ROV.

2.6.1 Thrusters

The thrust and propulsion are arguably the most important aspects to consider when making an ROV. The thrust determines the overall shape and the basic functions of other components and accounts for most of the power consumption and weight. There are three main types of thrusters: electrical, hydraulic, and ducted jet propulsion [7]. These types depend on the size of the ROV and the general purpose that it needs to fulfill. The main purpose of a thruster for an ROV is to have a high thrust to drag ratio so that there is a driving force to allow the ROV to move underwater [7]. Generally, an ROV uses two or more thrusters to propel the ROV in a way that is intuitive for its operator. The number of thrusters used depends on how many degrees of freedom for which the ROV is being designed [7]. A degree of freedom determines how an object can move through space. There are six degrees of freedom which include three rotational degrees of freedom about an axis and three translational degrees of freedom. Some examples of possible configurations for thruster placement and different thruster quantities are shown in Figure 25. Figure 26 illustrates the different degrees of freedom.

![Possible ROV thruster arrangement](image-url)

*Figure 25: Possible ROV thruster arrangement*
Some of the major components of a thruster include: a power source, an electric motor, a motor controller, a housing, and a propeller. The power source can come from either a surface power source or a battery power source [7]. The electric motor can vary depending on the design. The most common motor used is the DC motor, which is superior in its power, availability, variety, and reliability, but the motor also needs a gear ratio to reduce its high speed [7]. A common type of DC motor is the permanent magnetic DC motor which works by having magnets pull on both sides of a three plated propeller, which moves by being repulsed by the positive and negative fields of the magnets, as seen in Figure 27. Another type of DC motor that is often used is called a brushless motor. Brushless motors have longer service life, less operating noise, and greater efficiency [7]. Another important component that a thruster needs to have is the propeller. Propellers are designed to move and push water in the opposite direction of the ROV’s motion. The propeller can either be designed for power or for speed. Sometimes, propellers are designed to be more efficient than the other propellers used so that, for example, the forward thrusters run more efficiently and provide more thrust [7]. The housing for a thruster is another important thruster component. The housing keeps parts of the thruster together and protects thruster parts from wearing or getting damaged. Figure 28 shows an example of a complete thruster system.
Figure 27: Commutator and brushes

Figure 28: Complete thruster diagram


2.6.2 Buoyancy

ROVs must be able to stay submerged for hours at a time, making buoyancy very important in terms of the time it takes to complete a mission and the amount of power it uses during its assignment. Most ROVs remain close to being neutrally buoyant. A neutrally buoyant ROV will maintain its depth in the water without input from the thrusters or effects from currents. ROV frames are typically made of solid, dense materials such as metals and high grade plastics. Dense materials such as these add weight to the ROV when it is submerged in water. ROVs are usually made somewhat neutrally buoyant with the addition of a molded foam piece that forms the top part of the structure. If an ROV was designed with no added buoyancy, it would have to rely entirely on downward facing thrusters to keep it in the water column, somewhat akin to how a helicopter operates. If an ROV was made far too buoyant, thrusters would have to overcome the buoyant force to submerge the ROV. The further an ROV is from neutrally buoyant, the more power the vertical thrusters will require maintaining a constant depth.

It is more common for an ROV to be positively buoyant rather than neutrally buoyant. There are many reasons for this. One is that a neutrally buoyant ROV must be moved both upwards and downwards by a thruster. This means that in order to maintain an accurate depth, the vertical thruster must switch directions constantly to correct for drift. Another reason is that if the ROV were to approach the bottom of a sea or lake, a downward pointing thruster will disturb the sediment and cloud the view of the cameras. The sediment may not settle for minutes afterwards. If the ROV is positively buoyant, it can rise from the sea floor without any input from the thrusters. Perhaps the most important reason for making an ROV positively buoyant is as a safety precaution. ROVs are expensive to design and manufacture; if something were to break and the connection to the ROV was lost, a positively buoyant ROV would return to the surface allowing for an easy recovery. A neutrally buoyant ROV would be prone to drift along the currents at an unknown depth, and a negatively buoyant ROV would fall to the seafloor. A positively buoyant ROV would require the vertical thruster to provide a constant downward force to maintain a depth. This is favorable compared to the alternative.

Small ROVs typically have 1-2 lbs of buoyant force and very large ROVs may have hundreds of pounds of buoyant force. ROVs usually do not have an active buoyancy regulation system because they are often lowered into the water with the help of a crane and cage. The
density of the water does not change much with water depth because water is incompressible. The salinity of the water and the temperature of the water have a much more significant effect on the density of the water. Because of this, the ROV's buoyancy can be trimmed with the addition of small weights to match the conditions it will be operating in. Very large ROVs and manned submarines have the ability to change their buoyancy without external help. These systems usually consist of a chamber filled with air, much like a ballast. If the submarine needs to submerge, the chamber is filled with water and the air is released. If it needs to rise, a tank of pressurized air forces the water out of the chamber, thus increasing the volume of the submarine and increasing its buoyancy [7].

2.6.3 Materials

Choosing the right material for an ROV is important to prevent components on the ROV from breaking. There are many components to consider when looking for a material that is structurally sound for an ROV. Some of these properties include tensile strength, compressive strength, density, elastic modulus, and melting point. Tensile strength is the maximum stress that can be applied to a material before it fails by pulling on it. Compressive strength, on the other hand, is the maximum stress that can be applied to a material before it fails by pushing on it. Elastic modulus is a measure of how stiff a material is and can be found by looking at the ratio of stress to strain. Density is the measure of a material’s mass per unit volume, which is very important when calculating buoyancy. The melting point of a material is the temperature at which a material melts and becomes liquid-like. This is not too important for ROVs as they rarely see high temperatures underwater.

2.6.3.1 Frame

The frame of the ROV provides a firm platform on which the necessary mechanical and electrical components can be fixed. ROV frames have been made of many different materials from plastic composites to aluminum tubing. In general, the materials used are chosen to give the maximum strength with the minimum weight. Since weight has to be offset with buoyancy, this is very critical.

ROV frames can vary greatly in size and can change size depending on how much volume the remainder of the components takes up. The frame is the outermost layer of an ROV,
so it is usually the first component that an observer notices. The frame not only helps to hold other parts in place, but it also protects them from damage caused by collisions and harsh conditions.

The shape of the frame can vary depending on the design on the ROV. They can be a simple square frame made of rods and elbows or a more specific frame that encompasses the majority of the ROV. An example of two different kinds of ROV frame designs can be seen in Figure 29. This alteration in the surface area of the frame usually depends on how much support the ROV needs and on how much positive buoyant force is available. Smaller frames have the benefit of being lightweight, but they do not protect the inner components as well.

![Figure 29: A PVC frame (left) and a polypropylene frame (right)](image)

When choosing a frame for an ROV, it is important to decide on a material that is lightweight enough for the ROV to be neutrally buoyant and strong enough to survive without structural failure at the depths in which the ROV will be working. The properties of several different kinds of materials can be seen in Table 2 [17]. Aluminum, polycarbonate, and polypropylene would make for good materials to be used for an ROV frame. When compared to the more commonly used stainless steel, it is clear that these materials are much less dense and a fair amount cheaper too. However, stainless steel is also much stronger than the other materials both in tension and compression in addition to having higher stiffness. This does not mean that aluminum, polycarbonate, and polypropylene are too weak, as they have enough strength to withstand the pressures of the deep sea and can survive low to medium amounts of impact.
Table 2: Possible materials for an ROV frame

<table>
<thead>
<tr>
<th>Material</th>
<th>Density (lb/ft³)</th>
<th>Tensile Strength (ksi)</th>
<th>Compressive Strength (ksi)</th>
<th>Elastic Modulus (10⁶ psi)</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aluminum</td>
<td>156-181</td>
<td>8.4-17.9</td>
<td>4.35-72.5</td>
<td>9.86-11.3</td>
<td>$16.7/60'(L)</td>
</tr>
<tr>
<td>Polycarbonate</td>
<td>71.2-75.5</td>
<td>8.7-10.5</td>
<td>10-12.6</td>
<td>0.29-0.354</td>
<td>$15/12'(L)</td>
</tr>
<tr>
<td>Polypropylene</td>
<td>55.6-56.8</td>
<td>4-6</td>
<td>3.84-8.01</td>
<td>0.13-0.225</td>
<td>$18/60'(L)</td>
</tr>
<tr>
<td>Stainless Steel</td>
<td>474-506</td>
<td>65.6-325</td>
<td>4.35-72.5</td>
<td>27.4-30.5</td>
<td>$142.5/12'(L)</td>
</tr>
</tbody>
</table>

2.6.3.2 Floatation

The floatation of an ROV is where the majority of its positive buoyancy comes from. No matter how heavy the other parts of the ROV end up, as long as the floatation provides enough buoyancy, the ROV can become neutrally or positively buoyant. Floatation components need to be lighter than water in order to counteract the other components which are heavier than water.

A commonly used material for floatation is polymer plastic foam. Plastic foams are made up of small air bubbles that are surrounded by membranes of plastic material. There are two different kinds of foams: open-cell and closed-cell foams. Open-cell foams contain pores that are connected to each other and form an interconnected network that is relatively soft. These are porous meaning that water will flow through them. An example of open-cell foam is a sponge. Closed-cell foams do not have these pores and are resistant to water. An example of a closed-cell foam is a Styrofoam coffee cup. The differences between open-cell and closed-cell foams can be seen in Figure 30. Closed-cell foams are the better choice for use in the floatation of an ROV because they do not let in water, keeping air bubbles trapped inside which helps to create positive buoyancy. Closed-cell foams also have higher dimensional stability, low moisture absorption coefficients, and higher strength compared to open-cell foams.
The shape of a floatation part depends on the intended shape of the ROV and its frame. Most polymer foams can be molded into any shape, as they are usually made by a reaction which causes two liquids to expand and harden into a solid shape. The shape of the floatation is also important when looking at the drag forces which are present on the ROV.

Similar to choosing the material for an ROV frame, the floatation of an ROV needs a material that is lightweight, specifically lighter than water, yet strong enough to survive at great depths underwater. Table 3 shows the properties of two commonly used foams as well as polymer foams in general. Syntactic foams are air or micro balloon structures encased within a resin body. They are more suited for deep-sea ROVs since they are significantly stronger than other polymer foams. Unfortunately, syntactic foam is a rather expensive material to produce. Rather than this material, simple observational ROVs tend to use rigid polyurethane foam since they are effective at depths up to 1000 feet.
Table 3: Material Properties for some polymer foams

<table>
<thead>
<tr>
<th>Material</th>
<th>Density</th>
<th>Tensile Strength</th>
<th>Compressive Strength</th>
<th>Elastic Modulus</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rigid Foam (HD)</td>
<td>10.8-29.3 lb/ft³</td>
<td>0.174-1.0 ksi</td>
<td>0.406-1.74 ksi</td>
<td>0.029-0.0956 10⁷ psi</td>
<td>1.36-2.72 $/lb</td>
</tr>
<tr>
<td>Syntactic Foam</td>
<td>20-40 lb/ft³</td>
<td>N/A</td>
<td>3.5-12.3 ksi</td>
<td>163-394 ksi</td>
<td>$22.5/lb Buoyancy</td>
</tr>
<tr>
<td>Polyurethane Foam</td>
<td>2.16 lb/ft³</td>
<td>30-450 psi</td>
<td>30-230 psi</td>
<td>N/A</td>
<td>Coverage</td>
</tr>
</tbody>
</table>

Another option for use as floatation could be a PVC pipe with air trapped inside of it. Since air is lighter than water, this can make for an effective, low-cost floatation, as long as the PVC remains watertight. If this is sealed properly, it should be able to withstand similar water pressures as the polymer foams can. Figure 31 shows different floatation components: syntactic foam (left), polyurethane fiberglass (middle), and PVC floatation (right).

![Figure 31: Floatation components](image)

2.6.4 Lights

Lighting underwater is a difficult task. Light is absorbed much more quickly underwater than above it. There are also more debris and particles floating around in water than there is in open air. As such, finding lighting which is able to illuminate effectively underwater is a little more challenging than finding a light for the same job above water.

One difference between lighting underwater and above water is that there is a higher amount of matter suspended in the water. This can cause an effect called backscatter, which causes the light to bounce in random directions, including back towards the camera. Figure 32 demonstrates this effect.
To help minimize this effect, the light sources of the ROV can be separated from the camera by a small distance. This minimizes the amount of backscatter that the camera perceives, while still illuminating the target as effectively as possible. Figure 33 illustrates this configuration.

Lights that are used underwater must also be more powerful than lights above water, since the light is absorbed much quicker in water than in air.

One topic to consider is the type of light that will be used. There are many different types of bulbs, from light emitting diodes (LED) to compact fluorescent to halogen. LED bulbs are the most useful for underwater applications as they are much more energy efficient, and can provide enough illumination. The energy consideration is important, especially if the ROV will be running off of batteries, and if the lights aren’t very efficient, it will reduce the operating time of the ROV.
Heat dissipation is another topic to consider when thinking about lighting. One byproduct of all lighting sources is heat. This is another reason why choosing an LED lighting source proves to be superior to other bulbs. LEDs typically give off less heat than other lights. While this does not seem like it would be much of a problem in underwater environments, it still needs to be a consideration when choosing a light source. The housings of many spotlights are built with heat dissipation in mind. For example, the Rotundity CREE MR16 light has an efficient heat dissipating housing, shown in Figure 34 below.

![Figure 34: Rotundity CREE MR16 LED](image)

Lights that are used underwater must also be more powerful than lights above water, since the light is absorbed much more quickly in water than in air. The light shown above has a luminous intensity of 500 lumens. A lumen is described by the following equation.

\[
lumens = cd \times str
\]

A lumen is the measure of intensity of the light, shown in candelas (the average intensity of a candle), multiplied by the surface area of the sphere onto which it is projected, measured in steradians. Using these measurements, it is possible to find a lighting source that can provide enough light for an ROV.
2.6.5 Sensors

The sensors of an ROV are used to provide information about both the ROV’s environment and its system health. This information, transmitted as data to the operator or interpreted automatically by an on-board processor, can be used for simple data collection or for adjusting the performance of the ROV.

When looking for or designing sensors, a few key aspects are considered. The range of the sensor indicates the array of environments in which the sensor will function. The resolution of the sensor specifies the accuracy of the measurements collected. Also worth considering is the format of the sensor and its data: how it is connected to the system, what communication protocol it uses, and what level of control the operator has over it. Although typically insignificant for sensors, the voltage draw and power consumption of the device are also factored into the overall electrical requirements of the system.

As with many other components, an ROV’s sensors depend heavily on its intended functionality and environment. There are several sensors, however, that are common among many, if not all, modern ROVs, such as pressure, temperature, and orientation [18]. A summary of these common sensors can be seen in Table 4. Most of these sensors are not typically designed by the creators of the ROV, and instead are provided by third parties. For this reason, this project looked at sensor use and functionality, rather than specific sensor design.

Table 4: Common sensor types for ROVs

<table>
<thead>
<tr>
<th>Sensor Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pressure</strong></td>
<td>Determine ROV depth</td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td>Detect abnormalities in water</td>
</tr>
<tr>
<td></td>
<td>Stay within safe operating range</td>
</tr>
<tr>
<td><strong>Orientation</strong></td>
<td>Determine local ROV position</td>
</tr>
<tr>
<td></td>
<td>Accumulate over time for path tracking</td>
</tr>
</tbody>
</table>
2.6.5.1 Pressure Sensors

Pressure sensors are a must-have component for any ROV that dives more than a few meters. The pressure measurement itself is useful for staying within depth ranges that the frame material can handle. Additionally, the pressure can be used to calculate the current depth of the ROV, often to tenths or hundredths of a meter. Knowing exact depth in an unknown environment can be very important for applications meant to inspect or map specific locations. Calculating the depth based on pressure is done frequently in fluids applications:

\[
\text{depth} = \frac{P - P_{\text{atm}}}{\rho \ast g}
\]

where \( P \) is the water pressure measured by the sensor, \( P_{\text{atm}} \) is the atmospheric pressure, \( \rho \) is the density of the fluid (water in this case), and \( g \) is the constant that represents gravitational acceleration.

2.6.5.3 Temperature Sensors

Temperature is important to know about for an ROV since as depth increases in the water, the temperature decreases. Additionally, for ROVs that work on or near oil platforms, an oil leak or explosion would have an impact on the temperature of the surrounding water which should be acknowledged by the ROV in order to alert people on the surface. Many ROVs collect temperature information through the use of thermistors. Thermistors are resistors whose resistance is greatly influenced by temperature. Their resistance varies linearly with the temperature multiplied by a constant. Depending on the value of the constant, temperature and resistance could be either proportional or inversely proportional.

An example of a commonly used thermistor is the DS18B20 Model. This has a one wire interface and it is relatively inexpensive, small, and power efficient. It operates with temperatures that range from -55°C to 125°C (-67°F to 257°F) with accuracy of \( \pm 5°C \). This thermistor can convert 12-bit temperature readings into digital readings in 750 milliseconds. The temperature sensor itself is very small and consists of three wires: the ground, the data input/output, and the power supply voltage [19]. With this configuration, the temperature sensor
only requires one data port on a microcontroller. This entire system can be made waterproof so that it can be attached directly to the ROV. A picture of the DS18B20 can be seen in Figure 35.

![DS18B20 thermistor](image)

**Figure 35: DS18B20 thermistor**

### 2.6.5.2 Orientation Sensors

Orientation sensors are used for determining the ROV's position in three-dimensional space. They are especially important for navigation and positioning when video, sonar, or other visual feedback is unavailable.

One sensor used for orientation monitoring is simply a compass. Though they come in many designs (magnetic, fluxgate, etc.), compasses have one purpose: provide a heading measurement. This heading measurement represents the rotational orientation of the ROV in the plane relative to the seabed. This heading can in turn be combined with a distance measurement (from motor encoder readings or back EMF measurements) in order to track the absolute position and traveled path of the ROV. This orientation method is often referred to as dead reckoning, and is discussed further in section 2.6.7 Navigation.

Another sensor used for orientation is an Inertial Measurement Unit (IMU). IMUs also provide data readings for velocity and gravitational forces in addition to orientation; these measurements are typically derived from a three-axis accelerometer. Though IMUs are often paired with a global positioning system (GPS), it is not the case in deep-sea applications simply because GPS satellite signals do not reach far below the surface of the water. Due to their
complexity, IMUs are expensive relative to other orientation sensors, and are usually reserved for advanced navigation purposes only.

Longer-distance orientation (rather than orientation of the ROV in three-dimensional space) can be determined in several ways as well. Sonar is one such method, and is frequently used to navigate and map unknown spaces. In tandem with sonar, waypoint navigation is used to move around known environments. Waypoint markers, such as radio-frequency identification (RFID) chips or sonar reflectors, are set up ahead of time (often by the ROV itself), and are detected by a receiver on board the ROV [20].

Sensors allow for an ROV to collect data about its surroundings and transmit that data to the surface. This makes it so that the person operating the ROV has an understanding of what is happening to the ROV without having to actually be underwater. Sensors find information about the surroundings of the ROV by understanding either the ROV’s location or the condition of the ROV’s environment. There are many different kinds of sensors used for accomplishing these tasks. Some commonly used sensors in ROV systems include: radiation sensors, conductivity sensors, temperature sensors, depth sensors, pressure-sensitive depth transducers, magnetic flux gate compass modules, slaved or rate gyros, ultrasonic thickness gauges, imaging sonar, acoustic positioning devices, digital cameras, and multi-parameter environmental sensors.

There are several ways in which ROVs keep track of their orientation. For example, a miniature gyro is often used in ROVs made by Oceaneering. Mini Gyros, as shown in Figure 36, combine tri-axial accelerometers, tri-axial gyros, and on-board processors running a sophisticated sensor fusion algorithm to provide static and dynamic orientation, as well as inertial measurements. This advanced yet compact technology can calculate acceleration, angular velocity, and changes in pitch, roll, and yaw. When the Mini Gyro is connected to a computer, a person could rotate the gyro and see a computerized model rotating in a nearly one to one ratio with their movements. The Mini Gyro has an advantage over other forms of determining orientation because it is able to collect data along three axes rather than simply one axis [21].
The Mini Gyro is a great example of how a sensor can find several different pieces of data at once. Unfortunately, the Mini Gyro is rather expensive since it relies heavily on computers, and not every ROV company has the budget that Oceaneering does. Another well-known company in the field of ROVs is SeaBotix. SeaBotix uses an ultra-short baseline (USBL) positioning system to determine where its ROVs are in the water. USBL is a method of underwater acoustic positioning in which a transceiver mounted to a ship receives a signal from a transponder connected to an ROV and then calculates its range and angle with respect to the ship. The transceiver emits an acoustic wave which bounces off the transponder on the ROV and back to the ship as Figure 37 shows. The amount of time it takes to return and the changes in the phase angle are used to calculate the range and bearing of the ROV respectively [22]. This may not exactly determine the acceleration or velocity of the ROV, but it provides enough information for personnel on the surface to find out where the ROV is beneath them.
Another way to determine the orientation is to use a simple tilt sensor. A tilt sensor, displayed in Figure 38, works similar to an electrical switch. Inside a tilt sensor are two conductive metal spheres which remain in contact with each other in order to complete a circuit. As the sensor is tilted to increasing angles, eventually the metal spheres lose contact with each other, breaking the circuit and stopping the flow of electricity through the sensor. This activation angle is usually about 30 degrees. Tilt sensors have the benefit of being very small, inexpensive, low-power, and easy to use. Unfortunately, tilt angles do have some problems since they are not very precise with their measurements as they do not tell the exact angle of the ROV.

Figure 38: Tilt sensor

2.6.6 Tether

While some tetherless ROVs do exist, such as Saab's Seaeye Sabretooth AUV/ROV [20], it is not a prevalent design in the deep-sea industry, especially if the ROV does not have a high degree of autonomy. This is primarily due to the difficulty of underwater wireless communication; radio frequency (RF) waves with desirable bandwidths cannot be transmitted more than a few hundred meters at normal frequency, and lower frequencies limit data transmission rates [7].

Because of these communication restrictions, a large portion of ROVs are tethered devices. This is especially true for teleoperated ROVs (as opposed to autonomous ROVs), which need the higher bandwidth communication a tethered connection provides for visual feedback and constant operator control.

A tether is essentially a bundle of various cables, all of which are used in some part of the ROV operation. This bundle may consist of, but is not limited to: cables for power, visual data, sensor data, and operator commands, as well as floatation, insulation, and casing. The cables commonly consist of coaxial cables or fiber optics. Coaxial cables are the cheaper, lower-
bandwidth solution, while the more expensive fiber optic cables have better transfer rates and lighter construction [23].

The power cable provides the required voltage to the ROV, though there is often a backup battery on board. The visual data cable is used to transmit high-bandwidth video (i.e., camera feed, sonar visualization) for the operator, typically bypassing the on-board microcontroller and using fiber optic cable so as to reduce signal latency. The sensor data cables carry environment and system health information for the operator that has been aggregated and processed by the on-board microcontroller. The operator command cables carry instructions to the ROV from the operator. Coaxial cables are effective for sensor and command cables, due to their lower bandwidth requirement. Floatation is used to keep the tether neutrally buoyant, and can come in many forms. The insulation and casing are used between and around the other cables and floatation to unify them into a single tether. An example of a tether can be seen in Figure 39 [7]. It is common practice for this tether to only extend from the ROV to the tether management system, discussed in the section 2.4.1.1 Umbilical, TMS, and Tether.

Figure 39: Cross-sectional view of ROV tether
2.6.7 Navigation

The harsh, frequently unknown environments in which most ROVs travel can often be difficult to navigate. If the ROV has any degree of autonomy, the navigation problem is compounded; autonomous underwater vehicles (AUVs) must rely almost entirely upon navigation strategies, as they receive little to no operator input. As mentioned by Yuh and West [23], a couple of the most common methods for ROV navigation include, but are not limited to: acoustic positioning systems and dead reckoning with inertial navigation.

Acoustic positioning navigation relies on previously placed transponders with which the ROV may send and receive signals. Depending on the environment, different scales of acoustic positioning systems may be utilized, placing the transponders either at pre-mapped locations or on the surface ship itself. Similar to sonar or echolocation, the ROV measures the time it takes between sending and receiving the signal to determine the distance to the transponder. Some ROVs that use acoustic positioning systems are also configured to measure the angle to the transponder, which yields more accurate positioning data. The frequency of the acoustic signals can be adjusted to span long distances at the cost of lower communication rates, as seen in Table 5 [23].

<table>
<thead>
<tr>
<th>Frequency Type</th>
<th>Frequency Range (kHz)</th>
<th>Maximum Range (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low frequency (LF)</td>
<td>8 – 16</td>
<td>&gt; 10000</td>
</tr>
<tr>
<td>Medium frequency (MF)</td>
<td>18 – 36</td>
<td>2000 – 3000</td>
</tr>
<tr>
<td>High frequency (HF)</td>
<td>30 – 60</td>
<td>1500</td>
</tr>
<tr>
<td>Extra high frequency (EHF)</td>
<td>50 – 110</td>
<td>&lt; 1000</td>
</tr>
<tr>
<td>Very high frequency (VHF)</td>
<td>200 – 300</td>
<td>&lt; 100</td>
</tr>
</tbody>
</table>

Dead reckoning, as mentioned in section 2.6.5.2 Orientation Sensors, is used to estimate the location of some moving device. Essentially, with a heading and either distance or velocity information, one can determine the absolute position of the ROV as well as update the traveled path. However, Yuh and West mentioned that the accuracy of this system is often compromised by ocean currents, which are undetectable by distance and velocity measurements. Inertial measurements help to reduce this issue of drift by “integrating the acceleration twice in time”
[23], adding another level of awareness to the dead reckoning navigation system. A similar process is frequently used for navigation with boats and aircraft, as seen in Figure 40.

![Figure 40: Dead reckoning](image)

### 2.6.8 Camera

Visual inspection requires that the ROV have a camera of some kind. The camera not only acts as visual feedback for the engineer who is driving the ROV, it allows the ROV to record information about underwater structures for use in structural health monitoring. While there are many options when considering cameras, the majority of ROVs use cameras that are designed specifically for its purpose. Many work class ROVs have camera units that are designed separately with their own underwater housing that can withstand high pressures. The cameras that are used don’t need to be of the highest quality, as the pictures and videos themselves are not the purpose of the ROV. Observational ROVs often use high definition cameras, as their main purpose is to take pictures, often for scientific or exploratory purposes.

Some different types of cameras that can be considered for use in an ROV are board cameras, complete camera modules, and simple web cameras. Board cameras are extremely basic cameras which consist of the photo sensor, some peripheral electronics, and a lens. These cameras tend to be cheap since they have no housing, and must be mounted and housed by hand. Figure 41 is an example of a board camera.
On the opposite side of the spectrum, a complete camera module is ready to go upon purchase, and is easy to use. Generally these modules have their own software, or work with standardized software for video capture. An example of a complete camera module would be a GoPro camera, which can be seen in Figure 42 below.

Another option for cameras is to use a webcam. These cameras are typically very cheap and easy to use cameras that connect directly to a computer and have easy to use software. An example of a webcam can be seen in Figure 43 below.
When considering cameras, board cameras specifically, one of the major features to consider is the type of connection. There are many different types of connections where video is concerned. These include USB, serial, composite video, firewire, and many more. Some of these connections are used for connecting cameras to specific devices. For example, composite video is very useful when connecting a video camera directly to a TV or monitor. One of the easiest and most accessible types of connections is USB, which stands for Universal Serial Bus. This type of connection has become extremely popular and is an industry standard as far as computer devices are concerned.

2.6.9 Microprocessor

On the surface, the engineer driving the ROV is using a control unit. To communicate effectively with the ROV, the ROV must have a microprocessor. The job of the microprocessor is to communicate with the surface unit, as well as control each of the electronic components on the ROV. The microprocessor will be able to control when components such as the camera, lights, and motors are turned on and off. The microprocessor also receives data from all of the sensors. It is the job of the microprocessor to receive, store, and send this data back to the surface unit, which will then use the data to gather information about the ROVs surroundings. Figure 44 is an example of the movement of data in an ROV system.
In Figure 44, the blocks on the right are all of the sensors that are being used in the ROV. Each of these sensors is on at all times, and sends their data to the microprocessor. The microprocessor receives this data and might then make some decisions based on it, such as if the tilt sensor indicates that the ROV is at a bad angle. The blocks on the left are the electric components which are controlled by the microprocessor. These include the motors, lights, camera, health and safety indicators, and any additional manipulators. The microprocessor controls the power flow to these units. The microprocessor also receives input from the health and safety blocks, shown at the bottom of the diagram. These blocks are used to make sure that there is nothing wrong within the ROV itself. From there, the microprocessor sends data to the surface unit through a tether. It also receives data from the surface through the same tether.

There are many different manufacturers that build microprocessors, and microprocessor development boards. There are many different development boards that provide useful functions at low costs. The best microprocessor for use in a Class 1 Observational ROV has a low power
draw, and is able to communicate with many different devices at the same time. This can be seen in the system diagram shown in Figure 44. The microprocessor needs to have input and output ports for each device connected to it.

On the surface, to be able to control the ROV, there will need to be a work station computer with some form of data display. Often, this display takes the form of a graphical user interface (GUI) which is designed to provide quick and intuitive access to all important information about the ROV. While there are many options for a GUI, the cheapest and most accessible designs tend to come from widely accepted software, such as National Instruments’ LabVIEW or Java’s JFC/Swing libraries. Regardless of software, user interface must allow the operator to view sensor data, as well as send commands to the microprocessor.
CHAPTER 3: The ROVEN Sealion

3.1 Introduction

This chapter details the design process the project team used to create the ROVEN Sealion. After highlighting several original design ideas, the chapter discusses the physical and electrical component selection process and the analysis performed on these components. The chapter also outlines the functionality of the software powering the ROV. The chapter concludes with details of and observations made during a final test of the ROVEN Sealion.

The analysis, design, testing, and iteration outlined in Chapter 3 led to the final design of the ROVEN Sealion, as seen in Figure 45 and in Figure 46. Table 6 contains detailed final specifications for the ROVEN Sealion as compared to those of a typical deep-sea observational ROV.

Figure 45: Final design of the ROVEN Sealion
Table 6: Comparison between deep-sea ROV and this project's design

<table>
<thead>
<tr>
<th>Specification</th>
<th>Deep-Sea</th>
<th>Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Depth</strong></td>
<td>AVG: 3200 ft (975 m)</td>
<td>AVG: 25 ft (7.62 m)</td>
</tr>
<tr>
<td></td>
<td>MAX: 5000 ft (1524 m)</td>
<td>MAX: 50 ft (15.25 m)</td>
</tr>
<tr>
<td><strong>Maximum Pressure</strong></td>
<td>2100 psi (14480 kPa)</td>
<td>26 psi (178 kPa)</td>
</tr>
<tr>
<td><strong>Minimum Temperature</strong></td>
<td>-4° C</td>
<td>5° C</td>
</tr>
<tr>
<td><strong>Speed</strong></td>
<td>3 - 5 Knots (1.5 - 2.6 m/s)</td>
<td>3-5 Knots (1.5 - 2.6 m/s)</td>
</tr>
<tr>
<td><strong>Operation Time</strong></td>
<td>24 Hours</td>
<td>1 Hour</td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td>3 x 3 x 2 ft (0.91 x 0.91 x 0.61 m)</td>
<td>2.5 x 2 x 1.5 ft (0.76 x 0.61 x 0.46 m)</td>
</tr>
<tr>
<td><strong>Lighting</strong></td>
<td>Able to see 15 ft (4.6 m)</td>
<td>Able to see 2.5 ft (0.76m)</td>
</tr>
<tr>
<td><strong>Recovery</strong></td>
<td>Uses a cage/pulley</td>
<td>Uses onboard motor, positively buoyant</td>
</tr>
<tr>
<td><strong>Weight</strong></td>
<td>300 lbs</td>
<td>122 lbs</td>
</tr>
<tr>
<td><strong>Transmission (information)</strong></td>
<td>Tether</td>
<td>USB</td>
</tr>
<tr>
<td><strong>Frame</strong></td>
<td>Polypropylene</td>
<td>Polypropylene</td>
</tr>
<tr>
<td><strong>Floation</strong></td>
<td>Syntactic Foam</td>
<td>Polyurethane Foam</td>
</tr>
<tr>
<td><strong>Movement</strong></td>
<td>4 DOF</td>
<td>3 DOF</td>
</tr>
<tr>
<td><strong>Video Camera</strong></td>
<td>Black/White &amp; Color, 60 fps, 1080p, USB</td>
<td>Color, 30 fps, 360, USB</td>
</tr>
<tr>
<td><strong>Sensors</strong></td>
<td>Depth (pressure), Temperature, Compass, Sonar</td>
<td>Depth (pressure) Temperature, Compass</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>Tether</td>
<td>On-board battery 12V 18 Ah</td>
</tr>
</tbody>
</table>
3.2 Original Design

The design team went through several iterations before deciding on the final design for the ROV. To begin, the team started off with making a basic concept model of an ROV with all the possible components that an observational ROV with payload could include. This was essentially a box to put all of the group’s ideas together. This brainstorming model, seen in Figure 47, shared some similarities with the final model of the ROVEN Sealion. Like the final design, this model featured a camera and lights in the front of the vehicle, two forward motors on the sides, a top motor for movement up and down in the center of the floatation, a battery and tether in the back of the ROV, and a square frame to hold all the components together. This model differed from the final model because it featured a brush or a pressure washer for cleaning while performing structural health monitoring, and a pressure tank to control buoyancy. These non-essential components were omitted from future designs due to budget constraints. Once the ROV was constructed, there was enough room to include the option for a payload, so that future teams could add functionality to the ROVEN Sealion.

The original model also featured two rim-driven thrusters. These motors would have allowed the ROV to have an extra degree of freedom in the translational sway direction. Because of their hub-less design, rim-driven thrusters would have added more control to the movement of the ROV as they allowed a greater flow rate [24]. The design team decided to exclude these
thrusters from the final design because rim-driven thrusters were much too expensive for the project's budget and the extra degree of freedom was not necessary as the ROV could have sufficient maneuverability with a combination of yaw and surge. A later design, shown in Figure 48, also included these rim-driven thrusters and showed a potential orientation for these motors. This model was the first SolidWorks rendering of the ROV. It had a different frame that consisted of two rounded components connected by a rod and the floatation. The floatation was also different from the final model because it was assumed to be a container: hollow with an opening on top for releasing air or water.

Another model, which was the last rendition of the ROV before the final version of the ROVEN Sealion, can be seen in Figure 49. This model was very similar to the final design and it included many of the same materials and products, but there were a few differences as well. The shape of the frame was slightly different; the trapezoidal shape produced less drag than equivalent rectangular shapes. The ROV's frame was changed to a rectangular shape because the difference in drag was negligible on an open frame, and because it would simplify the joint connections between rods. This model had elbow connectors made of galvanized steel which could change the angle of the polypropylene rods. These were changed to PVC elbows since they cost and weigh less. The model included all three thrusters which were used in the final design as well as their correct placement, but the ways in which they were connected was quite different. The side thrusters were mounted on rail brackets that were set tangent to the thruster and attached by drive clamps.Threaded rods radiate outward from a cylinder that supports the
center thruster. These thruster mounts were changed to 3.5 inch diameter anvil pipe clamps with steel rods for each thruster. These were simpler to implement into the ROV.

The preliminary model also had a different floatation that was twice as thick as the final design, with no rounded edges to alleviate the drag forces. This model did not feature a mount for the battery and there were no containers for the electrical components. The camera had a different housing that was box-shaped rather than dome shaped, and was held in place by rods rather than railings.

3.3 Thrusters

The following sections discuss the selection of the thrusters as well as the experiments to determine the actual thrust of the selected component. Once the thrusters were selected and tested, a method for mounting them to the frame was designed.

3.3.1 Selection

There are a variety of ways in which an ROV can move about underwater. The most common method of movement is using a motor to spin a propeller which provides a force that pushes the ROV. The motor can vary depending on the specifications that the ROV would
require including different thrusts, weights, and depths. The ROVEN Sealion, a Class 2 ROV, requires a greater thrust than typical observational ROVs. The cost and weight must not be too high, as there will need to be multiple motors for the thrusters. The ROV should be able to support the weight and the thrust generated by the motors. Several different motors were found that could have potentially worked for the team’s ROV; these motors and their properties, many of which were determined empirically, can be seen in Table 7.

### Table 7: Comparison of motors

<table>
<thead>
<tr>
<th>Type</th>
<th>Thrust</th>
<th>Width of Thruster</th>
<th>Weight</th>
<th>Depth</th>
<th>Voltage</th>
<th>Power</th>
<th>RPM</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brushless</td>
<td>5.3 ft*lb</td>
<td>5 in</td>
<td>17 lbs</td>
<td>____</td>
<td>9 kW</td>
<td>12,000</td>
<td>$500</td>
<td></td>
</tr>
<tr>
<td>SeaBotix</td>
<td>6.4 ft*lb</td>
<td>3.74 in</td>
<td>1.5 lbs air</td>
<td>150 m (1000 ft)</td>
<td>0.11 kW</td>
<td>____</td>
<td>$495</td>
<td></td>
</tr>
<tr>
<td>Bilge Pump</td>
<td>1100 gallons/ hour</td>
<td>2-3/8 in</td>
<td>0.75 lbs</td>
<td>7 ft</td>
<td>12V</td>
<td>____</td>
<td>$31</td>
<td></td>
</tr>
<tr>
<td>Trolling Motor</td>
<td>30-lb</td>
<td>6.2 in</td>
<td>15 lbs (w/ shaft)</td>
<td>12V</td>
<td>____</td>
<td>____</td>
<td>$99</td>
<td></td>
</tr>
</tbody>
</table>

A popular type of motor used by many ROVs ranging from observational to small work class ROVs is brushless motors. Basic brushless motors do not include a propeller and need to be converted into a thruster with a propeller for use in an ROV. An example of a brushless motor is the SeaBotix thruster which is used by the SeaBotix ROV. The company sells the thruster and provides basic technical blue prints for the thruster as well. The only problem with the brushless motor and SeaBotix thruster is that the cost is too high for the project’s budget. At an average of $500 per thruster these would not be ideal even though they meet the rest of the ROV’s specifications.

Another type of motor that was available was a bilge pump. Bilge pumps are popularly used for homemade ROV projects since they are inexpensive and are not very large or heavy. Like the brushless motor, this motor needed to be fitted with a propeller, but bilge pumps can be more easily converted into ROV thrusters and instructions for this conversion process can be found easily on the internet. Although the bilge pumps are cheap, small, and light, the pumps do
not provide enough thrust. Bilge pumps work better for small observational ROVs and would not be suited for the ROVEN Sealion.

The last motor that was looked at was an electric trolling motor. Trolling motors are mainly used for small fishing boats. Since these motors are used for movement underwater, they are already equipped with propellers and are sealed tight to be waterproof. The weight of the motor was not a factor that the team considered; it could be easily offset by the floatation, as discussed in section 3.5 Buoyancy. The thrust and cost of the motor were more than desired, but still fit within the team’s specifications for a thruster. The model of trolling motor selected was the Minn Kota Endura C2 30, which is the same one detailed in Table 7.

3.3.2 Experiments

The Minn Kota Endura C2 motors needed to be tested experimentally to determine what force and speed both reverse and forward thrust provide how much current the motor uses for each speed setting, and the flow rate that the thrusters make. The testing area was located in the WPI biomechanics lab and consisted of a tub of water, a mounted force sensor, and a high-amperage 12-volt power source. At this point, it was known that the trolling motor was going to be used and one had already been acquired, but these tests needed to be done to confirm that the motor worked sufficiently before buying more motors. The motor that was used still had its long shaft and mounting clamp mechanism that it uses to attach to boats, but the top cover where the handle control resided was removed. The tub had a wooden plank across it to which the mounting clamp was attached. The motor was put in the tub with enough room so that it did not hit the bottom or splash water out of the top. The motor mount was fixed to the plank, which made it so that all movement of the motor would occur along the flow direction simulating the placement that would occur on the ROVEN Sealion. There was a quick release tilt lever on the mounting clamp which simulated a fulcrum around which the entire shaft and motor could be tilted. A diagram explaining the setup used to test the motor can be seen in Figure 50. The wires coming out of the shaft were attached to both a handle controller and the power supply which was set at 12 volts.
The first set of tests measured the electrical current (in amperes) drawn by each motor speed. To do this, the motor was simply run under its normal conditions at each speed; the Minn Kota Endura C2 has five forward and three reverse speeds. The average currents obtained from the power supply display for each speed setting can be seen in Table 8.
The second test was designed to find the amount of force that the thrusters produced. Using the set up in Figure 50, the top of the motor shaft was tied to a force sensor by using a string. The force sensor was connected to a vertical metal rod fixed to the tub. The force sensor was connected to a program called Logger Pro in order to record the readings from the force sensor. The fulcrum was released on the clamp, enabling the thruster to move the shaft. The string was pulled taut as the shaft exerted a force on the string. To account for this, the Logger Pro program was zeroed and recorded only forces that the thruster provided. The thruster was turned to each of its forward speed settings, and the force for each was recorded. For the reverse settings, the motor rig was reversed so that the force of the thruster still pulled on the string so that a force could be recorded. The average thrusts can be seen for each speed setting in Table 9.
Table 9: Average thrust for each motor speed

<table>
<thead>
<tr>
<th>Speed Setting</th>
<th>Thrust (lbs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>13.000</td>
</tr>
<tr>
<td>4</td>
<td>11.015</td>
</tr>
<tr>
<td>3</td>
<td>8.833</td>
</tr>
<tr>
<td>2</td>
<td>5.853</td>
</tr>
<tr>
<td>1</td>
<td>4.424</td>
</tr>
<tr>
<td>-1</td>
<td>1.304</td>
</tr>
<tr>
<td>-2</td>
<td>2.452</td>
</tr>
<tr>
<td>-3</td>
<td>5.431</td>
</tr>
</tbody>
</table>

The results from these experiments seem to be inconsistent with the thruster's expected values, especially in the second experiment. The advertised thrust was 30 lbs, but the thrust recorded in the experiment was about half as much, at 13 lbs. The problem was that there may have been a pseudo-vortex ring effect which was caused by the recirculation of water in the tub. The thrust from the thrusters caused the water to move and circulate about the small tub, making the flow and thrust readings inaccurate. But the experiments were still helpful as they gave approximate values and provided visual evidence of how well the motors worked. As a result of the experiment, it was decided that the amount of thrust provided and the current needed were within the values that were desired, therefore the Minn Kota Endura C2 trolling motor used in these experiments was used for the final design of the ROVEN Sealion.
3.3.3 Mount Design

The trolling motors were mounted on the ROV using steel threaded rods. Figure 51 shows an image of the top motor mount. A pair of pipe clamps holds the motor in place. For the top motor, the threaded rods are embedded in the floatation, and are held by steel trusses at each end. These components are held in place by steel nuts.

In-between the thruster and the pipe clamp is a rubber sheath which eliminates vibrations from the motor. Electric motors have very little vibration and therefore a single rubber sheath was effective. This rubber also creates a greater friction between the thruster and the pipe clamps so it doesn’t slip out or vibrate out while in use.

The thrusters on the sides of the ROVEN Sealion are similarly assembled, as seen in Figure 52. The difference between the top and side thrusters is that the side thrusters are at the end of the threaded rods. These thrusters are held with PVC slip T’s on the frame, as opposed to steel trusses. Figure 53 shows how these components are connected in the assembly. The top of the thrusters were sealed using silicon and urethane foam.
Figure 52: Side thruster mount

Figure 53: Side thruster mount detailed view
3.4 Material Selection

Deciding on what material to use for the structure of the ROVEN Sealion was important in determining what properties the ROV would have. The choice of material had an impact on the strength and durability of the ROV depending on the tensile, compressive, shear, and yield strength as well as the elastic modulus and fatigue limit. Since the ROV needed to be neutrally buoyant, it was crucial that the materials have a low density in order to minimize the weight and provide a buoyant force to help negate the total weight of the ROV in the water. The material also needed to be able to withstand the temperatures and pressures that could be present deep underwater.

3.4.1 Frame

The frame made up the "backbone" of the ROV by holding all of the key components of the ROV together. The ROVEN Sealion has a box-shaped design that consisted of an adaptable and sturdy frame of connected rods, as seen in Figure 54. The frame's open shape allows water to flow through the ROV, reducing drag, as well as leaving space for the addition of various components.

The rods used for the frame needed to be strong, lightweight, and cost efficient. Metals such as aluminum were candidates for the rods since they were strong enough to support any loads that the ROV would be subjected to and they were very cheap to manufacture. However, aluminum is much denser and heavier compared to other materials, such as plastics. The design
team decided to make the frame out of plastic. The two primary choices for the rods on the frame were polypropylene and polycarbonate. Table 10 depicts the differences between polypropylene, polycarbonate, and aluminum. Both polycarbonate and polypropylene have been used on other existing ROVs. At first, it seemed like polycarbonate was going to be used because it was a strong material that could survive at the depths of a deep sea oil platform. This material was suggested to the design team by the materials professor at Worcester Polytechnic Institute, Satya Shivkumar. This material seemed like a good choice for the ROV since it met the desired specifications and although it was more expensive than aluminum, it was still relatively cheap.

Table 10: Material comparison for the ROV frame

<table>
<thead>
<tr>
<th>Material</th>
<th>Density</th>
<th>Tensile Strength</th>
<th>Compressive Strength</th>
<th>Elastic Modulus</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aluminum</td>
<td>150-181 lb/ft³</td>
<td>8.41-79.8 ksi</td>
<td>4.35-72.5 ksi</td>
<td>9.85-11.9 10⁶psi</td>
<td>1' rod: 53.34/12'(l)</td>
</tr>
<tr>
<td>Polycarbonate</td>
<td>71.2-75.5 lb/ft³</td>
<td>8.7-10.5 ksi</td>
<td>10-12.6 ksi</td>
<td>0.29-0.354 10⁶psi</td>
<td>1' rod: 51.12'(l)</td>
</tr>
<tr>
<td>Polypropylene</td>
<td>55.6-58.8 lb/ft³</td>
<td>0-6 ksi</td>
<td>3.56-8.01 ksi</td>
<td>0.13-0.225 10⁶psi</td>
<td>1' rod: 53.6/12'(l)</td>
</tr>
</tbody>
</table>

After some further research the design team decided to use polypropylene for the rods that made up the frame. Polypropylene had similar properties to polycarbonate since they are both thermoplastics and have similar melting temperatures and therefore use similar manufacturing processes. Polypropylene is a vinyl polymer with a methyl group attached to it while polycarbonate has a polycarbonate group attached to bisphenol A. Their structures can be seen in Figure 55. Polypropylene was not as strong as polycarbonate because it lacks aromatic rings in its structure, but it was still strong enough to meet the requirements for the ROVEN Sealion, and after further analysis it was shown that the stress from the weight of the ROV was less than half of the yield strength of polypropylene. Polypropylene was also cheaper than polycarbonate with a price comparable to aluminum. Polypropylene was lightweight and it had a density that is slightly less than water allowing it to add to the buoyancy of the ROV rather than take away from it [17].
The polypropylene rods were held together using PVC elbows and PVC slip T's. PVC is a stronger material than polypropylene so it was sturdy enough to hold the polypropylene in place. As a safety precaution and to prevent the elbows from slipping off, the rods were screwed into the PVC fittings. Four steel trusses that spanned between the ends of the ROV and the floatation supports were added to help stabilize the frame. They were attached at an angle from the steel rods that held the thruster to the vertical rods of the frame.

3.4.2 Floatation

The floatation was the main material that determined the buoyancy of the ROVEN Sealion; its properties determined whether or not the ROV would sink or float in the water. Rather than using air-filled containers or buoyancy altering ballasts, the design team decided to use a closed-cell rigid polymer foam. Since syntactic foam was much too expensive to buy, the team decided to use polyurethane foam for the proof of concept model.

The polyurethane foam used for the ROVEN Sealion was made from a two-part liquid expanding foam purchased from US Composites [25]. This liquid pour foam made it easier to create the desired shape of the floatation that is shown in Appendix A, drawing number 004. The group decided to use 4-lb density foam rather than 2-lb density foam (which is usually used for marine floatation applications) because 4-lb density foam can be used for structural applications and can withstand stronger loads. This foam had a parallel compressive strength of 90 psi, a tensile strength of 110 psi, shear strength of 70 psi, and a flexural strength of 120 psi. The foam offered 58 lbs of buoyancy per cubic foot and expanded up to 15 times its liquid volume [25].

The floatation was made by constructing a mold out of corrugated fiberboard and then pouring the foam mixture in layers until it expanded to its full size. The resulting shape was
smoothed out by sanding it and adding more pour foam where needed. Polycrylic paint was coated onto the floatation to add extra protection from water getting to the foam and to make the foam smoother. Once the foam was completed, it was painted yellow for aesthetic reasons and to make the ROV easier to spot when submerged. Figure 56 depicts the steps that went into the production of the floatation.

![Figure 56: Floatation making process](image1.jpg)

### 3.5 Buoyancy

In Chapter 2, it was discussed that the ideal buoyancy for an observational ROV was slightly positive to aid in retrieval if control was lost. The buoyancy is based on the amount of water an object displaces and the weight of that object. In order to make the ROV nearly neutrally buoyant, it had to be designed so that it displaced approximately the same weight of water as its own weight. The volume and weight (or density) of an object and the density of water must be known to calculate the buoyancy that the object has. Because the ROV was made up of many different components with different densities, the team had to be sure that the total
buoyancy was correct. This involved adding the immersed weight of each component together after the ROV was designed. When a design had been finalized, the team made a table to find the total immersed weight of the assembly, without the floatation. This allowed the team to figure out what volume of floatation was needed to make the ROV positively buoyant. The team also decided upon an extra payload of at least 40 lbs in case the floatation did not provide as much buoyant force as advertised or if the team decided to add extra, negatively buoyant components.

The buoyancy of the ROV was calculated using the following equations:

$$\sum_{i=1}^{n} N_i \left( V_i \left( \rho_i - \rho_{H_2O} \right) \right) = W_{IT}$$

$$\sum_{i=1}^{n} N_i \left( \left( W_i - V_i \rho_{H_2O} \right) \right) = W_{IT}$$

Each type of component is assigned a number with \( n \) being the highest numbered component. The quantity of each component is \( N \), which is multiplied by the difference between the object’s actual weight and the weight of the water displaced. This was found in two different ways based on what information was available to the team. In the first equation, the volume, \( V \), and density, \( \rho \), of the object is known. The volume is multiplied by the difference between the component density and the water density. The result is the immersed weight of the component. In the second equation, the weight and volume of the component is known. This equation is essentially the same as the previous one, but with the volume distributed through the densities. The total immersed weight, \( W_{IT} \) of the ROV is the sum of all of the individual components’ immersed weights. If this total is negative, it means that the object is positively buoyant. Table 11 shows each component, its quantity, density/weight, its volume, and its immersed weight. The total immersed weight is also shown. The density of water used in these calculations was taken to be 0.036 lbs/in\(^3\).
In this table, the materials of the components were listed where relevant. In the cases where the density was not known, the spaces were left blank. SolidWorks was used to obtain the densities, weights, and volumes of the components except where there was no SolidWorks approximation, for example, the battery and the thrusters. The team calculated that the polypropylene rods were actually positively buoyant which allowed for the addition of a small amount of extra payload. The payload required to bring the ROV to its ideal buoyancy was calculated to be 47 lbs which brought the entire calculated weight to approximately 122 lbs and the immersed weight to approximately -3 lbs. This is the net force pushing the ROV upwards when the center thruster is not in use. By looking at the graph in Figure 75, the team was able to determine that the ROV would rise at 0.5 ft/s without input from the thrusters.

When conducting the physical test of the ROV in Lake Quinsigamond, the team found that these calculations were not entirely accurate. These calculations allowed the team to prepare a payload close to the one that was required; however, the ROV required approximately 70 lbs of
additional payload in order to submerge. This may have been caused by the piecewise construction of the floatation and made it less dense than expected.

3.6 Force Analysis

By understanding what forces were applied to the ROVEN Sealion, the design team was able to understand how each component reacted to the loads that were being applied to them. The loads that were being applied mostly came from the weight of the ROV. These forces from the motors caused reaction forces in most of the parts. Setting the sum of the forces and moments in the x and y directions to zero helped the design team to decide on where to put the motors and support rods such that the ROV would be most stable.

3.6.1 Frame

The forces on the frame were analyzed by looking at a single point of the frame and summing up the moments in the x and y directions. Figure 57 shows the moments in the y direction. This helped the group to find out how high up to put the motors on the side of the ROV. Figure 58 shows the moments in the x direction for the frame. This aided the group in deciding how far apart to put the vertical polypropylene rods which were used for supporting the frame.

![Figure 57: Forces and moments of the frame in the y direction](image)
\[ \sum M_y = F_{motor}y - R_{bar}19.3 = 0 \]

F<sub>motor</sub> was the force of the thruster and was assumed to be 30 lbs. This was the maximum thrust that the thrusters were advertised to produce. The reaction force on the polypropylene rod, R<sub>bar</sub> could be calculated by using the equation \( R_{bar} = \frac{F_{motor}y}{19.3} \) where y is a distance that could be adjusted for the motor. If the motor was put 9.65 inches below the top of the ROV, then the reaction force would be 15 lbs.

\[ \sum M_x, upper = R_{bar1}x_2 + R_{bar2}x_3 + R_{bar3}31.3 - F_{thrust}31.3 = 0 \]
\[ \sum M_x, lower = R_{bar1}x_2 - F_{w,battery}x_1 + R_{bar2}x_3 + R_{bar3}31.3 - F_{thrust}31.3 = 0 \]

The force of the thruster, F<sub>thrust</sub> was assumed to be 30 lbs. The weight of the battery, F<sub>w,battery</sub> was assumed to be 12 lbs. The force of the thruster was distributed along the horizontal polypropylene rods affecting each of the reaction forces of the vertical rods. The reaction forces
R_{bar1}, R_{bar2}, and R_{bar3} were dependent on the force of the thruster and the weight of the battery. Once these were calculated, one could determine what settings would cause the ROV to reach equilibrium.

### 3.6.2 Floatation

Figure 59 depicts the free body diagram for the floatation. This shows the forces in the x and y direction that affect the floatation. This helped the design team to know if the floatation would be able to withstand the forces from the center thruster as well as from the weight of the ROV and its drag.

\[ \sum F_y = R_{1y} + R_{2y} + F_{buoyancy} + R_{3y} + R_{4y} - F_{thrust} - F_{weight} - F_{drag} = 0 \]
\[ \sum F_x = R_{1x} + R_{2x} + R_{3x} + R_{4x} - F_{drag} = 0 \]

The force of the thruster was assumed to be 30 lbs. The weight of the entire ROV, F_{weight} was estimated to be 122 lbs. The drag force, F_{drag} was calculated in section 3.10.2 and Table 12 shows that the drag on the floatation could be approximated at about 12 lbs. The reaction force R_{4y} was the same as R_{bar3}. The force of buoyancy from the floatation, F_{buoyancy} was calculated to be 80 lbs as seen in Table 11 of section 3.5 Buoyancy. The weight of the ROV was a distributed
force that caused reaction forces at points where the floatation was supported by the steel threaded rods and the polypropylene rods.

3.6.3 Trusses

The steel trusses were added to the ROV to provide additional support to the frame and steel rods which supported the center motor. Figure 60 shows what kinds of forces were applied to each truss. From this analysis, the group was able to determine if the trusses were able to provide enough support to the ROV without failure. Additionally this showed the group what angle would be best for the trusses to provide the most support.

\[
\sum F_x = F_{T1} \cos \theta - F_{T2} \sin \theta = 0
\]

\[
\sum F_y = F_{T1} \sin \theta + R_1 + R_2 - F_{T2} \cos \theta - F_w - F_{motor} - F_{applied} = 0
\]

The force of the motor, \( F_{motor} \) was again assumed to be 30 lbs. The weight of the truss estimated to be 0.75 lbs. \( F_{T1} \) and \( F_{T2} \) represented the force of tension on the truss. This provides enough information to calculate what angle (\( \theta \)) the truss would need to be to create these tensional forces. The optimum angle to have the truss positioned would be 45°. \( F_{applied} \) represents the force that is applied by the truss to the polypropylene rod which it is attached to.
The weight of the truss and the force of the thruster were distributed forces that caused reaction forces at points where the truss was supported by the steel threaded rods and the polypropylene rods.

### 3.7 Using Finite Element Analysis

The finite element method (FEM) is used in order to approximate a solution to boundary value problems. The main theory behind FEM is dividing a component into small sections (finite elements) to which simple equations can be applied. These equations approximate a more complex overall equation for the whole component. This process is similar to approximating a curve with the use of straight lines as seen in basic calculus [26].

The application of FEM is Finite Element Analysis (FEA). Thus the process is simply dividing complex analysis into simple analysis for small elements. FEA was accomplished with the use of SolidWorks. In order to divide the component the software uses mesh generation. The mesh generation process creates polygonal or polyhedral meshes that will approximate the required geometry, simplifying the analysis through the use of simpler shapes and equations. The program contains an FEM algorithm and thus can solve complicated domains.

In this project FEA was used in order to approximate von Mises Stresses, axial stresses and displacement on individual components and the entire frame.

### 3.8 Stress Analysis

Von Mises stresses were computed using SolidWorks. Von Mises stresses were necessary in order to predict if components would deform plastically. The material starts yielding when the von Mises stress reaches the yield strength ($\sigma_y$) of the material. The materials tested were the ones supporting the most loads, and were essential for the functionality of the ROV.

In this project, von Mises stresses were used to determine at which points the materials will fail along the x, y, and z axes. Principle stresses are assigned to these axes. Figure 61 shows Principle stress coordinates for a cylinder. The loadings used in von Mises stresses are results of simple uniaxial tensile tests. Von Mises stresses are independent of the first stress invariant and thus it can be applied to ductile materials to analyze their plastic deformation. The loads applied to the materials were predetermined through experiments, material properties, and force analysis [27].
Thus by assigning principle stresses (replacing the 1, 2, 3 coordinate system as seen in Figure 61 with an x, y, z coordinate system) in $\sigma_{xx}$, $\sigma_{yy}$, and $\sigma_{zz}$, as well as the pressures in x with respect to y, y in respect to z, and z in respect to x, one can solve for von Mises stresses with the following equation:

$$\sigma_{vM} = \sqrt{\frac{1}{2}\left[\left(\sigma_{xx} - \sigma_{yy}\right)^2 + \left(\sigma_{yy} - \sigma_{zz}\right)^2 + \left(\sigma_{zz} - \sigma_{xx}\right)^2\right] + 3\left(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2\right)}$$

The main stresses are applied in the same direction as the axis, thus the right side of the equation can be ignored.

### 3.9 Applied Forces

The shear direction had the greatest possibility of failure. The force applied to calculate the von Mises Stress in the y-direction for the top rod embedded in the floatation was the weight of the components under water. It was assumed that the ROV would be going at constant
velocity for most of the travel, thus acceleration was neglected. At constant velocity the drag force is equal to the thrust given by the top motor.

The first bar analyzed was the top, front polypropylene rod that supports most of the ROV’s weight. Thus the total force on the rods was equal to that of the components weight. The weights can be found in Table 11; the immersed weights were used to replicate a more accurate load.

3.10 Component Analysis

The following four sections discuss the analyses performed on the components that are subjected to the greatest forces when the ROV is active. These analyses help predict if these materials will plastically deform or collapse under the loads applied by the weight of the components and the thrust of the motor. Finite element analysis was used to estimate the von Mises stresses at the maximum load that each component would experience with a safety factor of at least 2.

3.10.1 Polypropylene Rod Analysis

The first components analyzed were the polypropylene rods to determine whether or not they would break or plastically deform under the loads the ROV would experience. The bar chosen for this study was the top, front polypropylene rod since it experiences the most stress. The top rods had to withstand a maximum load of 80.16 lbs, which is the weight of the relevant components ROV. Figure 62 shows the von Mises stresses on the bar. The maximum stress is located at the middle of the bar, which is expected since the forces are applied from the sides and the floatation pulls the bar up. The maximum stress experienced is 1,252.5 psi. This is less than half the minimum yield strength of polypropylene, which is 3000 psi. The load applied during this analysis didn’t take into account additional support added by the team; thus it can be safely assumed that the polypropylene rods can withstand the weight of the ROV.
Figure 62: von Mises stresses on top polypropylene rod

The other analysis of interest is displacement of the rods. If too much displacement occurs under these loads, it can dislocate joints and move components. Figure 63 shows the displacement of the polypropylene rod. Due to its properties, polypropylene is able to absorb shocks and vibrations due to its flexible nature. The maximum displacement expected is approximately 0.34 inches. This displacement is minor and can be ignored.
3.10.2 Threaded Rod Analysis

The threaded rods support the motors and are therefore essential for the ROV. Figure 64 shows the von Mises stresses experienced by the threaded rods through the weight of the motors and thrust. The thrust was assumed to be 30 lbs, since the trolling motors are advertised to give a maximum of 30 lbs of thrust. The total force on the top threaded rods is 37.21 lbs, which is the combined weight and maximum thrust on those components. The maximum stress experienced by the threaded rod is 3461.2 psi. These rods are steel and have minimum yield strength of 29579.5 psi, which is eight times the amount of stress the rod will be experiencing.

Figure 65 is the amount of displacement that the rod would experience at this load. The maximum amount of displacement is 0.0086 inches, proving to be insignificant for the purposes of this project.
Figure 64: von Mises stresses experienced by threaded rods

Figure 65: Displacement threaded rod would experience at max load
3.10.3 Rail Analysis

Figure 66 shows the von Mises stresses on the steel rail used to support the battery and the dome. The largest load applied to these rails was the battery which had a weight of 12 lbs. Since the design is made to be able to withstand extra components the applied loads for calculations was 36 lbs. This is 3 times the weight of the battery; this gives a large safety factor and also makes sure the rail can be used for supporting the ballast or extra batteries if future teams need them. With this load the maximum stress experienced is around 15000psi, which is only half the yield strength of the steel rail.

![Figure 66: von Mises stresses on rail](image)

Displacement on these rails would cause the battery to detach so it’s essential to keep it at a minimum. The maximum deformation the rail experiences is .03198 inches as seen in Figure 67.
3.10.4 Whole Frame

Similar analysis as those described above were done to the main basic frame of the ROV. Figure 68 shows the displacement the frame would experience under maximum loads. Again, these analyses are done without taking into consideration the additional support added to all of these components in order to provide a larger safety factor.
The loads applied to the frame were distributed among 8 points in order to simulate the actual weight applied to the real structure. The net load applied to the frame was 78.31 lbs, which is the estimated weight on these sections of the frame.

Figure 69 shows the axial stresses on the whole frame per beam. The axial stress is a tension or compression stress that has been applied by an axial load. Since it is calculating uniaxial stress, it is uniform throughout the entire cross section. The loads were applied at the 8 joints of the frame. Notice that the frame is fixed at the top middle where the floatation would be holding the frame. The maximum axial stress is 14 psi, which is minimal stress, thus the materials used in the design are more than capable of withstanding the force. Axial stress is calculated using the following equation, where P is the axial force, and A is the cross-sectional area [28].

\[
\sigma_{\text{axial}} = \frac{P}{A}
\]

Figure 69: Axial stresses on frame by beam
3.10.5 Strength vs. Weight and Buoyancy

The reason certain materials were chosen over others, even though their strength is significantly less, is due to the fact that they are very dense, and would thus sink the ROV. Polypropylene is buoyant and helps to keep the ROV afloat, and can withstand large amount of stress, fatigue, and can absorb vibrations and impact. Steel was used for additional support because of its accessibility and low cost. Even though steel has high yield strength it’s too dense to use for the entire frame, so its inclusion in the ROV design was minimized.

3.11 Flow Analysis

Analysis of the fluid flow around the surfaces of the ROVEN Sealion was important in determining its speed and maneuverability. The motion of fluid has many different properties and effects that can create a number of forces that can assist in determining the shape, the materials used, and the amount of power required to move the ROV. Drag forces are a major factor in determining the shape and power consumption of the ROV. Pressure forces must also be taken into account when choosing materials and creating enclosures.

3.11.1 Initial Flow Tests

Although case studies of ROVs in the industry yielded some information about an ideal shape for an observational ROV, the drag forces and lift forces that flow around the shape of the ROVs were difficult to determine using video footage and general statistics. For work class and observational class ROVs, there appeared to be 3 basic shapes. In this section, the shapes being referred to are the cross-sectional profiles of the ROVs when viewed from the side. The first shape considered was a trapezoid. The shape is a trapezoid in which the rear end of the ROV is perpendicular to the top and bottom and the front end is at a slope such that the top of the ROV extends out farther than the bottom. This shape is more common among work class ROVs that may have robotic manipulators, cameras, and lights on the front. The arms are positioned lower on the ROV which brings them back to the center of gravity and brings the manipulating end closer to the cameras which are located towards the top. The second shape considered was similar to the trapezoid shape, but was better approximated as a letter box style pentagon with the point located mid-way between the top and the bottom on the front of the ROV. This shape
is common among observational ROVs because the camera is located at the central point of the front and therefore affords the camera the widest field of view in the up and down directions. The third shape considered was that of an extended hexagon, similar to the pentagon shape except with a point located at both ends of the ROV. This shape was common in ROVs that needed to travel in both forward and reverse.

Simple 3D models of these shapes were created in SolidWorks; their shapes and dimensions can be seen in Figure 70.

![Figure 70: Initial flow analysis shapes](image)

Each of these shapes was then used in a flow analysis using SolidWorks Flow Simulation. The flow was set to model the shape as moving forward through water at a speed of 4 ft/s. This speed was chosen based on the maximum speed observed in the case studies. Through these flow simulations the team was able to choose a shape best suited to the objective.

When moving in the forward direction, the trapezoidal shape generated a considerable amount of lift due to the sloping front of the model. This was deemed unsuitable for the ROVs design because a lift force could cause the ROV to be unstable as it approached higher speeds. An ROV of this shape, without the proper thruster placement and center of gravity location could end up becoming inverted or rising in the water as it moved forward, which could impact the amount of control the operator had over the ROV. The pentagonal model, because it was
symmetric in the XY-plane, did not generate any lift in the forward direction. The square back of the model did create four vortices at the rear; however the vortices allowed some of the flow to stay laminar as it left the top and bottom surfaces without creating friction drag on the rear surface of the model. Like the pentagonal model, the hexagonal model also did not generate lift. However, it created a larger amount of drag than the pentagonal model due to the point at the back. This point created a poor hydrofoil shape and bought the laminar flow of the water closer the rear surfaces, which later caused the generation of two large vortices behind the rear point. This laminar flow along the rear surfaces also increased the amount of friction drag, caused by the interaction between the viscous flow and the surfaces.

The pentagon shape was chosen as the ideal shape for the ROV. It did not generate any unwanted forces, thus increasing controllability and it had the least amount of drag. The reverse direction was not considered as important because the ROV would only need to travel at its top speed in the forward direction in order to arrive at its point of operation. The reverse direction would only be needed in adjusting the ROVs position relative to the structure it was working on. Unfortunately, as the team progressed with the design, it became apparent that the ROV would assume a more open shape as opposed to the solid objects of the initial models. This required a second generation of flow tests in which the final model of the ROV was used.

3.11.2 Final Flow Tests

The final design of the ROVEN Sealion, due to the open frame aspect, would be difficult to model accurately by equations. The number of simplifications would make the pursuit of the drag and lift forces generated by the shapes fruitless. Instead, like the model shapes before, the final SolidWorks model of the ROV was used in the SolidWorks Flow Simulation add-on. Because the final design had already been chosen and built, the goal of these simulations was to determine the drag forces on the ROV as it moved in different directions. From these drag forces, the team could calculate the stresses on the different supports of the ROV as well as estimate its top speed in the four directions.

The final model of the ROV contained many components and many surfaces. Some of the components had to be suppressed in order to run the flow simulation because of the limited computing power available. The smallest components were suppressed, as well as the components that were not impacting the flow. These components included: the elbows, the slip
T’s, the screws, the nuts and bolts, the rails, the pipe clamps on the thrusters, and the threaded rods. This left the most important components: the floatation, the polypropylene rods, the camera dome, the thrusters, the battery and the waterproof boxes.

The flow simulations were then performed in each of the four directions: up, down, forwards, and reverse. In the forwards and reverse directions, a velocity of 4 ft/s was used. In the up and down directions, a velocity of 1.5 ft/s was used. The goals set for the simulations were the drag forces and the lift forces. These goals allowed the team to get an idea of the power required to reach these speeds and also see if the ROV would have unstable tendencies.

Table 12 shows the drag and lift forces measured by the program in each direction, as well as the drag in the forward direction if there was no floatation. The reason for this is because the team suspected that the floatation would generate a larger amount of drag on the top of the ROV which could cause the ROV to tilt upwards as it moved forward as if a lifting force was being created.

<table>
<thead>
<tr>
<th>Movement</th>
<th>Speed (ft/s)</th>
<th>Drag Opposing Movement Direction (lbs)</th>
<th>&quot;Lift&quot; (lbs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forwards</td>
<td>4</td>
<td>34.3</td>
<td>3.98</td>
</tr>
<tr>
<td>Forwards (no flotation)</td>
<td>4</td>
<td>22.9</td>
<td>0.97</td>
</tr>
<tr>
<td>Backwards</td>
<td>4</td>
<td>28.27</td>
<td>7.25</td>
</tr>
<tr>
<td>Up</td>
<td>1.5</td>
<td>26.97</td>
<td>1.32</td>
</tr>
<tr>
<td>Down</td>
<td>1.5</td>
<td>11.29</td>
<td>0.73</td>
</tr>
</tbody>
</table>

From this table it is apparent that the floatation creates approximately 12 lbs of drag focused on the top part of the ROV frame. This means that when the ROV moves forward it will experience a torque which, if not accounted for, could cause the front end of the ROV to rise higher than the rear end. However, the team determined that the weight on the lower part of the ROV would be sufficient to counteract this torque. This was also apparent in the physical test of the ROV, which will be described further in 3.19.1 Flow Observations. Similarly, the lifting forces are also very small compared to the drag force and could easily be counter-acted by placing the thrusters in the right position.

The forward direction has the greatest amount of drag force at the ideal maximum speed. Some of the causes of this can be seen in Figure 71.
The multicolored bands show the path of the water as it flows around the surfaces of the ROV. The colors in the bands correspond to different pressures as seen in the small chart in the figure.

The reverse direction has a smaller amount of drag than the forward direction. The flow in reverse can be seen in Figure 72.
This is likely due to the camera mounting plate being located in the back. This reduces the number of vortices formed within the ROV frame. The lower drag in reverse is convenient, because the design of the thruster’s propeller does not provide as much thrust in the reverse direction as it does in the forward direction.

The upwards direction has the greatest amount of drag because in this direction, the ROV can be approximated as a flat plane moving perpendicular to the flow. This approximation has a high form-drag coefficient. From Figure 73 the dramatic flow around the floatation is clear.

As seen in the picture, the flow also shows many vortices forming within the ROV frame. The downwards direction has a lower drag than the upwards direction. In Figure 74 the flow moving up through the center thruster port can be seen.
The reduced drag may have something to do with the thrusters, battery, and rods being between the flow and the floatation. These objects likely introduce turbulence in the flow, allowing the flow to separate more easily when it reaches the flat plane of the floatation.

By knowing these drag forces and the speeds they were measured at, the team was able to derive the drag coefficient and frontal area of the ROV in each direction. By obtaining these values, the team was able to predict the velocity of the ROV at varying amounts of power from the thrusters. The team used the standard drag equation and the flow was assumed to have a sufficient Reynolds number to perform the calculations.

The equation below shows the standard drag equation where $\rho$ is the density of water at 0.036 lbs/in$^3$ and $v$ is the free-stream velocity in ft/s. $(C_D A)$ as a whole is taken as the unknown and is constant.

$$F_D = \left(\frac{1}{2}\right)\rho v^2 C_D A$$

This equation can be re-arranged to find $(C_D A)$. The frontal area, $A$, is difficult to calculate when working with complex shapes and flows.

$$(C_D A) = \frac{2F_D}{(\rho v^2)}$$

The $(C_D A)$ coefficient of each flow direction can be seen in Table 13.
The nature of the drag equation is such that the drag force increases as the speed of the flow increases. This means that there is a point at which the ROV stops accelerating and reaches its maximum velocity at a given thrust.

\[ F_{D_{\text{max}}} = F_{\text{Thrust}} \]

In order to find the maximum velocity, the drag equation can be re-arranged to find \( V \). The drag coefficient and the frontal area used to find the velocities in each direction were obtained from the earlier form of the drag equation and Table 13.

\[ V_{\text{max}} = \sqrt{2F_{\text{thrust}}/\rho(C_D A)} \]

The velocity for each direction was then plotted (shown in Figure 75) over a range of thrust from the trolling motors mounted on the ROV. The advertised thrust of 30 lbs per motor was used as the maximum expected thrust. The forward and reverse directions, because they used 2 motors, was plotted up to 60 lbs and the up and down directions were plotted up to 30 lbs because one motor is used in those cases.

<table>
<thead>
<tr>
<th>Movement</th>
<th>Speed (ft/s)</th>
<th>(Cd*A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forwards</td>
<td>4</td>
<td>0.068709936</td>
</tr>
<tr>
<td>Forwards (no flotation)</td>
<td>4</td>
<td>0.045873397</td>
</tr>
<tr>
<td>Backwards</td>
<td>4</td>
<td>0.056630609</td>
</tr>
<tr>
<td>Up</td>
<td>1.5</td>
<td>0.384188034</td>
</tr>
<tr>
<td>Down</td>
<td>1.5</td>
<td>0.160826211</td>
</tr>
</tbody>
</table>
Figure 75: Plot of Thrust vs. Velocity for each movement direction

Here it is shown that the ROV can theoretically reach a higher speed in reverse than in the forward direction, however, as discussed in the thruster experiments section, the thrusters do not generate as much thrust in the reverse direction. Also from the thruster experiments it was found that the thrusters may only generate a maximum of 13 lbs of thrust in the forward direction. This means that the maximum amount of thrust in the forward direction is approximately 26 lbs, which corresponds to a velocity of 3.5 ft/s on the graph. This is fairly close to the desired maximum velocity of 4 ft/s. As discussed in the thruster experiment section, however, it is likely that the maximum thrust could be much higher.

3.12 Adjustability

The ROV was designed so that its components could be moved. This allows the center of gravity to be changed, which in turn allows new payloads to be added, such as manipulators or additional motors. Figure 76 shows a side view of the ROVEN Sealion. The motors are mounted on vertical polypropylene rods that can slide horizontally along the frame, towards either the front or rear of the ROV. These are secured by drive clamps, thus making it easier for them to be adjusted. Similarly, the PVC slip T’s can move up or down along these vertical rods.
The other vertical polypropylene rod helps support the weight of the top motor and the floatation, and helps in preventing deformation of the frame. This rod can also be used for supporting manipulators or additional payloads. Just like the rods supporting the motor, this rod can slide horizontally along the frame.

The battery mount consists of steel rails which are also easily moveable. Figure 77 shows the battery mount, with the black UB12180 12V 18Ah sealed lead acid battery in the center. These mounts are held in place via U-Bolts attached to the frame, and can be moved around the ROV in order to change the center of mass, or to rearrange components. This mount can hold more than just the battery, and can even be placed on the sides of the ROV depending on the design needs. The camera mount, seen later in Figure 82, uses a similar rail mount mechanism.
3.13 Processor Selection

As discussed in section 2.6.9 Microprocessor, the microprocessor is the brain of the ROV, and controls all of the electronics, receives data from the sensors, and transmits data to the surface. In the case of this project, the microprocessor needed to be able to control three different motors and the lights. The microprocessor also needed to be able to receive data from the sensors. Another basic consideration is the amount of memory that is required to store the code that will be run on the microcontroller. The flow chart seen in Figure 78 shows how the electronics interact with the microprocessor. Each color corresponds with a different type of interaction. The sensors, shown in blue, send their data to the microprocessor. The microprocessor receives this data and sends it to the surface, where the driver can view it. The camera sends its data to the surface directly, as there is no need for it to interact with the microprocessor. The driver can send commands to the microprocessor from the surface. From these commands, the microprocessor controls the three motors and the lights. As can be seen, the microprocessor is the key component in this system, and without a microprocessor that fits, the entire system would not work.
After considering many different microprocessor boards, it was decided that the Arduino Uno was the best fit. Some of the other possibilities include the MSP430 series of microcontrollers, and Stellaris development boards, as shown in Figure 79. The MSP430 series can be run at very low power, and there are a very large number of these microprocessors to choose from. The Stellaris development boards are impressive and provide a lot of functionality, but are costly when compared to the other microcontrollers that were considered.

Another advantage of using an Arduino microcontroller is the amount of documentation available. The Arduino is open source, which means that the architecture of the microprocessor...
is freely available, and anyone with access to the technology can manufacture it. There is a large number of user made libraries, which include functions that have been previously created to perform specific tasks. There are also many examples of code that use these libraries, which helps to understand exactly how these functions work. Using these libraries makes programming the Arduino very easy. In addition to the software support, there are a number of Arduino Shields which have been created. These shields are add-ons to the Arduino which provide functionality very easily. Some examples of shields are the wireless communication shield, servo motor shield, and the Ethernet shield. Each of these shields is made to perform a specific job, and interface directly with the Arduino by placing the shield on top of the Arduino, connecting the correct pins. These shields can be very useful for performing specific tasks very easily.

The Arduino Uno has 6 analog input pins. Four of these pins were used in the ROV. Two were used for the differential output of the pressure sensor, and two were used for the output of the magnetic compass. The Arduino Uno also has 14 digital pins. These pins can be used to read data, as well as control devices. Six of these pins are capable of pulse width modulation, as described in 3.17 Motor Control. If more PWM pins are required, it is possible to use any of the digital pins as PWM by using code to emulate a PWM output. For this project, four PWM pins were used to control each of the three motors and the lights. The temperature sensor was also connected to a digital pin. This leaves quite a bit of room for expansion among the digital ports, but not so much among the analog ports [29].

![Figure 80: Arduino Uno board](image_url)
While an Arduino Uno was chosen for this project, there are several other Arduino boards that could also work. The Uno is the most basic board, and as such costs the least out of all of the microcontroller boards that were considered. Its versatility, support, and cost were ultimately the factors that lead to the choice of using the Arduino Uno.

The microprocessor was contained inside of a watertight 1050 Pelican micro case. Holes were drilled in the side of the case to allow for the USB cable and the wires to the speed controllers to leave the case. These holes were sealed with silicon. The compass, pressure sensors, and the speed controllers were contained in a similar fashion inside of the 1050 case and an additional 1040 case.

3.14 Light Selection

The lights that were used were 3-watt LEDs. The brand that was chosen was the CREE MR16 because it provides 500 lumens and functions with 12 volts of DC power. The LEDs are also dimmable. The LEDs were tested in a local lake before the camera was selected. As a substitute camera, a basic handheld camera with a video function was used and placed within a water tight case that had a clear top. An LED was secured on the side of the case so that the camera, which was pointing toward the clear top, was parallel with the light. This setup was placed in the water and tied to a handle. It was moved to various locations under the surface throughout the test. The camera light setup was then taken out of the water when enough footage was acquired and the video from the camera was taken out. Unfortunately, not much could be seen in the video footage after looking at the video. This was because the voltage used with the lights in this experiment was lower than the normal operating voltage. The light was also pointed parallel to the direction of the camera and did not intersect with its line of sight. This reinforced the background research from section 2.6.4 Lights. The experiment did show how well the lights work underwater and how bright they can be. In the final ROV, two LEDs were used and were angled toward the line of sight of the camera. This set up as well as the set up for the experiment can be seen in Figure 81.
3.15 Sensor Selection

Section 3.14 discusses the different sensors used on the ROV. These were extra payload options that the team obtained in order to monitor the conditions that the ROV operated in.

3.15.1 Camera Selection

There were several aspects to consider when selecting a camera for the ROVEN Sealion. The most important requirements for such a camera are outlined below:

- Color video
- Live-stream video capability
- Smaller than 8 in$^3$ to fit in encasing
- Under $100 (preferably less) to fit within budget
- USB connection (preferred)

A possible option would be a GoPro camera which has good quality, but does not have live-stream footage and is rather expensive. Another possible camera was a CCD or CMOS board camera which does not have live-stream and is difficult to incorporate as it is not ready to work and needs to be built into an electrical board. The camera that was chosen was a basic webcam. As seen in Table 14, the Logitech C210 webcam is small, inexpensive, and has live-stream footage capability. The webcam uses a USB connection to directly connect to a laptop computer.
Table 14: Specifications for the Logitech C210 webcam

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions</td>
<td>6 x 3 x 8.2 inches</td>
</tr>
<tr>
<td>Weight</td>
<td>8 oz</td>
</tr>
<tr>
<td>Resolution</td>
<td>360p</td>
</tr>
<tr>
<td>Connection type</td>
<td>USB</td>
</tr>
<tr>
<td>Cost</td>
<td>$25 (approx.)</td>
</tr>
</tbody>
</table>

The camera was mounted onto a flat polypropylene surface. It was mounted in the center of the polypropylene with no possibility of movement. The camera was then covered with a clear plastic, 8-inch diameter dome [30]. The camera and the dome were sealed to the polypropylene surface with silicon. The polypropylene surface was grinded so that the USB cable from the camera could fit under the dome without going through the plastic surface. This mounting setup for the camera was then attached to steel rails, as seen in Figure 82; this entire assembly was mounted onto the front of the ROV using U-Bolts.

Figure 82: Camera mount assembly
3.15.2 Pressure Sensor

As discussed in section 2.6.5.1 Pressure Sensors, pressure sensors are important for determining depth in underwater environments. Having pressure information also allows the operator to avoid submerging the ROV (or other device) beyond its rated pressure limits.

The speed of the ROV can be estimated by positioning two pressure sensors perpendicular to one another. This would measure the speed of the flow of water moving past the ROV similar to how a pitot tube measures the speed of the flow of air on an airplane. One pressure sensor measures the pressure of the water opposite to the direction of the flow and the other sensor measures the water pressure perpendicular to the flow. The sensor measuring the pressure perpendicular to the direction of the flow measures the static pressure of the water. The static pressure is based on the depth of the water according to the equation:

\[ P_{\text{static}} = \rho g h \]

Where \( \rho \) is the density of the water, \( g \) is the acceleration due to gravity, and \( h \) is the depth of the water. The sensor measuring the pressure into the flow direction measures the static pressure in addition to the dynamic pressure. This total pressure is known as the stagnation pressure. The dynamic pressure is based on the velocity of the flow according to the equation:

\[ P_{\text{dynamic}} = \left(\frac{1}{2}\right) \rho v^2 \]

Here, \( v \) is the velocity of the flow of water. The dynamic pressure can be found by subtracting the static pressure from the stagnation pressure.

\[ P_{\text{stagnation}} - P_{\text{static}} = P_{\text{dynamic}} \]

Once the dynamic pressure is known, the equation can be re-arranged to find the velocity of the fluid flow which, if measured in the free-stream, should provide an accurate reading of the ROV’s speed.

\[ V = \sqrt{\frac{2(P_{\text{stagnation}} - P_{\text{static}})}{\rho_{H_2O}}} \]

When investigating pressure sensors that could potentially be integrated into the ROVEN Sealion sensor system, some issues were encountered. The most prominent of these issues was with finding sensors rated for pressures higher than 1 atmosphere (pressure at sea-level) that still
fit within the project budget. The majority of inexpensive, ready-to-use pressure sensors are indented for use in aviation, at pressures less than 1 atmosphere.

Another issue was determining how to submerge the sensors if they were not waterproof. Several solutions were designed, including using a long, thin, vinyl tube, which is open at one end and is of the same diameter of the pressure sensor’s tube. The vinyl tube would be positioned such that the air contained in the tube would not allow the water to enter. This has the benefit of allowing the pressure sensors to be contained in an area separate from the point at which the pressure is measured. Another solution would be to place the pressure sensors in a sealed compartment, such as a large cylinder. This cylinder would be sealed on one end with an elastic membrane, such as a balloon. The balloon will stretch into the cylinder as the water pressure increases and thus transfer this pressure to the air in the cylinder. As the water pressure decreases the air pressure in the cylinder would also decrease. This would require calibrating the pressure sensors to water pressure. The elasticity of the balloon exerts a reaction force and therefore does not transfer the pressure to the air exactly. Due to the equations involved in calculating the forces of the balloon, calibrating this would be considerably difficult without actual experimentation.

After this research and preliminary design phase, the pressure sensors determined to best suit the needs of the project were two GE NPC-1220-030A-1L sensors, as seen in Figure 83. The specifications for these NPC-1220 sensors can be seen in Table 15 [31].

Figure 83: The GE NPC-1220 pressure sensor
Table 15: Specifications for the GE NPC-1220 pressure sensor

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Pressure</td>
<td>30 psi</td>
</tr>
<tr>
<td>Operating Temperature Range</td>
<td>-40 to +125 °C</td>
</tr>
<tr>
<td>Pressure Type</td>
<td>Absolute</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.1%</td>
</tr>
<tr>
<td>Output Type</td>
<td>Analog voltage</td>
</tr>
</tbody>
</table>

In order to properly harness the analog voltage output of the NPC-1220 pressure sensor, a differential amplifier circuit was required. This circuit uses two operational amplifier chips to, as the name implies, amplify the voltage difference between the two outputs (reference and measurement values) of the pressure sensor. This amplification is necessary in order to have a signal large enough to be sensed by the Arduino microcontroller's analog inputs. Figure 84 shows the circuit required for an NPC-1210 [32]; though it is a slightly different pressure sensor, the principle that the output needs to be amplified still applies.
Figure 85 shows how to connect the NPC-1220 pressure sensor [31]. While this circuit seems a bit complicated, it isn’t actually that hard to understand. The positive input is connected to a 5-volt rail, and the negative input is connected to ground. This allows the pressure sensor to output the correct voltage which corresponds with the pressure of the environment. When combining this circuit with the previous circuit for amplifying the output, an operational pressure sensor can be built.

![Circuit diagram for NPC-1220 pressure sensor](image)

<table>
<thead>
<tr>
<th>Lead</th>
<th>Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-Out</td>
</tr>
<tr>
<td>2</td>
<td>-In</td>
</tr>
<tr>
<td>3</td>
<td>+Out</td>
</tr>
<tr>
<td>4</td>
<td>+In</td>
</tr>
<tr>
<td>5, 8</td>
<td>Current Set Resistor</td>
</tr>
<tr>
<td>7, 8</td>
<td>No Connection</td>
</tr>
</tbody>
</table>

Figure 85: Circuit for the NPC-1220 pressure sensor
3.15.3 Compass Module

Mentioned in section 2.6.5.2 Orientation Sensors, magnetic compasses are one way to partially determine the orientation of the ROV. The project team preferred using a magnetic compass over other orientation sensors because it still provided enough information for dead reckoning navigation (discussed in 2.6.7 Navigation), while also being low cost. The compass module selected for the ROVEN Sealion was the Honeywell HMC6352, whose specifications are detailed in Table 16 below [33].

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage</td>
<td>2.7 to 5.2 V</td>
</tr>
<tr>
<td>Supply current</td>
<td>1 mA at 3 V</td>
</tr>
<tr>
<td>Measurement type</td>
<td>Magnetic heading</td>
</tr>
<tr>
<td>Measurement resolution</td>
<td>0.5 degree</td>
</tr>
<tr>
<td>Update rate</td>
<td>1 to 20 Hz</td>
</tr>
</tbody>
</table>

This magnetic compass module uses the popular I2C interface; connecting it to the Arduino microcontroller was as simple as having a clock connection (SCL) for synchronization purposes, and a data connection (SDA) to send information to and from the compass. On the Arduino Uno R3, the default configuration for SCL and SDA are pins 5 and 4, respectively. These connections, along with the power (VCC) and ground (GND) connections can be seen in Figure 86.

Figure 86: The Honeywell HMC6352 magnetic compass module
In order for the Arduino microcontroller to communicate with the HMC6352, a small amount of programming in the Arduino software environment was required. With the process outlined below, it was possible to quickly communicate with and collect data from the compass module for transmission to the surface unit from the Arduino.

- Setup
  - Set HMC6352 slave address (0x42)
  - Set HMC6252 read address (0x41)
  - Begin I2C communication

- Loop
  - Send “collect data” command (compass collects data and stores it on-chip)
  - Request and store the collected data
  - Convert collected data to a true heading measurement
  - Send the converted data to the surface unit for display

Unfortunately, there are some drawbacks to using such a simple device for sensing orientation. Most significant among these drawbacks is the possibility of magnetic interference from the nearby Minn Kota Endura C2 trolling motors used as thrusters for the ROV. Brushed motors like these are prone to cause electrical magnetic interference (EMI) that could potentially influence the measurements of the magnetic compass. Additionally, the HMC6352 can only provide reliable heading readings when completely horizontal; measurements taken when tilting the compass module proved that the HMC6352 is inaccurate by many degrees when at non-ideal angles.
3.15.4 Temperature Sensor

As discussed in section 2.6.5.3 Temperature Sensors, a temperature sensor is used to determine if there are any irregularities in the surrounding water, as well as to generally tell what the temperature of the surrounding environment is. In addition, an internal temperature sensor could also tell the driver whether the internal circuitry is becoming too hot to operate.

The most important feature of a temperature sensor is the different range of temperatures that it can sense. The range of temperatures that bodies of water can possibly be is 25 °C to about 0 °C. The temperature varies as the depth increases, as well as with different compositions of minerals in the water. The temperature sensor that was used for this project is able to capture a very wide range of temperatures which greatly exceeds the range of temperatures commonly found in bodies of water.

The temperature sensor that was chosen for this project is the DS18B20; this sensor’s specifications can be seen in Table 17 [34]. This temperature sensor works very well for this project, as the 1-wire interface (seen in Figure 87) is very easy to use with the Arduino Uno. As with the other sensors described in this section, and with all the electronics used in this project, they all had to be waterproofed so they would work correctly in the environment that the ROV would be traveling in. One bonus of the temperature sensor that the team chose to use was that it was a waterproof unit to begin with.

Table 17: Specifications for the DS18B20 temperature sensor

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature Range</td>
<td>-55 °C to +125 °C</td>
</tr>
<tr>
<td>Power Voltage</td>
<td>3.0 V to 5.5 V</td>
</tr>
<tr>
<td>Accuracy</td>
<td>±0.5 °C accuracy between -10 °C to +85 °C</td>
</tr>
<tr>
<td>Output Type</td>
<td>16 bit binary word</td>
</tr>
<tr>
<td>Connection</td>
<td>1-Wire interface</td>
</tr>
</tbody>
</table>
This sensor has one wire for data, and two wires for power and ground. These are connected to the Arduino Uno, which handles the rest of the calculation. The code to obtain readings from the temperature sensor, which is outlined below, can be seen in full in Appendix D.

- **Setup**
  - Create OneWire object for temperature sensor
  - Begin serial communication

- **Loop**
  - Send command to sensor telling it to prepare a temperature reading
  - Read the sensor’s temporary data to get raw temperature reading
  - Combine two bytes of reading into one real temperature value
  - Send data as degrees Celsius to surface unit for display

### 3.16 Electrical Power Supply

When selecting the components that have the most electrical draw, such as the motors, it is important to first consider how power will be supplied, and how much will be needed to meet the requirements of the application. Power is commonly supplied in two forms: over a power cable from a remote power supply or on-board by some kind of battery.
3.16.1 Electrical Power Requirements

Power supplied over a cable, while alleviating the added weight from the battery, adds complication; power deteriorates over distances according to the type of wire and quantity of current. Supplying the average 13 amps required for each Minn Kota Endura C2 motor on the ROV would be inefficient or costly for long distances. With a theoretical peak amperage of approximately 40 amps, and including a safety factor, AWG 4 wire (with a cost of $1 per foot) would be a likely candidate, as shown in Table 18 [35]. Obviously, this method would also require some form of power supply on the surface, detracting from the portability of the ROV.

<table>
<thead>
<tr>
<th>AWG gauge</th>
<th>Diameter (inches)</th>
<th>Max current for power transmission (amperes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.2576</td>
<td>94</td>
</tr>
<tr>
<td>3</td>
<td>0.2294</td>
<td>75</td>
</tr>
<tr>
<td>4</td>
<td>0.2043</td>
<td>60</td>
</tr>
<tr>
<td>5</td>
<td>0.1819</td>
<td>47</td>
</tr>
<tr>
<td>6</td>
<td>0.1620</td>
<td>37</td>
</tr>
</tbody>
</table>

Supplying power via an on-board battery quickly became the ideal choice for the project. Not only does a battery add to the portability of the ROV, it also avoids the issue of power deterioration over long distances. There are, however, a few issues with selecting a battery appropriate for the intended application. Batteries are rated by both their voltage and their Ampere-hour capacity. The project team knew that the battery would have to be 12 volts, as required by the Minn Kota Endura C2 motors. Ampere-hour capacity was more difficult to determine. The project objective was to have an ROV that would be able to operate at peak performance for one hour. To meet this objective, the on-board battery needed to provide power to all electrical components on the ROV, possibly simultaneously. A diagram illustrating the power system of the ROV, with estimated maximum current draws for each component, can be seen in Figure 88.
From this power system, it was possible to calculate the total maximum current draw:

\[
I_{\text{max}} = I_{\text{motors}} + I_{\text{LEDs}} + I_{\text{sensors}}
\]

\[
I_{\text{max}} = 45 + 1.5 + 0.007 \text{ A}
\]

\[
I_{\text{max}} = 46.507 \text{ A}
\]

These calculations were done assuming that each motor was running near its maximum capacity for the majority of the time the ROV was in use. During typical use, however, each motor would be running far less frequently, thereby consuming less power:

\[
I_{\text{typ}} = \left( I_{\text{motors}} \times \frac{1}{3} \right) + I_{\text{LEDs}} + I_{\text{sensors}}
\]

\[
I_{\text{typ}} = 15 + 1.5 + 0.007 \text{ A}
\]

\[
I_{\text{typ}} = 16.507 \text{ A}
\]
With these more reasonable calculations, a final estimate of power consumption for the ROV was found to be approximately 16.5 amps. As seen in Figure 89, in order for the battery to last above its recommended 10V limit for one hour, an ampere-hour capacity of approximately 16 ampere-hours would be required [36].

![Figure 89: 12V battery discharge characteristics](image)

**3.16.2 Battery Selection**

With this 16.5-amp power requirement in mind, batteries rated around 16Ah were researched and analyzed for reliability and underwater capability. In order to function underwater, the battery would have to be of the sealed lead acid variety (seen in Figure 90 below), a type of battery which is leak-resistant in any orientation [37].
The final battery selected for the ROVEN Sealion was the UB12180 12V 18Ah sealed lead acid battery, seen in Figure 90. Not only would this work in any orientation underwater, it would also last the desired one hour in during typical ROV use. Detailed specifications for the selected battery can be seen in Table 19 [38].

**Table 19: Specifications for the UB12180 battery**

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery type</td>
<td>Sealed Lead Acid</td>
</tr>
<tr>
<td>Rated voltage</td>
<td>12 V</td>
</tr>
<tr>
<td>Rated ampere-hours</td>
<td>18 Ah</td>
</tr>
<tr>
<td>Weight</td>
<td>11.7 lbs</td>
</tr>
<tr>
<td>Dimensions</td>
<td>7.1 x 2.9 x 6.6 inches</td>
</tr>
</tbody>
</table>
3.17 Motor Control

As detailed in section 3.3.2 Experiments, each Minn Kota Endura C2 motor was found to draw anywhere between 6.4 and 30 amps at 12 volts, depending on the speed setting. Further experimentation with the motors’ mechanical speed controller yielded the results seen in Figure 91. In these experiments, the electrical resistance of each connection to the mechanical speed controller was measured at different settings. The results of the experiment showed that each wire on the motor could be given power independently, and the speed controller would send out specific voltages to the motor based on which wires were active.

![Figure 91: Trolling motor speed settings by wire](image)

Based on the thrusts obtained in testing the trolling motors, the project team came to the decision that excluding the red wire, as seen in the highlighted tiles in Figure 91, and thereby fixing the motor to a specific configuration was the best choice. At this setting, the motors would consume 13 amps while still providing 9 lbs of thrust at maximum speed (see Table 8, Table 9). The fixed configuration only establishes the maximum motor speed; by varying the
input voltage to the motor with a device like an electronic speed controller, the motor speed could be varied within this setting.

The mechanical speed controller provided with each motor would be very difficult to control remotely; the only method of switching between motor speeds would be to physically rotate the throttle with some type of actuator. Instead, the team opted to use an electronic speed controller (ESC) with each motor, which would serve as an adequate replacement to the more complicated mechanical speed controller. ESCs are commonly used to control both brushed and brushless DC motors. By sending a certain variable signal to an ESC, the operator is able to linearly adjust the speed of the attached DC motor [39].

In selecting an electronic speed controller (ESC) appropriate for the Minn Kota Endura C2 motors, the team had several requirements:

- 12V rating
- 15A+ capacity (each)
- Brushless DC motor configuration
- Reverse capability

The ProBoat PRB2314 electronic speed controller met all of these requirements; the specifications for the PRB2314 ESC are detailed in Table 20 [40].

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reverse</td>
<td>Yes</td>
</tr>
<tr>
<td>Maximum Current (continuous)</td>
<td>40 A</td>
</tr>
<tr>
<td>Input Voltage</td>
<td>4.8 - 12 V</td>
</tr>
<tr>
<td>Motor Limit</td>
<td>15 turn brushed</td>
</tr>
<tr>
<td>PWM Frequency</td>
<td>1000 Hz</td>
</tr>
<tr>
<td>Full On Resistance</td>
<td>0.070 Ω</td>
</tr>
</tbody>
</table>

Electronic speed controllers, including the PRB2314, are driven with pulse-width modulated (PWM) signals. In fact, PWM is a common technique for driving many electrical devices. By adjusting the width of electronic pulses, PWM can simulate certain power levels
needed to control electrical devices, such as DC motors. PWM is characterized by both its period and a duty cycle percentage that represents high-time for the signal. For example, a PWM signal with a frequency of 1 kHz (period of 1 millisecond) and a 40% duty cycle will be high for the first 0.4 milliseconds and low for the remaining 0.6 milliseconds. Various duty cycle configurations can be seen in Figure 92.

![Duty Cycle 10%](image)

![Duty Cycle 30%](image)

![Duty Cycle 50%](image)

![Duty Cycle 90%](image)

**Figure 92: Pulse-width modulated signals**

The PRB2314 ESC used in this project required a PWM signal with a 20 ms period. In accordance with common remote-control signals, three major settings were defined, as seen in Table 21. For example, when providing a 2 ms pulse every 20 ms to the ESC, the ESC would allow a full 12V signal to pass through from the power supply to the motor. By varying the pulse width between these three settings, the motor speeds could be varied with a nearly-continuous degree of resolution (255 distinct motor speeds, as further discussed in section 3.18 Software Design).
Table 21: Pulse width settings for the PRB2314 electronic speed controller

<table>
<thead>
<tr>
<th>Setting</th>
<th>Pulse Width (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Forward</td>
<td>2.0</td>
</tr>
<tr>
<td>Neutral</td>
<td>1.5</td>
</tr>
<tr>
<td>Full Reverse</td>
<td>1.0</td>
</tr>
</tbody>
</table>

3.18 Software Design

It was necessary to develop some form of software to work in conjunction with the Arduino Uno R3 microprocessor to control and receive information from the multitude of electrical components on the ROV. More specifically, there was a need for software to give the user control over the thrusters and lights, and the ability to view the ROV’s sensor readings. In the final version of the project, the software system was a combination of Processing [41] and Arduino [42] software with simple user input, serial communication, and a graphical user interface for information display. Figure 93 shows a flowchart of the program used on the ROVEN Sealion during testing.
To give the user easy-access, intuitive control over the ROV, a PlayStation 3 (PS3) Dual Shock 3 controller was used as the only input device. With drivers from MotioninJoy software, configuring the PS3 controller for use with the ROV was straightforward. The analog sticks and triggers were used for thruster control, and one face button was used to toggle the LED lights; a diagram of this configuration can be seen in Figure 94.
3.18.1 Processing

On a personal computer (PC) running Windows, the Processing software was used to write and execute a program to control user input, communicate with the Arduino, and display information for the user on a graphical user interface (GUI). Essentially, the PC served as the surface unit for the ROV system, as discussed in section 2.4 ROV System. The program, written in Java, conformed to the format of all Processing programs: an endless loop preceded by a brief setup period. The basic flow of the program is outlined below, and is followed by more detailed information about each part of the program. The full code used in the final version of the ROVEN Sealion software can be found in Appendix C.

- **Setup**
  - Controller inputs: three “sliders” (two analog sticks, one trigger pair), one button
  - Serial communication: 9600 baud

- **Loop**
  - Read state of four controller inputs
  - Map slider inputs from native format to bytes: [-1.0, 1.0] to [0, 255]
  - Send inputs as bytes over serial connection to Arduino along with a check byte (for simple error detection)
  - Draw and update GUI with slider positions and state of light
The aforementioned PS3 controller drivers, along with the ProControll library [43], made it possible to configure a virtual version of the PS3 controller, which could be monitored for user input. Input data for the thrusters was read from the vertical direction of both analog sticks and both secondary triggers (L2, R2). By pairing the triggers together, three “sliders” were established; a slider is an object in the ProControll library that has a single axis of continuous values ranging from -1.0 to 1.0. Input data for the lights was read from the “cross” (or X) button on the face of the controller. This controller configuration can be seen in Figure 94.

The user input data from the PS3 controller then needed to be processed to be sent to the Arduino and for use in the GUI. This required each portion of this input data to be mapped to a value from 0 to 255 (one byte); one byte for each thruster, and one byte for the lights. In general, mapping a value entails adjusting that value from its native range of values to a new range of values, a process similar to “scaling down” dimensions for some physical model. A portion of the mapping process can be seen in the code snippet shown in Figure 95. A separate map function was used for each half of the sliders in order to include a tolerance range in the middle. This tolerance range was necessary in order to allow for some slack or imperfections in the analog sticks on the PS3 controller itself. The button input did not require mapping, as it was already within a one-byte range; its value would either be 0 (not pressed) or 8 (pressed). The lights could be toggled by simply detecting whether that value was non-zero.
Once mapped, the user input data could be sent out serially via USB to the Arduino microprocessor. This five-byte message consisted of a check byte, left motor speed, right motor speed, top motor speed, and lights status. The check byte, arbitrarily chosen as the “#” character, was included for simple error detection. If the signal is sent with no error, the receiver detects the correct check byte and can then proceed with the rest of its program.

In addition to mapping and sending the user input data, the Processing program also displayed a continuously updating graphical user interface (GUI). This GUI, seen in Figure 96, was used to give the user information about the system, such as the commands being sent to the Arduino, the current state of the lights, and what various controller inputs' functionalities.
3.18.2 Arduino

The software used to control the Arduino is based on Processing, but uses C and C++ for greater control of the low-level Arduino hardware. A program for the Arduino was loaded onto the board prior to its use, so that the microprocessor could work on its own. The Arduino was needed not only to transfer user input to the thrusters and lights, but also to potentially read and send back sensor data.

Due to its similarity to Processing, the Arduino software also consists of two sections: a setup period followed by an endless loop. The program used on the Arduino is briefly outlined below, and the following paragraphs discuss each portion in more detail. The full code used in the final version of the ROVEN Sealion software can be found in Appendix D.

- Setup
  - Create and “attach” mock servo objects, one for each thruster’s speed controller
  - Serial communication: 9600 baud
  - Set pin for ROV lights to output

- Loop
  - Wait until there is incoming serial data
  - Confirm that check byte is valid, only continue if it is
  - Store each incoming byte (left speed, right speed, top speed, lights status)
- Map received motor speeds to PWM values: [0, 255] to [1000, 2000]
- Send PWM commands to each thruster’s speed controller
- Send high or low value to lights, depending on value received
- Send lights status back to Processing for display via serial connection

When setting up the program for use in the ROV system, there are a few details to consider. In configuring the Arduino for use with the electronic speed controllers (ESCs), “servo” objects were used. These objects, though typically meant for use with servo motors, are actually perfect for use with the ESCs, as they simply output a PWM signal with optional reverse. Another detail of note is that the serial communication, used to transfer and receive data from the surface unit, must have a baud rate identical to the one used in the Processing program.

Getting input from the user is essentially the reverse of sending it from the surface unit with Processing, which was detailed earlier. Unless all five bytes of the pre-established data packet have arrived, no action will be taken to control the thrusters or lights. Once all five bytes have arrived, the first byte (check byte) is compared to a predetermined value; if these two bytes are identical, the program determines that the data has arrived without error. The next three bytes in the data packet are the thruster speed values; each of these are mapped from a byte range (0 to 255) to a PWM range in microseconds (1000 to 2000), and then sent to the electronic speed controllers. More details on how PWM and the electronic speed controllers interact with the thrusters can be found in section 3.17 Motor Control.

### 3.19 Final Test

The final test for the ROVEN Sealion was performed on the afternoon of Wednesday, April 17, 2013. The test had been planned for earlier in the month, but some setbacks were encountered when finalizing the ROV. Luckily, the weather conditions were favorable, and there was only a slight current in Lake Quinsigamond, where the test was conducted. The ROV was tested from a dock in the lake, in water approximately 8 ft deep. The following sections discuss the team’s observations of the ROVEN Sealion’s performance during the test.
3.19.1 Flow Observations

Due to the short length of the tether and the lack of any reference points, the team was not able to determine the maximum speed of the ROV in any of its directions. By analyzing the video footage, it appears that the ROV may have been moving at 2-3 ft/s, however it is not known if this was with the thrusters at full power or not. What the physical test did reveal was that the ROV was not prone to drifting. In other words, when input from the controller ceased, the ROV did not continue to move. The large drag forces in all four directions attributed to this behavior. This is ideal for an observational ROV, which must remain focused on certain areas in order to gather data. Also observed from this test was the ability of the center thruster to make the ROV submerge. It was interesting to note that the center thruster required much more power to move the ROV from the surface to 1-2 feet below the surface then it did when moving from 2 feet below the surface to 4 feet below the surface. This is because at the surface, the center thruster was located only 2 inches below the water line. When the thruster begins to spin it only has a small amount of water to push against. This effect continues until the ROV is approximately 1-2 feet below the surface of the water. This can be easily seen by comparing Figure 97 and Figure 98.

Figure 97: ROV descending with turbulent region attached
The first figure shows the ROV as it submerges from the surface to 1 foot under the water. There is a clearly defined region of turbulent water just above the center thruster. The second figure shows the ROV as it has descended approximately 2 feet. Here, the region of turbulent water begins to separate from the thruster propeller and vanishes completely when the ROV has descended another foot. When this area of turbulent water dissipates, the ROV descends faster than before and it becomes much easier to control the ROV’s depth because the thruster has a column of water to push against.

3.18.2 Structural Observations

From the video footage the team was able to see minor displacement in the side horizontal polypropylene rods. A small amount of displacement was expected, yet these tests and analyses were done using an estimated ballast of about 40 lbs. Since the ROV proved to be more buoyant than expected, the team had to add an extra 30 lbs., for a total of 70lbs of extra
weight. Since all the analyses were done with a safety factor of at least 2, this extra weight did not prove to make much difference to the displacement in the frame.

There was a leakage into the top thruster. This thruster experienced some mechanical problems during the initial sealing. Urethane foam expanded into the motor and interrupted the contacts need for the motor to work. The team was able to fix this problem, but during the second round of sealing it was decided to not use urethane to avoid this problem. At the very end of field testing the top motor experienced a leak. The sealed was made from only silicon, which experienced constant pressure from the water, which may have caused the eventual leakage. Unlike urethane foam, silicon does not expand, so it could make a proper barrier in between the wires. After testing, the thruster was opened and cleaned in order to maintain its functionality.
CHAPTER 4: Conclusion

4.1 Project Objectives, Constraints, and Limitations

The ROVEN Sealion was designed and built with the goal of mimicking the observational ROVs used by oil companies for structural health monitoring. The main part of an oil platform that the ROVEN Sealion would be inspecting is the blowout preventer. The blowout preventer serves as a last resort to stop oil leakage during pipe failure by closing off the pipe.

After researching industrial ROVs, the team chose the size and resulting approximate weight of the ROVEN Sealion. Using these specifications, the team designed multiple prototypes that met the requirements. This research helped in the selection of the materials. Since the main purpose of the ROVEN Sealion was structural health monitoring on blowout preventers, a field of vision of 5 ft. from the camera was desired.

The project budget proved to be the biggest constraint for the team. Many of the components that would be ideal to make the ROV reach depths up to 5000 ft were too expensive for a budget of $960. These budget limitations led the team take price into consideration before deciding on certain parts and components. The camera was one of the components limited by the budget. The team used a Logitech Webcam to record and to obtain feedback underwater, yet the picture quality did not match that of real ROVs, which use HD video quality. The urethane floatation was another item which was used instead of the team’s original choice of syntactic foam. Additionally, the electrical components of the ROV were sealed within Pelican Micro Cases using silicon rather than being sealed within a custom-made encasing. Although their performance exceeded expectations, the inclusion of the Minn Kota Endura C2 trolling motors used to propel the ROV was also a result of a limited budget.

4.2 Results and Future Work

The project resulted in a working ROV capable of basic monitoring. During testing, the ROVEN Sealion was able to monitor a PVC pipe and a dock in Lake Quinsigamond, in Worcester, MA. The ROV functioned as predicted by its design with some minor hindrances.

There were some factors that could not be included in the final test and there were some parts that could have been improved upon. The sensors were working but were not tested in the
field. The team was not able to conduct enough tests to verify all components were working properly. These tests included field testing the temperature sensor, pressure sensor, and compass module. Many other readings that the team desired, such as the depth the ROV reached, the actual velocity, and the temperature of the water during testing were not measured because of a lack of time.

In the finished ROV, the capacity of the battery was never fully tested. As such, the maximum length of time the ROV could reliably run was calculated, but never confirmed. The frame components, which included the polypropylene rods, threaded rods, steel trusses, and PVC joints all withstood the loads encountered in testing and showed no signs of plastic deformation. From the video captured during testing, a small displacement could be seen in the top horizontal polypropylene rods that support the motors.

The final ROV was 23 lbs more buoyant than predicted. The initial prediction was calculated assuming that the floatation provided a uniform 58 lbs/ft³ buoyant force, which would give the ROV 47 lbs of buoyancy. However, since the floatation was created in layers and in various environmental conditions, its buoyancy varied from the expected value. The solutions were mixed by hand meaning that there was no accurate way to ensure that the solutions were uniform. To compensate for the surplus buoyancy, additional weight was added when testing the ROV.

Time constraints as well as a limited budget caused many of the extra components and payloads that were originally planned for the ROVEN Sealion to be dropped. The team decided to design the ROV so that extra payload could be easily added without having to change the overall ROV design. The ROV’s initial design was intended to be used for structural health monitoring on offshore oil platforms, yet the related payloads were not added, allowing for future groups to take the project in a different direction.

The ROVEN Sealion is an observational class ROV with some payload capability, so it can be applied to multiple fields outside of structural health monitoring. The ROVEN Sealion is estimated to be able to work at depths of up to 25 ft. At this point, the waterproof sealing for the thrusters and electronics would most likely break or leak. If these seals are reinforced, the ROV could reach depths necessary for monitoring the majority of lakes.

The ROVEN Sealion could perform sea floor mapping and monitoring if equipped with the appropriate type of payloads. The ROVEN Sealion is inexpensive compared to similarly-
sized ROVs, making it more accessible to organizations such as police departments, sailing clubs, and boat owners that need to find lost items in lakes. The ROV can also be used to inspect ships already in the water, or other submerged structures.

The team was planning to include a Cathodic Protection Probe, Magnetic Particle Inspection capability, and a pressure washer or cleaning brush. Adding a manipulator capable of interacting with the environment or a control panel is suggested as future work for the ROVEN Sealion.

The limited budget didn’t allow the design team to include all five thrusters as originally intended, which eliminated one degree of freedom. Future work could include adding these additional thrusters, or redesigning the thruster connection to the ROV frame.

Depending on the type of data cable used in the tether, signal strength can deteriorate rapidly over long distances. Using only USB connections, the ROVEN Sealion’s tether length was limited to approximately 15 ft. Future work in modifying the cable type (to Cat 5, for example) or designing an alternate communication system, such as transmitting wirelessly to a floating receiver, would overcome this limitation.

Completely waterproofing the main electrical components of the ROV proved challenging. Sealing the microcontrollers, sensor circuits, and speed controllers required drilling holes in their Pelican Micro Cases. Although silicone typically works well for sealing these kinds of cases, sealing a bundle of wires left some imperfections. Future iterations of the ROVEN Sealion project could design and construct a custom waterproof encasing for electrical components.

Working with high-power devices, such as the Minn Kota trolling motors, put the Pro Boat electronic speed controllers (ESCs) at risk. Simply touching two leads in the tightly-packed waterproof cases could cause an ESC failure. Unfortunately, the Pro Boat ESCs were not possible to repair. Future work on the ROVEN Sealion could involve the design of an H-bridge speed control circuit for each thruster.
REFERENCES


## APPENDICES

### Appendix A: Part Drawings

#### Part Index

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Name</th>
<th>Weight (lbs)</th>
<th>Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>3 Way Elbow Fitting</td>
<td>0.097</td>
<td>PVC</td>
</tr>
<tr>
<td>002</td>
<td>USB 12180 Battery</td>
<td>12.25</td>
<td>Multiple</td>
</tr>
<tr>
<td>003</td>
<td>Anvil Pipe Clamp</td>
<td>0.576</td>
<td>Galvanized Steel</td>
</tr>
<tr>
<td>004</td>
<td>Flotation</td>
<td>15.75</td>
<td>Urethane Foam</td>
</tr>
<tr>
<td>005</td>
<td>Minn Kota Endura C2 30</td>
<td>8.986</td>
<td>Multiple</td>
</tr>
<tr>
<td>006</td>
<td>Slip Tee</td>
<td>0.051</td>
<td>PVC</td>
</tr>
<tr>
<td>007</td>
<td>Threaded Steel Rod</td>
<td>2.010</td>
<td>Harden Steel</td>
</tr>
<tr>
<td>008</td>
<td>Threaded Steel Rod Top</td>
<td>1.319</td>
<td>Harden Steel</td>
</tr>
<tr>
<td>009</td>
<td>Steel Rail</td>
<td>1.092</td>
<td>Galvanized Steel</td>
</tr>
<tr>
<td>010</td>
<td>Dome</td>
<td>1.234</td>
<td>Acrylic</td>
</tr>
<tr>
<td>011</td>
<td>Truss</td>
<td>0.792</td>
<td>Galvanized Steel</td>
</tr>
<tr>
<td>012</td>
<td>U-Bolt</td>
<td>0.060</td>
<td>Steel</td>
</tr>
<tr>
<td>013</td>
<td>Support Nut</td>
<td>0.036</td>
<td>Steel</td>
</tr>
<tr>
<td>014</td>
<td>1040 Pelican Micro Case Series</td>
<td>0.742</td>
<td>PP Copolymer</td>
</tr>
<tr>
<td>015</td>
<td>1050 Pelican Micro Case Series</td>
<td>0.802</td>
<td>PP Copolymer</td>
</tr>
<tr>
<td>016</td>
<td>Dome Support</td>
<td>1.018</td>
<td>Polypropylene</td>
</tr>
</tbody>
</table>
Slip Tee

Dimensions: 1.3000

Material: PVC

Weight: 0.051

Scale: 1:12

Revision: 2
ROVEN Sealion

Threaded Steel Rod Top

Dimensions may not be accurate as this part was not made by the techn.
U-Bolt

Dimensions may not be accurate as the part was not made by the team.

Material: Steel

Weight: 0.60 kg

Scale: 1:1

Sheet: 1 of 1

WPI

PROPRIETARY AND CONFIDENTIAL
This information contained in this drawing is the property of the company and is intended solely for the use of the original holder.

Drawing

Title: ROVAN Section

WPI

012

A

1
Appendix B: Thruster Exploded View
Appendix C: Processing Code

The following program is the Processing half of the code used to control the thrusters and lights of the ROVEN Sealion in its most recent test.

/** This program interfaces with a PS3 Dualshock3 controller for input.
* Analog stick vertical values are sent in a series of 5 bytes to an
* Arduino for motor control.
* The "cross" button is sent to the Arduino to toggle LED lights.
* @author kmalehorn
* * Drivers for the PS3 controller on Windows: MotioninJoy or XInput (DS3 wrapper).
* ProControll library for controller interfacing: http://creativecomputing.cc/p5libs
/procontroll/index.htm
* * Based on code from:
* - Matthew Spinks (http://www.youtube.com/watch?v=_EgXrcxZPnE)
* - nahamancygig (http://www.instructables.com/id/Use-a-PS3-Controller-to-control-
an-Arduino-NXT-Bot)
*/

import procontroll.*;            // for gamepad interfacing
import processing.serial.*;      // for serial communication with Arduino program
import net.java.games.input.*;
import java.io.*;                // for printing

ControllIO controll;
ControllDevice gamepad;
ControllSlider leftSlider;
ControllSlider rightSlider;
ControllSlider topSlider;
ControllButton crossButton;

Serial myPort;        // for connecting to Arduino over serial
PFont f, b;           // for display fonts; corresponds to "data" folder in cd
char HEADER = '#';    // header byte sent to Arduino for check

static char LEFT_Y_SLIDER  = 6;   // "6: Y axis absolute"
static char RIGHT_Y_SLIDER = 2;   // "2: Z Rotation absolute"
static char TRIGGER_SLIDER = 1;   // "1: Slider absolute"
static char CROSS_BUTTON = 2;      // "2: ?" because X symbol not supported
// buttons: triangle (0), circle (1), cross (2), square (3)

boolean BUTTON_TOGGLE = false;    // value for cross (X) button toggle

float tolerance = 0.24;          // used in mapping to determine if stick is off of zero
// 0.1 is close to min; if too sensitive, increase to 0.2~0.3
// 0.24 good when using triggers, otherwise they jump at ~25%

static boolean DEBUG = true;      // if true, program will print/display more messages
static boolean SERIAL = false;     // if true, program will try to establish
// serial connection with Arduino

int previousTime = 0;             // used to print messages with certain time interval

// for DEBUG display only, not required for program; shows original [-1,1] values
float origLeftY, origRightY, origTopY;
/** Initialization
* Sets up display size, display font, serial communication, and gamepad sliders.
*/
void setup(){
  size(800,400);    // size of popup display window (width, height)
f = loadFont("ArialMT-48.vlw");        // load normal display font
b = loadFont("Arial-BoldMT-48.vlw");   // load bold display font

// The microcontroller will connect on the first port listed with a 9600 baud rate
//println(Serial.list());
if(SERIAL){
  myPort = new Serial(this, Serial.list()[1], 9600);   // for Arduino interfacing
  println("Using serial port: " + Serial.list()[1]);
}

ccontroll = ControllIO.getInstance(this);
gamepad = controll.getDevice("MotioninJoy Virtual Game Controller");
/*  gamepad.printSticks();  gamepad.printSliders();  gamepad.printButtons();  */

// get slider objects
leftSlider  = gamepad.getSlider(LEFT_Y_SLIDER);
rightSlider = gamepad.getSlider(RIGHT_Y_SLIDER);
topSlider   = gamepad.getSlider(TRIGGER_SLIDER);
// NOTE: in MiJ, triggers (topSlider) configured so leftTrig:+, rightTrig:-

// get button object
crossButton = gamepad.getButton(CROSS_BUTTON);
// NOTE: when pressed, button reads as decimal value 8.0

if(DEBUG) println("setup complete");
}

/** Main loop
* 1) Gets controller input values
* 2) Maps them to 8-bit values
* 3) Sends these 8-bit values to Arduino over serial
* 4) Displays their values and position on screen
*/
void draw(){

  // get slider values
  float leftY  = leftSlider.getValue();     // left stick vertical "slider" value
  float rightY = rightSlider.getValue();    // right stick vertical "slider" value
  float topY   = topSlider.getValue();      // combined trigger "slider" value
  // (right +, left -)
  float buttonVal = crossButton.getValue(); // value of cross (X) button

  if(DEBUG){    // for debug display only, not required for program
    origLeftY = leftY; origRightY = rightY; origTopY = topY;
  }

  // map left and right Y axes from [-1,1] to 8-bit range for both output and display
  leftY  = mapSlider(leftY);
  rightY = mapSlider(rightY);
topY   = mapSlider(topY);
// send slider position data to Arduino over serial as 5 bytes
// (checkbyte, left motor, right motor, top motor, lights on/off)
// note: java/processing interprets bytes as signed, but C/arduino interprets them
// as unsigned; be careful when sending messages between the two
if(SERIAL) sendPackage(byte(leftY), byte(rightY), byte(topY), byte(buttonVal));

// get serial from Arduino (just for light status in GUI for now)
if(SERIAL){
  while(myPort.available() > 0){  // waiting lightStatus
    if(myPort.read() == '4'){ // checkbyte disabled
      int inBuffer = myPort.read();
      if(inBuffer != -1){
        buttonVal = inBuffer;
      }
    }
  }
}

// draw visual representation of sticks, lights
updateDisplay(leftY, rightY, topY, buttonVal);

/********************************* HELPER FUNCTIONS *********************************/

/** Maps slider values from [-1,1] to [0,255] (one byte).
 * @param slider The slider value to map.
 * @return The mapped slider value in a one-byte range (note: still a float).
 */
float mapSlider(float slider){
  // maps left & right slider values for output & display, with a certain tolerance
  // original -1 ...... 0 ...... 1
  // return 255 ..... 127 ..... 0
  // note: values reversed because -1 is full forward on analog stick
  if(slider < -tolerance) {  // negative value
    // map from [-1,-tol] to [255,127]
    slider = map(slider, -1, -tolerance, 255, 127);
  }
  else if(slider > tolerance) {  // positive value
    // map from [tol,1] to [127,0]
    slider = map(slider, tolerance, 1, 127, 0);
  }
  else {
    slider = 127;  // neutral value in any other case
  }
  return slider;
}

/** Writes motor data to Arduino over serial.
 * @param leftMotor Byte representing left stick's vertical direction & magnitude.
 * @param rightMotor Byte representing right stick's vertical direction & magnitude.
 */
void sendPackage(byte leftMotor, byte rightMotor, byte topMotor, byte buttonVal){
  myPort.write(HEADER);
  myPort.write(leftMotor);
  myPort.write(rightMotor);
  myPort.write(topMotor);
  myPort.write(buttonVal);
if (DEBUG == true) {
  if (buttonVal > 0) buttonVal = 1; // just for print to terminal
  if (millis() - previousTime >= 3000) { // once per second
    previousTime = millis();
    println("Sending serial package: left = " + (leftMotor & 0xFF) + ", right = " + (rightMotor & 0xFF) + ", top = " + (topMotor & 0xFF) + ", lights = " + buttonVal);
  }
}

/** Updates the on-screen representation of the joystick slider values.  
* NOTE: *Not* required for program to function; only for visualization.  
* @param leftY The left slider decimal value [0,255].  
* @param rightY The right slider decimal value [0,255].  
* @param topY The trigger slider decimal value [0,255].  
*/
void updateDisplay(float leftY, float rightY, float topY, float buttonVal) {
  background(255); // clears display window (white)

  // set the coordinates for the left and right stick objects
  int topline = width*1/4; // for motor sliders (aka "leftHalf")
  int leftLine = topline - 100;
  int rightLine = topline + 100;

  int rightHalf = width*3/4; // line for components on the right half

  int lineHeight; // starting height for blocks of text; adjusted before use
  int space; // used to give horizontal space between blocks of
             // text centered around line

  rectMode(CENTER);

  fill(154, 154, 154); // light gray

  // draw sliders
  rect(leftLine, height/2, 20, 256);
  rect(rightLine, height/2, 20, 256);
  rect(topLine, height/2, 20, 256);

  // draw center lines
  line(leftLine - 40, height/2, leftLine + 40, height/2);
  line(rightLine - 40, height/2, rightLine + 40, height/2);
  line(topLine - 40, height/2, topline + 40, height/2);

  // divider lines to separate parts of display
  line(width/2, 10, width/2, height - 10);

  // draw rectangles for sliders
  fill(0); // black
  rect(leftLine, height/2 - leftY + 127, 40, 20); // reversed on purpose
  rect(rightLine, height/2 + rightY - 127, 40, 20);
  rect(topLine, height/2 - topY + 127, 40, 20);

  // label sliders
  textAlign(CENTER, CENTER);
  textFont(16);
  text("LEFT", leftLine, height/2 - 140);
  text("RIGHT", rightLine, height/2 - 140);
  text("TOP", topline, height/2 - 140);
// show decimal values for slider positions [0,255]
textField(f,16);
textAlign(CENTER, TOP);
text(int(leftY), leftLine, height/2 + 140);
text(int(rightY), rightLine, height/2 + 140);
text(int(topY), topLine, height/2 + 140);

// title info
textAlign(CENTER, BOTTOM);
textField(b,18);
text("Thruster Controls", topLine, height/2 - 160);

// LED lights status (on/off)
space = 2;
textAlign(RIGHT, BOTTOM);
text("Lights:", rightHalf - space, height/2 -160);
textAlign(LEFT, BOTTOM);
if(buttonVal > 0){  // black if ON
  fill(0);  // black if ON
  text("ON", rightHalf + space, height/2 -160);
}
else{
  fill(170);  // light gray if OFF
  text("OFF", rightHalf + space, height/2 - 160);
}

// controls (static reference displayed to help new users)
space = 4;
textAlign(CENTER);
textField(b, 18);
fill(0);
startHeight = height/2 + 80;
text("Controls", rightHalf, startHeight);
textAlign(RIGHT);
textField(f, 16);
text("Left Stick (vertical):" , rightHalf - space, startHeight + 30);
text("Right Stick (vertical)", rightHalf - space, startHeight + 50);
text("R2 Trigger:", rightHalf - space, startHeight + 70);
text("Cross (X) Button:", rightHalf - space, startHeight + 90);
textAlign(LEFT);
textField(b, 14);
text("LEFT THRUSTER", rightHalf + space, startHeight + 30);
text("RIGHT THRUSTER", rightHalf + space, startHeight + 50);
text("TOP THRUSTER (down)", rightHalf + space, startHeight + 70);
text("TOGGLE LIGHTS", rightHalf + space, startHeight + 90);

// display extra info (orig slider values & binary values being output to Arduino)
if(DEBUG == true) {
  fill(0);
  textAlign(CENTER, BASELINE);
textField(f,14);

text(origLeftY, leftLine, height/2 + 170);
text(origRightY, rightLine, height/2 + 170);
text(origTopY, topLine, height/2 + 170);

text(binary(byte(leftY),8), leftLine, height/2 + 190);
text(binary(byte(rightY),8), rightLine, height/2 + 190);
text(binary(byte(topY),8), topLine, height/2 + 190);
}
}
**Appendix D: Arduino Code**

The following program is the Arduino half of the code used to control the thrusters and lights of the ROVEN Sealion in its most recent test.

```cpp
/** Serial communication with Processing to control electronic motor
 * via PS3 controller input.
 * @author kmalehorn
 *
 * Receives 5-byte data from Processing sketch (also called thrusters_lights):
 * - First byte is a checkbyte (arbitrarily chosen to be '#').
 * - Second byte is left motor speed
 * - Third byte is right motor speed
 * - Fourth byte is top motor speed
 * - Fifth byte is lights status (on/off)
 * Motor speeds have domain [-128,127]
 *   Positive = forward, negative = reverse, zero = neutral.
 * Motor speeds are converted into PPM (PWM) signal [1000,2000]
 *   for a Minn Kota Endura C2 trolling motor controlled
 *   by a ProBoat PRB2314 electronic speed controller.
 *
 * (code partially based on code from Matthew Spinks)
 */

#include <Servo.h>
#define DEBUG 0

// global motor values
const int leftMotorPin  = 9;    // pin to ESC for left motor
const int rightMotorPin = 10;   // pin to ESC for right motor
const int topMotorPin   = 11;   // pin to ESC for top motor
const int lightsPin     = 13;   // pin to CREE LED lights (same as on-board LED)
Servo leftMotor;
Servo rightMotor;
Servo topMotor;

// global LED light values
byte prevInput  = 0;
char lightStatus = LOW;

void setup(){
  Serial.begin(9600);    // serial connection, 9600 bps (baud)
  Serial.flush();

  pinMode(lightsPin, OUTPUT);  // on-board LED for testing purposes

  leftMotor.attach(leftMotorPin);  // init "servo" motors by associating with pin
  rightMotor.attach(rightMotorPin);
  topMotor.attach(topMotorPin);

  if(DEBUG) Serial.println("setup complete");
}
```
```c
void loop(){
  if (Serial.available() > 4) { // expecting 5 bytes  
    // (check, left, right, top, lights)
    // get incoming byte:
    byte checkByte = Serial.read();

    if(checkByte == '#') { // '#' chosen arbitrarily (also: 0b0010011)
      if(DEBUG) Serial.println("checkByte OK!");
      // get motor "speeds" from Processing, range: [0,255]
      byte leftByte   = Serial.read(); // note: interpreted as unsigned
      byte rightByte  = Serial.read();
      byte topByte    = Serial.read();
      byte buttonByte = Serial.read();

      // remap to PPM signal [1000,2000] with 1500 neutral
      int leftPPM   = mapPPM(leftByte); //round(map(leftByte, 0, 255, 1000, 2000));
      int rightPPM  = mapPPM(rightByte);  
      int topPPM    = mapPPM(topByte);   

      // write PPM signal to electronic speed controllers
      leftMotor.writeMicroseconds(leftPPM);
      rightMotor.writeMicroseconds(rightPPM);
      topMotor.writeMicroseconds(topPPM);

      // toggle LED lights if button pressed
      toggle(&lightStatus, buttonByte); // toggles lightStatus value
      digitalWrite(lightsPin, lightStatus); // uses lightStatus value
                                      // to turn lights on/off
    }
  }
}

/** Maps incoming byte to PPM values for electronic speed controller.
 *  Used to ensure accurate mapping.
 *  @param inByte The byte to map.
 *  @return The mapped value as an integer, in the range [1000,2000].
 */
int mapPPM(byte inByte){
  float tempF = map(inByte, 0, 255, 1000, 2000);
  tempF = round(tempF);
  return (int)tempF;
}

/** Toggles specified value if second (incoming) value is non-zero.
 *  @param toToggle Pointer to value to toggle (LOW/HIGH or 0/1)
 *  @param input Incoming value used to decide whether to toggle
 */
void toggle(char* toggleVal, byte input){
  // toggle if input non-zero AND different than previous
  if((input > 0) && (input != prevInput)){
    if(*toggleVal == LOW) *toggleVal = HIGH;
    else *toggleVal = LOW;
  }
  prevInput = input;
}
```
This is the most recently-used code for interfacing with and obtaining data from the DS18B20 temperature sensor and the HMC6352 magnetic compass module.

```c
/** Code adapted from bildr.org
 * @author kmalehorn
 * Configures and gets/prints readings from DS18B20 temperature sensor.
 * Reads the heading from the HMC6352 compass and prints it via serial.
 */

#include <OneWire.h>  /* for HMC6352 compass */
#include <Wire.h>     /* for DS18B20 temperature sensor */
#include <stdio.h>

int DS18S20_Pin = 2; //DS18S20 Signal pin on digital 2
int HMC6352SlaveAddress = 0x42;
int HMC6352ReadAddress = 0x41; //"A" in hex, A is get data command

int headingValue;

//Temperature chip I/O
OneWire ds(DS18S20_Pin); // on digital pin 2

void setup(void) {
  /* HMC6352 */
  // "The Wire library uses 7 bit addresses throughout.
  //If you have a datasheet or sample code that uses 8 bit address,
  //you'll want to drop the low bit (i.e. shift the value one bit to the right),
  //yielding an address between 0 and 127."
  HMC6352SlaveAddress = HMC6352SlaveAddress >> 1; // I know 0x42 is less than 127, but this is still required

  Serial.begin(9600);
  Wire.begin();
}

void loop(void) {

  // temperature
  float celsius = getTemp();
  Serial.print(celsius);
  Serial.print(" (");
  Serial.print(convertTemp(celsius));
  Serial.println(")");

  // compass
  float heading = getHeading();
  Serial.print(heading);
  Serial.println(" degrees");

  delay(400); // just here to slow down the output so it is easier to read
}
```
/** Gets and returns the temperature from the DS18S20 sensor in degrees Celsius.  
* @return Temperature in degrees Celsius.  
*/
float getTemp(){
    byte data[12];
    byte addr[8];

    if ( !ds.search(addr)) {  
        // no more sensors on chain, reset search 
        ds.reset_search();
        return -1000;
    }
    if ( OneWire::crc8( addr, 7) != addr[7]) { 
        Serial.println("CRC is not valid!");
        return -1000;  
    }
    if ( addr[0] != 0x10 && addr[0] != 0x28) {  
        Serial.println("Device is not recognized");
        return -1000;  
    }
    ds.reset();
    ds.select(addr);
    ds.write(0x44,1); // start conversion, with parasite power on at the end
    byte present = ds.reset();
    ds.select(addr);
    ds.write(0xBE); // Read Scratchpad

    for (int i = 0; i < 9; i++) { // we need 9 bytes
        data[i] = ds.read();
    }
    ds.reset_search();
    byte MSB = data[1];
    byte LSB = data[0];

    float tempRead = ((MSB << 8) | LSB); //using two's compliment
    float TemperatureSum = tempRead / 16;

    return TemperatureSum;
}

/** Converts temperature from degrees Celsius to degrees Fahrenheit.  
* @param celsius The temperature (in Celsius) to be converted.  
* @return The temperature in degrees Fahrenheit.  
*/
float convertTemp(float celsius){
    return ((celsius * 1.8) + 32);  
}
/** Gets and returns the heading value from the HMC6352 magnetic compass.
 * @return Heading in degrees (CCW from North)
 */
float getHeading(){
  //"Get Data. Compensate and Calculate New Heading"
  Wire.beginTransmission(HMC6352SlaveAddress);
  Wire.write(HMC6352ReadAddress);    // The "Get Data" command
  Wire.endTransmission();

  //time delays required by HMC6352 upon receipt of the command
  //Get Data. Compensate and Calculate New Heading : 6ms
  delay(6);

  Wire.requestFrom(HMC6352SlaveAddress, 2); //get the two data bytes, MSB and LSB

  //"The heading output data will be the value in tenths of degrees
  //from zero to 3599 and provided in binary format over the two bytes."
  byte MSB = Wire.read();
  byte LSB = Wire.read();

  float headingSum = (MSB << 8) + LSB; // (MSB / LSB sum)
  float headingInt = headingSum / 10;

  return headingInt;
}