

RoboSub Team Killick 2017-2018

Mid-project Report

Fall Semester 2017

By:

Nick Baron
Marta Camacho
Ross Dobitz
Jake Hannafious
Ryan Harty
Jordan Lankford
Moez Malik
Daniel Mansfield
Billy Phillips
Oren Pierce
Jeremy Naeve

O2P/VIP Members:

Al Alothman
Jake Harmon
Samual McCallum
Angel Sanchez
Katie Wood

Department of Electrical and Computer Engineering

Colorado State University
Fort Collins, Colorado 80523

Project advisor(s): _Dr. Anthony Maciejewski, Megan Emmons, Chris Robbiano_____

Approved by: _____Chris Robbiano, Megan Emmons_____

Abstract

The RoboSub project goal is to design and build a fully autonomous underwater vehicle (AUV) to compete in the United States Navy RoboSub competition in San Diego, California. Historically, the top-placing teams have been from universities and combine students from multiple engineering disciplines. Although the project is expressly for a competition, it offers students the opportunity to develop an autonomous vehicle, an area many leaders in industry are aggressively pursuing.

The project, and team has been broken down into three and a half subteams or categories with the half team being controls which, for the first semester, included members from two of the other subteams. The three full teams are mechanical, power and propulsion, and vision and sensors. All are key areas of focus with specific tasks to develop a vehicle, with controls integrating sensory input and deciding the appropriate motor output for the developed mechanical system. The project as a whole implements a wide variety of technology: from image processing and recognition, data acquisition from inertial measurement units and pressure transducers, and current measurement - all of which are required to correctly control the motors. The RoboSub project has an additional challenge in that it is a Vertically Integrated Project (VIP) so it includes underclassmen as well as seniors with the goal of continually evolving the project. Underclassmen become involved at earlier stages of the project and are familiar with design choices so can more quickly grow the project when they become full senior members.

The project is advancing out of the beginning stages and one of the principal findings so far is that designs and methods must be constantly evaluated. This helps the team continually push the envelope by adding capabilities to the AUV. Testing failures help the team further improve and design a robust vehicle that is better able to perform more difficult tasks. Due to the competition based nature of the project as well as the evolutionary nature of a VIP team, the team has learned that documentation explaining decisions and progress will act as a foundation for future teams to build off of and develop state of the art AUV's that are competitive.

This year's team has accomplished much during the Fall semester. The propulsion system has been completed and is able to take instructions from both a gaming controller as well as a Raspberry Pi. A new chassis design was built and tested underwater with the propulsion system. Computer vision has made great headway in shape and color detection that will prove essential in achieving autonomy. Collecting data from the Inertial Measurement Unit (IMU) has been completed as well as for the pressure transducers to better control the thrusters and supplement data gathered from computer vision. As a whole, the team is continually progressing towards the ultimate goal of building a base for future teams to use to compete in the RoboSub competition.

Table of Contents

Title

Abstract

Table of Contents

1. Introduction
2. Review of Previous Work
 - a. Mechanical Subteam
 - b. Power and Propulsion Subteam
 - c. Vision and Sensors Subteam
 - d. Controls Subteam
3. Mechanical Subteam
 - a. Testing Conducted on Previous Designs
 - b. Design Objectives and Constraints
 - i. Housing
 - ii. Torpedo Launcher
 - iii. Ballast System
 - iv. Manipulator Arm
4. Propulsion Subteam
 - a. Introduction
 - b. Hall Effect Sensing
 - c. Motor Control
 - i. Direct
 - ii. Indirect
5. Sensors and Vision Subteam
 - a. Vision
 - i. Object Detection
 - ii. Identifying Color
 - iii. Returning Information
 - b. Hardware and Interconnects
 - i. Main Computer Board
 - ii. Cameras
 - c. IMU
 - i. Intro
 - ii. About the IMU
 - iii. Feedback
 - iv. Orientation
 - v. Dead Reckoning
6. Controls Subteam
 - a. Controls Designs and Decisions
 - i. Power and Propulsion Side

ii. Vision and Sensors Side

7. Conclusions and Future Work

- a. Mechanical Subteam
- b. Power and Propulsion Subteam
- c. Vision and Sensors Subteam
- d. Controls Subteam

References

Bibliography

Appendix A – Abbreviations A-1

Appendix B – Budget B-1

Appendix C – Project Plan Evolution C-1

List of Tables and Figures

Figure 1: CAD Rendering of Work Completed Fall 2017

Figure 2: T200 Thruster Package

Figure 3: HobbyKing 30A ESCs

Figure 4: ACS712 Hall Effect Sensors

Figure 5: Motor State Diagram

Figure 6: Motor Control Diagram

Figure 7: Diagram of Nvidia Jetson TX2

Figure 8: GoPro Hero4 Silver Edition

Figure 9: Raspberry Pi Camera

Figure 10: J20 connector for Jetson board

Figure 11: B101 HDMI to CSI-2 converter

Figure 12: Pressure transducers hardware

Figure 13 : The Normalized-Cross Correlation

Figure 14: Algorithm output for an idealized-edge-detected circle

Figure 15: Unprocessed Image

Figure 16: Processed Image

Figure 17a: Sample 1

Figure 17b: Average Color of 17a

Figure 18a: Sample 2

Figure 18b: Average Color of 18a

Figure 19: Sparton IMU with Development Board

Figure 20: CAD Rendering of Hydrodynamic Fairings

Chapter 1 - Introduction

The RoboSub project is a competition put on by the U.S. Navy and RoboNation and is held annually at the TRANSDEC facility in San Diego California. The goal is to design a fully autonomous underwater vehicle (AUV) to traverse an obstacle course and complete a wide variety of tasks, all in 15 minutes or less. Additionally the vehicle must be under 125 lbs and battery powered with the open circuit voltage of each battery less than 60V DC. The vehicle can only compete once, then the following years team must design an entirely new vehicle.

The obstacle course changes from year to year with different themes. There have been consistencies from year to year which include the following: line tracking, passing through gates, buoy identification, hitting targets with a torpedo, and manipulation of objects with a robotic arm. Additionally, the weight restriction and time to complete the course have continually awarded more points for lower values on both. This has been the primary focus of both last year's team as well as this year's.

Currently, the RoboSub team is comprised of 11 seniors. The breakdown is as follows: 7 electrical engineering majors, 3 mechanical engineering majors, and one computer engineering major. Additionally, the project has 5 underclassmen who are part of the vertically integrated project (VIP) program, of which the RoboSub project is. This allows the project to incorporate underclassmen to gain experience working on the team as well as join the project as seniors with a well developed understanding of project requirements. The role of these students this year is to validate and test the designs the seniors create to help the team create a more robust system. Having undergraduate students also helps the project continue to develop and progress. The hope is that this project will consistently be offered for years to come and actively compete and place well, if not win, the RoboSub competition.

This year's team is split into three and a half subteams which are power and propulsion, vision and sensors, mechanical, and controls. Controls is a new subteam introduced this year and is considered a half team because it is integrated into both power and propulsion and vision and sensors subteams. The primary role of the controls subteam is to create the "brain" of the AUV by turning vision and sensors data into motor commands. Power and propulsion is responsible for taking instructions in the form of a serial string and turn those into pulse width modulated (PWM) signals which are sent to the electronic speed controllers (ESCs) to control motor speed. The vision and sensors subteam generates data from an inertial measurement unit (IMU), pressure transducers, and cameras to collect data to give to controls to determine where the AUV is located and what it needs to do. The mechanical subteam works on the chassis as well as all torpedo launchers and mechanical arms and any extremities on the chassis. With one year of experience, and a few returning members of the original team, this year's team was able to work even more with other subteams to begin the critical testing phase.

The team set out with the goal to have a vehicle in the water that would track and follow a line. This was a very ambitious task given that many facets of the design would be reevaluated and eventually changed. What was accomplished this semester was water testing of the new chassis design to find out how to better water proof the design. Additionally, the propulsion system was tested using manual input from a gaming controller to find out how the chassis and thrusters behave in water as well as what speeds we could travel at. The team also began implementing an accelerometer to perform correctional movements to keep the vehicle balanced as it moved through the water.

This report serves to describe the work done during the Fall 2017 semester and outlines what is expected to be done during the Spring 2018 semester. Design choices as well as part selection will also be described with the purpose of aiding future teams by explaining the team's decision making process as well as sharing mistakes and what was learned from those mistakes.

The second chapter of this paper will provide a critical review of the work done by the 2016-17 team as a whole as well as delving into the work of the individual subteams. Chapter 3 will describe, in detail, the work done by the mechanical subteam during the Fall semester as well as address design changes and decisions that changed from the 2016-17 year's team. The power and propulsion subteam, vision and sensors, and controls subteams will follow the same format in chapters 4, 5, and 6. Chapter 7 will describe future work to be done during the Spring semester by team.

Chapter 2 - Review of Previous Work

The 2016-17 RoboSub team was the first team to work on the project. Due to the nature of a new project, last year's team worked primarily on investigating how teams that won the competition went about the design process as well as what parts were consistently being used and how the AUV was designed.

Mechanical Subteam

During the Fall 2016 semester, the previous mechanical team was dedicated to designing the three main mechanical components of the vessel. This was done through the use of statistical analysis, simulations in computational fluid dynamic software, industry design reviews, CAD modeling, and use of fluid and material computations.

During the Spring 2017 semester, the previous mechanical team was focused on fabricating their electronic housing and frame design from the previous semester and developed a preliminary safety system to bring the submarine back to the surface when power is lost, and prototyped 3D printed propellers and motor shrouds

Power and Propulsion Subteam

The power and propulsion subteam focussed mainly on picking motors and developing a thruster package to move the AUV which included ESCs, brushless DC motors, and LiPo batteries. Last year, two motors and two ESCs were used. Initially, a 22A brushless DC motor was used because it had similar current draw as the BlueRobotics T200 thruster [3]. The hope was to be able to 3D print both the shroud and propeller in an attempt to obtain similar thrust as the T200 thruster which is 11 lbs forward and 9 lbs backwards. In practice, the most thrust produced by the thruster assembled was about 4-5 lbs both forward and backwards. The mechanical team calculated that it would take about 22 lbs of thrust to move at 1 m/s, a number similar in performance to winning teams, and the thruster configuration would not meet the specification of two thrusters per axis producing 10 lbs of thrust total) This lead to the decision to use both a 60A motor and ESC in an attempt to produce more thrust, which it succeeded in by producing about 11-12 lbs bidirectionally. While this achieved the total thrust goal, it would come at a cost of needing more batteries to power the thrusters. The thrusters however would not be tested on the chassis design due to time constraints.

In addition to thruster design and packaging, the power and propulsion subteam worked on a PCB layout to route power and communications as well as a battery management system (BMS) to balance battery cells and disconnect the battery from the system. A circuit for the main PCB was designed but wasn't placed on a PCB due to the ongoing part selection within both the propulsion and power subteam as well as the vision and sensors subteam. The battery management circuit was designed around a pre-existing circuit design found in the white papers for BMS controllers manufactured by Texas Instruments which the BMS to be made by the subteam also would implement. The design went through major changes due to the size of motors changing. Both of these tasks would go on as tasks to be picked up by the 2017-2018 RoboSub team.

Vision and Sensors Subteam

The 2016-17 Sensors began work on implementing hardware for the AUV. They purchased the Intel Joule 570x as the image processing unit due to its compact form factor and ability to interface with many components. They also chose to use a GoPro Hero 4 as the eye of the AUV. A Raspberry Pi model 3 development board was purchased to implement the pressure transducers. The pressure transducers were donated to the team at the beginning of the year by Gene Fatton. The pressure transducers are an analog sensor so a high precision analog to digital converter was purchased. The 2016-17 sensor subteam had investigated and purchased several different IMUs in addition to the other hardware. They soon discovered that the quality of the IMU was crucial to the overall end goal dead reckoning. This was due to the drift of the IMU and will be covered further later on in the paper. This is what led them on the Spartron AHRS-8 military grade IMU. In addition to deciding on an IMU, they also built a modular code which can connect to any IMU. The only addition to the code that is needed when adding a new IMU is the specific connectivity of that IMU. This will enable future team to purchase other hardware and implement it with ease.

Controls Subteam

The controls subteam was not a part of the project. As they were doing more proof of concept and proof of idea which didn't get to the stages of integration of the different subteams. That brought about the need for a couple more members of the team and the creation of a new subteam. Throughout this semester our two members have worked with the propulsion and sensors subteams to get a better understanding of what each team was working on.

We have also worked with the basic idea that our primary control board would be a Raspberry Pi because of the simplicity of use. The ability to connect each one of these systems easily and reliably was a major factor in us selecting not only our control board, but our method of communication through the AUV as well. The ease of connecting the hardware components using ethernet routed through an ethernet switch resolved all of the needs we had for hardware communication in the AUV.

Chapter 3 - Mechanical Subteam

Testing Conducted on Previous Designs

The Mechanical subteam conducted testing on the previous year's AUV design from 9 September 2017 to 15 September 2017, attaining thermal, hydrodynamic, usable space, and practical usage data. Two detailed thermal tests were conducted in water, once with hot objects placed into the AUV with tide flowing over the body of the sub for 20 minutes, and once with heat generating hand warmers along with hot objects placed into the housing in attempt to simulate the effect of operating electronics. Another less detailed thermal test was conducted while also testing motors on the previous year's design. After roughly four minutes under water one minute of which the motors were run at various speeds, the AUV was pulled from the water and the internal air temperature and ESC surface temperatures were measured with an IR thermometer gun.

During the detailed tests it was found that the internal temperature of the submarine decreased at a slower than desired rate (less than one degree Celsius per minute) and the gradient of temperature from one end of the housing to the other was the inverse of what was expected by the previous mechanical team. The previous team designed the aluminum endcap to be a heatsink for the entire housing. The results of the testing revealed though that the air near the aluminum endcap was two to five degrees Celsius warmer than the far end of the housing, made of thin PVC tubing. This indicated that the heatsink was not operating as intended and that the housing was not dissipating heat at a safe rate. The results of the IR thermometer test after motor testing showed an internal air temperature in excess of 47 degrees Celsius and an ESC surface temperature of over 85 degrees Celsius, a temperature that caused the ESCs to temporarily cease to properly function, and a temperature above the operating range for several circuits that would be included in the functional AUV.

The hydrodynamics tests conducted consisted of a pull test, connecting the housing and frame to a tow strap and measuring the force required to pull the sub and recording the pull speed, and a motor test, connecting two of the previous team's designed thrusters to the frame and driving them for one minute out of four minutes spent in the water. The pull tests showed that in order to reach a speed of 0.6 m/s forwards nearly 30 lbs of thrust would be required, and to reach the same speed sideways it would take nearly 50 lbs. The motors constructed by the mechanical team that year were only able to produce about five pounds of thrust each. When tested with the motors attached in a pool and run at full throttle, the AUV was able to travel at a speed of approximately 0.4 m/s, which was significantly below the desired speed of 1m/s. Fluid flow simulations were conducted in SolidWorks and severe vortex generation was discovered within the exoskeleton, presenting itself as a likely cause of the severe drag forces.

During the testing some usable space and practical usage concerns also began to emerge. Initially the drawer slider design was developed to increase ease of access to internal components. However, the drawer slider system was placed into a tube with too little space to allow most components to cleanly slide in, and due to the brackets used to hold them in place, there was surprisingly little internal volume. Installing and removing the sealed endcap proved to be exceedingly difficult, and oftentimes the act of installing the endcap would result in minor injuries to the team members and an improper and unsafe seal. And if the endcap was not installed properly the act of removing it would be even more difficult and

dangerous than installing it. Using the competition rules and the results of the above tests the following requirements list was created.

Design Objectives and Constraints:

For the 2017-18 school year the mechanical team developed the constraints for the physical systems of the AUV as such:

Housing:

1. The housing shall be watertight to withstand pressure at a depth of 20 ft. from the surface for a minimum of 20 minutes
2. The housing shall weigh no more than 50 lbs. when dry
3. The housing shall dissipate heat in order to maintain an internal temperature of no more than 60 degrees C
4. The housing shall withstand impacts with solid surfaces when at a speed of 1 m/s
5. The housing shall fit all electronics such that they are visible and easily accessible when the housing open
6. The housing shall be constructed to hold or connect to all submarine systems

Torpedo Launcher:

1. The torpedo launcher shall launch the projectile a minimum of 36 in.
2. The torpedo launcher shall launch the projectile in a straight line within a 4 in. radius of the center axis of the launcher
3. The torpedo launcher shall generate a recoil that pushes the submarine no more than 6 in. in any direction
4. The torpedo launcher shall withstand an internal pressure of at least 60 psi

Ballast System:

1. The ballast system shall create neutral buoyancy within a margin of 0.1%
2. The ballast system shall cause the submarine to sink
3. The ballast system shall be watertight
4. The ballast system shall fill its volume with water 100%
5. The ballast system shall drain its volume a minimum of 90%

Manipulator Arm:

1. The manipulator shall grasp an object without definite edges and a handle
2. The manipulator shall grasp and manipulate golf balls
3. The manipulator shall sense when it has grasp of an object
4. The manipulator shall store golf balls in a predictable way on its extremity. OR The manipulator shall place golf balls in a predictable way in a storage device
5. The manipulator shall retrieve a specific color of golf ball without needing to manipulate another golf ball first

6. The manipulator shall have the structural integrity to push and pull levers with a maximum applied force of 15 lbs
7. The manipulator shall have joints that have a maximum increase of friction force due to load of 20%, summed over all joints
8. The manipulator shall have the modularity to grab, store, retrieve, and place golf balls as well as pull and push levers and plates, without the aid of attitude adjustments to the submarine
9. The manipulator shall have the modularity to push and pull tasks with minimal aid by the submarine position, maximum of 1 foot of submarine housing displacement
10. The manipulator shall have the modularity to place golf balls in bins with their brims breaching the surface of the water with a maximum aid in attitude adjustment of the submarine housing of 90 degrees about any axis
11. The manipulator shall be long enough to breach the water and place golf balls in bins that breach the water
12. The manipulator shall collapse in a way that it does not extend more than a maximum of 2 inches away from the bottom of the component housing
13. The manipulator shall collapse in a way that it stays within the area defined by the length of the bottom of the component housing and 8 inches centered on the width of the bottom of the component housing
14. The manipulator shall collapse in a way that does not impede the function of the bottom housing propeller

Work Completed Fall 2017

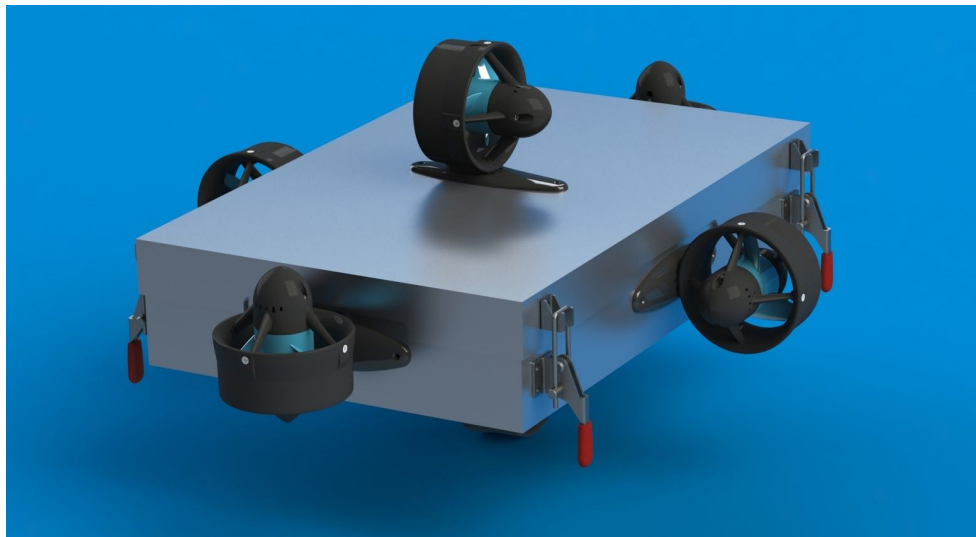


Figure 1: CAD Rendering of Work Completed Fall 2017

With respect to these defined requirements the mech team prioritised work on the electrical housing and chassis to be completed first. It was decided early on that the electrical housing and submarine chassis would be the same component to save weight and volume, meaning that all components attached to the AUV would be connected to the same body that ensured water stayed away from electronics. Throughout the Fall 2017 semester the mechanical team worked through several design concepts for the electronics housing, initially building hydrodynamic smoothing elements into the design

of the chassis, but eventually realizing that a rectangular prism design would be the most efficient way to minimize weight and maximize volume while maintaining a low buoyancy percentage. The dimensions of 12 inches wide by 18 inches long by 3 inches tall were decided on, as they would create a volume that displaced enough water to create a gross buoyant force of around 35 lbs when accounting for the volume of motors. To ensure the internal volume and ease of access requirements were met a simple lid design was developed, allowing the entire top surface of the rectangular prism body to be removed to access the electronics.

A basic seal was developed from simple off-the-shelf components and a plexiglass window was glued into the lid. This basic seal design worked for 17 hours at about one foot underwater, but upon consultation with professional engineers we came to the conclusion that this seal design would fail under deeper water pressures. The plexiglass window failed pressure testing, as it would be subjected to over 600 lbs of force at 16 feet of depth underwater, and it came delaminated when one of the Mechanical subteam members started to transfer their weight onto it. A second design of the lid was started at this point, constructed entirely from aluminum, to increase heat transfer and increase pressure resistance. Plug welds were used to create blind holes to mount motors and lid attachment latches creating, essentially, a continuous piece of aluminum. High-density, closed-cell neoprene foam was used to create the seal for the lid. Originally hand cut for the first test, and then water jet cut for better precision and less risk of structural compromise.

Chapter 4 - Power and Propulsion Subteam

As stated earlier, it is the job of the power and propulsion team to take commands from a Raspberry Pi and generate the desired duty cycle to control motor speed using ESCs. The propulsion system must also be able to monitor itself through the use of Hall Effect sensors and alert the system if the motors are drawing too much current, which will drain the battery faster. The other role of the subteam is the power portion, meaning the subteam must find a way of powering all hardware within the vehicle. However, before delving into these aspects the subteam had to address thruster and ESC choices to correlate with the new chassis.

Part Modifications

This year, the propulsion subteam has decided to buy a prebuilt thruster package due to our decision of moving onto tasks that required more attention such as motor control and current monitoring. The propulsion subteam decided to purchase a T200 Thruster package, which is a water-grade 22A motor that is capable of providing 11 lbs of thrust moving forward and 9 lbs of thrust moving backward. Moreover, this thruster package has a prebuilt shroud as well, so we don't have to worry about making a custom shroud. The thruster package is shown below in figure 2.



Figure 2: T200 Thruster Package [3]

During the beginning of the semester, we had a chance to test out the old AUV design using the older thrusters. During that test, we were hardly able to move the AUV underwater faster than a speed of 0.5m/s due to its massive weight constraints. Moreover, we had run the AUV for a grand total time of 70 seconds before the ESCs reached a temperature of 180 degrees fahrenheit. This semester, we were able to test out these motors on the new chassis design at the fieldhouse pool at CSU. The motors worked wonderfully as we were able to move the AUV at a faster speed (around 2m/s) underwater as compared to the older thruster and chassis design. Moreover, we had a runtime of around 10 minutes and this barely caused the new ESCs to reach a temperature of 80 degrees fahrenheit. This leads me to another change we had for this semester.

This semester, the propulsion subteam switched over to 30A ESC instead of 60A ESCs due to a lower current draw required from the new thrusters. This also helped with reducing heat generation produced from the ESCs. The 30A ESCs are shown below in figure 3.



Figure 3: HobbyKing 30A ESCs

Current Monitoring

The current monitoring is what will be replacing the BMS this semester. This is going to be done using ACS712 Hall Effect Sensors capable of sensing current upto 30A. They are shown below in figure 4.

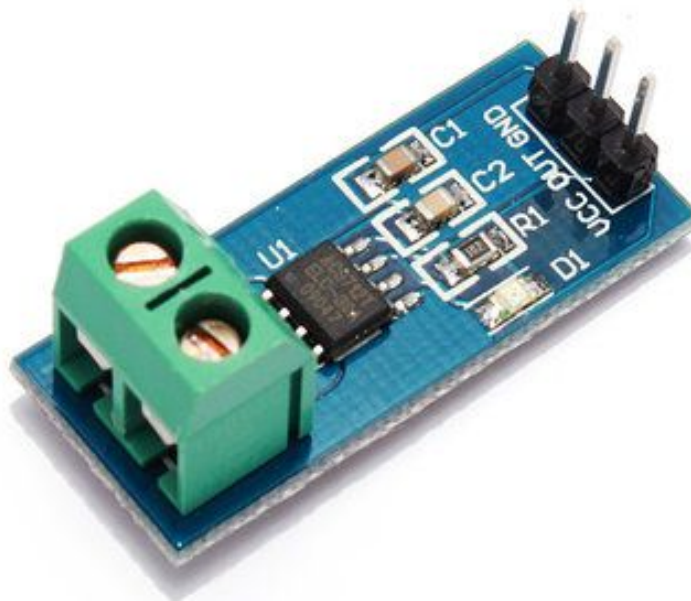


Figure 4: ACS712 Hall Effect Sensors

We are going to be using a total of 6 hall effect sensors to measure the current draw for all 6 motors. The way the Hall Effect sensors are integrated into the circuit layout is by connecting two wires to the green housing shown in the figure above. They are basically acting as a high current ammeter to sense current flow. The Hall Effect sensors will then send the sensed current drawn by the motors using the analog pin, labelled “OUT” in the figure above, which is in between the pins labelled VCC and GND. The hall effect sensors will be continuously sending the data collected from all 6 motors to the Arduino every second so that the arduino knows exactly how much current is being drawn at all times. Moreover, if the amount of current drawn by the motors exceeds 10A, then the Arduino will be able to alert the Raspberry Pi to reduce the amount of thrust so that the current draw is decreased and also to prevent the battery from being drained too quickly.

These Hall Effect sensors have an offset preset so they require a few lines of code so that we can take the sensed value and display the value in Amperes. Moreover, the values that were initially sensed had a lot of fluctuations between them. We were getting values that had a range of 0.5A. In order to reduce this error, we had to implement code that would take multiple samples and then display the average current. This coding was done using the Arduino that will be used by the propulsion subteam for motor control. The way in which we initially tested the Hall Effect sensors was using a power resistor device connected in series with an ammeter and the hall effect sensors. During this test, we were limited by the ammeters ability to only measure current up to 2A, but we were still able to get good results. This helped bring down the fluctuation range down to 0.1A which was a big improvement.

Motor Control

While the primary objective of RoboSub is to build a fully autonomous vehicle, a couple of methods for motor control were developed to test and validate the new design of the chassis. These methods of motor control will be referred to as Direct Motor Control and Indirect Motor Control.

Direct Motor Control

Direct motor control involves controlling the AUV by some means of human intervention, or in this case, controlling roll, pitch and yaw movements via a gaming controller. This method of control was essential to getting the AUV in the water for drive testing as quick as possible. Using the gaming controller, this allowed for tests to be carried out such as observation of AUV drive performance and how much heat is being generated by the electronic speed controllers (ESCs) inside the body. After one pool test, there were observations of quick movement and low heat generation, however there were also issues with AUV stability. These issues will be addressed in the coming semester.

Indirect Motor Control

Indirect motor control is a method of motor control and actuation using only onboard hardware devices. This method of motor control involves the use of an Arduino Mega acting as a *slave* controller that will continuously receive a string of 7 values designating motor state and speed. The first value of that string, “motor state” refers to one of three states, either *Disarmed*, *Armed* or *Error*. The flow of control for these states can be seen in Figure 5: Motor State Diagram. The purpose of the motor state is to specify when motors will be active, or inactive. The remaining six values contain a pulse width modulated (PWM) signal designating a speed to each motor. The *Disarm* state is recognized by a 0 which means the motors receive power, but does not pass through any PWM signals to ensure the AUV is not yet moving. The “Armed” state is recognized by a value of 1 and supplies both power and PWM values to allow for the movement of the AUV to commence. Finally the *Error* state is presently recognized by a 2 meaning all power needs to be cut off from the motors and no more PWM signals will be accepted. After the motors have gone through a *reset*, they proceed back to the *Disarmed* state until the master controller sends more information for movement. The *Error* state involves the use of hall effects sensing in which a maximum threshold current has been detected by one of the hall effects sensors and thus notifies the *Master* controller to send an error state to the Arduino and go into the disarmed state. This flow of control can be noted by Figure 6: Motor Control Diagram.

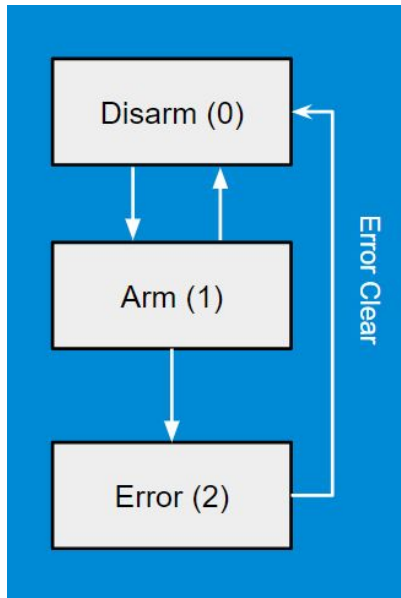


Figure 5: Motor State Diagram

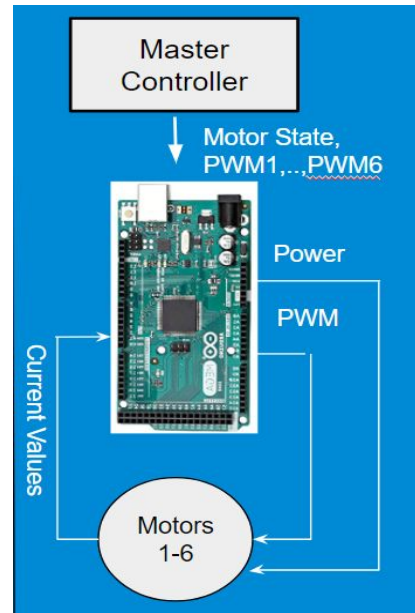


Figure 6: Motor Control Diagram

Chapter 5 - Vision and Sensors Subteam

The sensor team primarily programed in python and MATLAB. Python was used for all finalized code due to its versatility and for its extensive addon libraries. A few libraries that were used are, numpy, matplotlib, and opencv. MATLAB was mainly used for prototyping language, as it have many supported filters and easy to implement functions. Github is an online repository that stores just about any language of code. It allows large teams to work on the same code at the same time, which is not always easy as new code can cause faults in older code and with others teammates code. It also allows the general public access to our code which is necessary as several libraries have GNU license. This is why this Github was used for this project. Doxygen was used to meticulously document all of the finalized and prototyping code. It is an open source program that is able to read comments in code and produce a very detailed and organized document of the code in several different formats. This will help anyone who wants or needs to know how our code functions and will help any future team build upon it without breaking it.

Hardware and Interconnects

In order to be competition ready, the AUV needs many sensors to perform the proper tasks. In addition to the actual sensors, the proper hardware to interface these sensors is just as important. The goal of the competition is to complete the required tasks in as little time as possible. The only way to accomplish this is by physically moving the AUV as fast as possible but in a safe manner, and by processing the data from the sensors as fast as possible. Processing data is more in the scope of the sensors team and that is what was focused on.

Main Computer Board

For the 2017-18 year, many changes were made to the sensors hardware. One of the main changes made was the vision processing unit. The decision to change the vision processing computer was based on two reasons. The first was camera related. When the new AUV design was reviled, the option to implement more cameras became available. With the robotic arm and torpedo launcher, the decision was made to use a camera for each. If a camera is used for both the torpedo launcher and the arm, the code written to do our main image processing can be easily used to guide an arm or fire a torpedo. The second reason a new board needed to be chosen was sheer processing power. The goal of the new processing unit was to have something powerful enough to be the central brain, and do all the image processing. The central brain will receive data from all the other modules, make decisions based on that data, and control the proper parts of the AUV. Image processing is computationally expensive and will require a fast GPU for parallelization. With these ambitious goals, the Intel Joule was not going to meet our requirements, and the AUV needed to have flexibility for possible renditions in the future. With that, research was done to determine the best option for our specific AUV. The Nvidia Jetson TX2 development board was chosen for two reasons. The first reason is the computing power; the Jetson boasts two powerful CPUs which isn't usually seen in a development board. The first one being the Nvidia custom designed dual core "HPM Denver", and the second being a quad core ARM A57. The Jetson also has an Nvidia custom designed GPU running the 16nm Pascal architecture introduced in 2016. It has 256 CUDA cores which will greatly speed up image processing computations. It has 8 gigabytes of low power DDR4 RAM and 32 gigabytes of eMMC flash storage for the operating system. The operating system is Ubuntu 16.04 which is a Linux based operating system. This is a huge plus because all the image processing code is being developed on that platform. This board will provide more than enough processing power to do both image processing and controlling. The second reason this board was chosen is because of the I/O available and the expansion capabilities. On the board there are the following ports: Ethernet, SD card, HDMI 2.0, USB 3.0, micro USB, SATA, PCI Express 4x, and up to 6 CSI 2.0 (Camera Serial Interface) inputs. The board is capable of decoding and rendering up to six 4K cameras at 60 frames per second in real time. This board meets and exceeds all the desired specs and even adds expandability for the future. Figure 7 shown below is an image of the Jetson board labeled with the I/O.

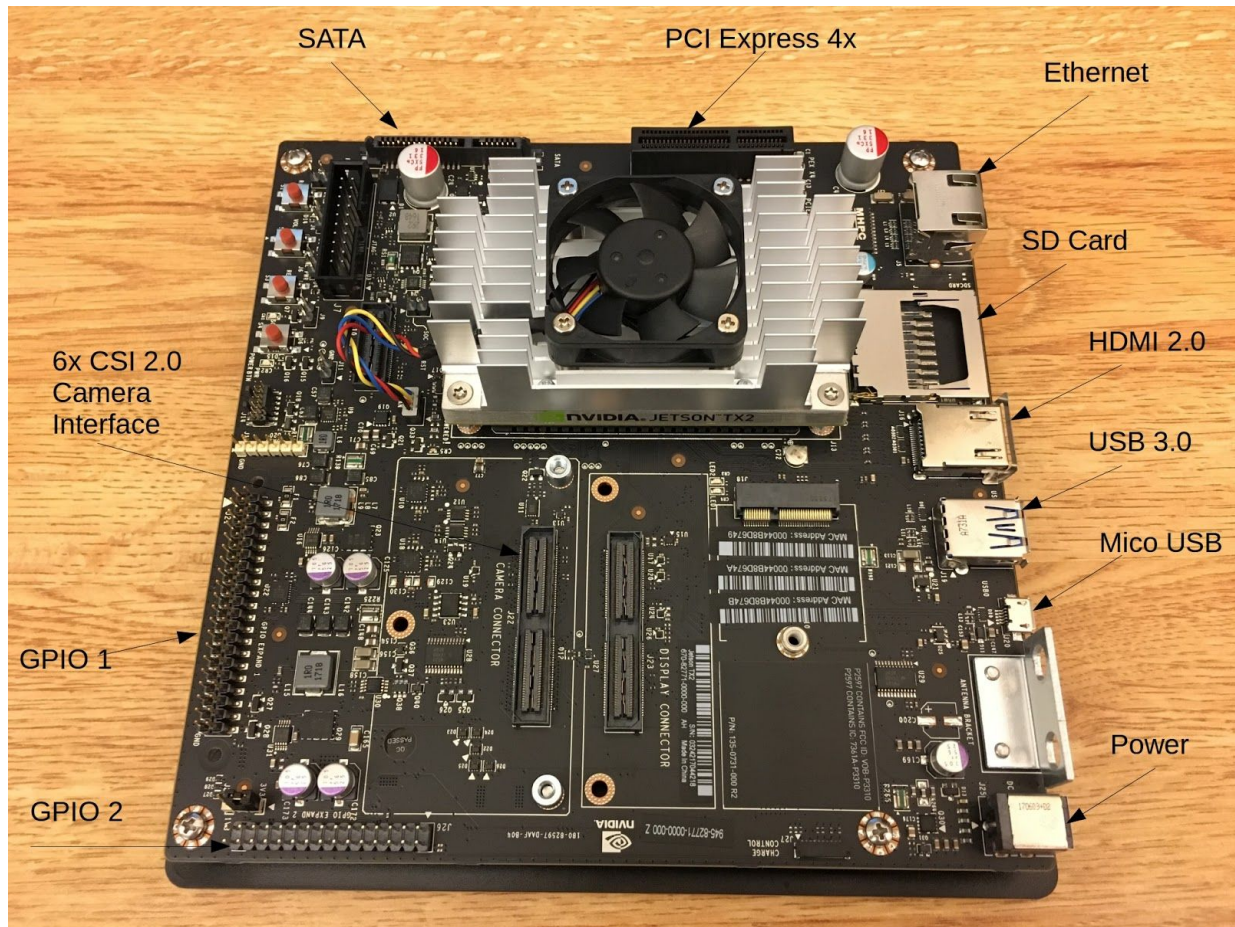


Figure 7: Diagram of Nvidia Jetson TX2

Cameras

With the edition of the Jetson TX2 board, the opportunity to interface with multiple cameras became a reality. The camera from the previous year, the GoPro Hero 4 will be used do to its high quality. Figure 8 is a picture of the camera.



Figure 8: GoPro Hero4 Silver Edition

The GoPro can take video at many different frame rates and resolutions, but a resolution of 1920x1080 at 30 or 60 frames per second will be standardized across all cameras. The other cameras that will be used are the Raspberry Pi cameras. These cameras are very small, inexpensive, and can still take video at 1920x1080. The cameras come equipped with a CSI 2 connector out of the box. Figure 9 below shows a picture of the camera.



Figure 9: Raspberry Pi Camera

The Jetson board interfaces the cameras via a camera serial interface (CSI) using a proprietary connector. In order to connect the cameras correctly, additional hardware is needed. A company called Auvideo makes connectors and expansion boards specifically for the Jetson and they created a connector that interfaces between the proprietary connector and the CSI 2 connector. Because the Jetson can interface with up to 6 CSI cameras, the J20 has 6 CSI camera connectors on it. Figure 10 shows the top and bottom of the J20 taken from Auvideo's website [2].

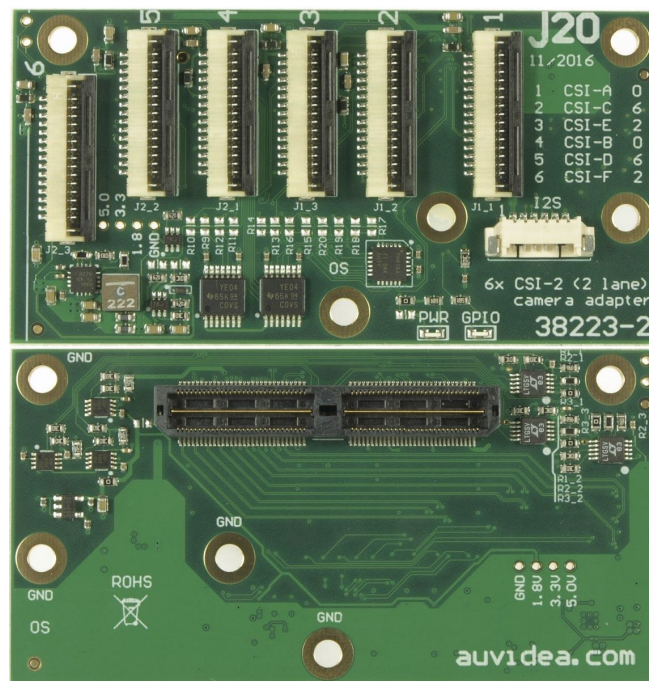


Figure 10: J20 connector for Jetson board

With the proprietary connector and the CSI 2 bridge, the GoPro causes a problem. The GoPro uses a mini HDMI Port to stream video. Because the GoPro is high quality, it will be a valuable asset on the AUV. With this, another adapter is needed. The same company that makes the J20 adapter has another solution to our problem. The B101 HDMI to CSI-2 bridge is that answer. This adapter takes an HDMI input and outputs video as CSI-2. This can feed into the J20 adapter and into the Jetson board. Figure 11 shows the B101 carrier board.

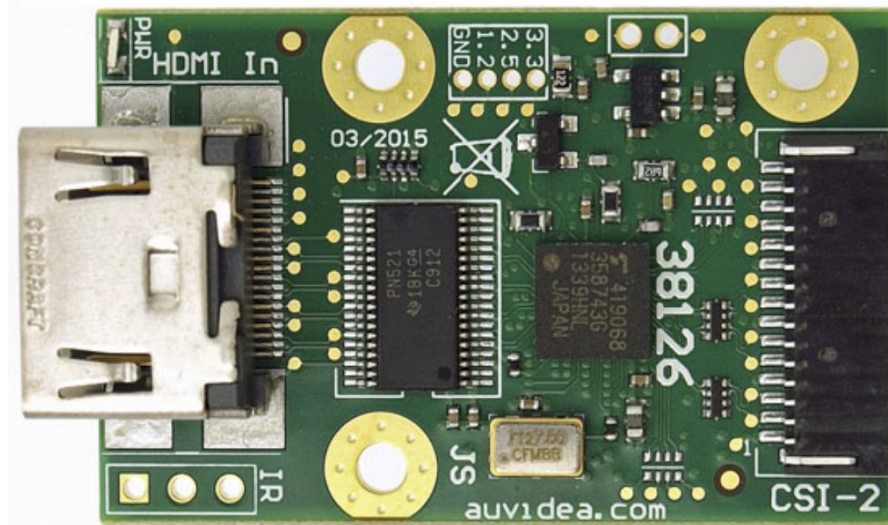


Figure 11: B101 HDMI to CSI-2 converter

Pressure Transducers

One of the main things the sensors team needs to know is depth. To do this, a piece of hardware is needed called a pressure transducer. These take in an analog input based on the surrounding pressure and output a voltage based on that pressure. These are very common instruments used in submarines, drones, and even commercial boats. To interface a pressure transducer, an analog to digital converter is needed. With that, a carrier board is needed to transfer the data to the Jetson module. A Raspberry Pi module and a high precision analog to digital converter carrier board were used to interface with the pressure transducers. The voltage output from the pressure transducer is 0-100mV. This will be amplified to VDD of the raspberry pi (5 volts) so that when the analog signal is converted to a digital signal, the most resolution is present. A TI LF412 JFET op amp will be used to accomplish this task. Figure 12 below shows the pressure transducers hardware.

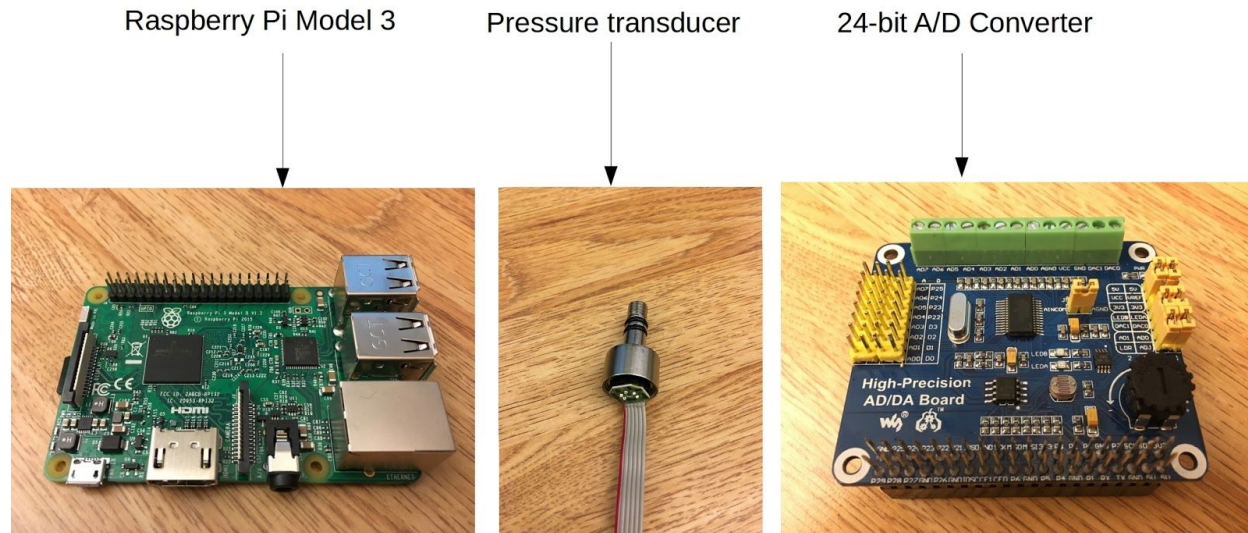


Figure 12: Pressure transducers hardware

Vision

The primary task of the vision team is to bridge the gap between how a human sees and makes decisions, and how a computer sees and makes decisions. It is important to understand how the sub is going to identify objects. This is the information that will play a crucial role in completing a given set of tasks. An object is something that carries a definition and a certain set of attributes. These attributes must be identified by the sub and assigned to an object such as a buoy, gate, or target. This can be done by thinking about how human beings identify and classify objects. From a young age, humans are taught basic shapes and colors, which helps us to identify our surroundings. For example, it is important to analyze what makes us identify a square as a square, or what makes us identify red as red. In order for the sub to be able to recognize these things, code developers must first “teach” it the definitions of certain shapes and colors. This is a method of machine learning called “guided” learning. With an arsenal of predefined knowledge about what could describe what it is looking for, the sub will be able to receive and carry out certain objectives. It will also be able to provide feedback to the sub about what it “sees” or has accomplished. The primary attributes used to describe what the sub sees are shape, color, and position in the frame. The algorithms used to obtain this information are stored as functions, and tests are built to ensure that the tests receive and return information that is useful.

Object Detection

The object detection algorithm determines whether an object likely appears in the frame, as well as a region of interest for that object. Applying the cross-correlation, the algorithm proposed efficiently calculates an indicator determining the likelihood of an object (such as a circle) match, and a region of interest in the form of an object center and the size of the object in NxM (column by row) pixel format.

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

Figure 13 : The Normalized-Cross Correlation

The algorithm requires simply a grayscale image. Slight image processing in the form of edge detection is assumed to be used in the grayscale image. To compensate for variances in the intensity in the image, the normalized-cross-correlation, as seen in Figure 13, handles the computation, producing correlation values between 0 and 1, where 1 indicates a perfect match. A template or mask, T , tests for an object in the input image or environment, I . The mask can be thought of as what the sub “knows” a circle to be. For the mask to scan every pixel in the environment properly, a border with a size of half the mask dimensions and intensity of 0 is applied as well.

Designed initially in MATLAB, the algorithm was then re-implemented in Python, taking advantage of the NumPy and OpenCV libraries. Two versions of the algorithm were implemented, one of a static and the other of a dynamic environment. The static version operates on a single environment for each iteration of the object mask, whereas the dynamic version selects an adequately sized subset of the environment based on a first pass of the object mask across the original environment and the size of the mask at the current iteration. An important conclusion is made initially: that if an object were to exist, it must be larger than an arbitrary minimum, and smaller than a maximum size; this guarantees a maximum computation time. For an user defined number of sliced versions of the mask and a range of object size, the regions of interest based on the center of the object and the calculated object size are stored for each stage of iteration, permitting the complete parallelization of the algorithm.

Several conditions, in addition to the dynamic environment selection, minimize the computation time of the algorithm. As indicated previously, the method ensures a maximum computation time, which ends the computation for extreme cases. In order to prevent an overshoot and too much extra computation time, a trailing-stop checks whether or not the current probability indicator is within a percentage (20% for example) of the previously stored maximum probability in that slice of the mask. If a new maximum is attained, it replaces the threshold for the next iteration. The downside of a larger trail is longer computation time, while a small trail might cause a region of interest to be neglected due to random variations. Once a threshold for the correlation value is reached (0.95, for example), a match is nearly guaranteed, the region of interest, with the max correlation value, and the associated object related to those values is stored.

For a 1728x896 pixel base environment input, with 20 slices of a mask from 10% to 200% of the base size, both versions of the algorithm correctly determine the object size and location. However, the static version completes the computation in 66 seconds, while the dynamic version finishes in 7 seconds, without parallel processing or the CUDA technology in the Jetson board. Not only can this algorithm match a mask with an object in an environment, it can pinpoint the location of the object in terms of x and y values in the frame. This information will be crucial for the control systems of the sub for guidance purposes.

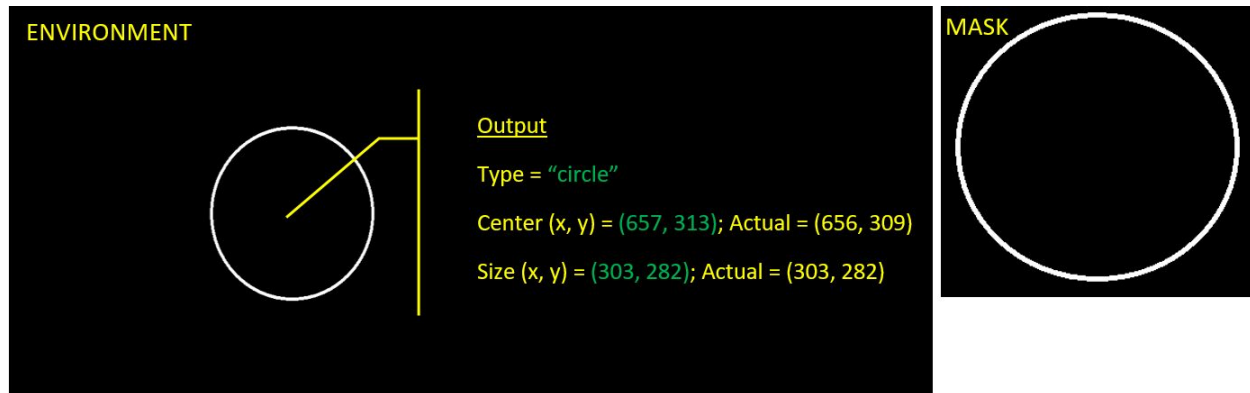


Figure 14: Algorithm output for an idealized-edge-detected circle

By implementing the masks, or objects to test for, in sliced stages, the algorithm parallelizes completely; this allows the full use of the CUDA technology on the Jetson board. This slicing, in addition, provides an extremely simple method for recording multiple types of objects, including simultaneously. While not a current feature, the algorithm promises a relatively straight-forward method for tracking identically sized objects. In its current form, if two objects exist within the same slice of mask, only one is recorded. Increasing the number of mask slices or reducing the range of sizes would alleviate this problem, though not entirely solve it. Instead of vectors, each component of the algorithm's results can be matrices instead and since the same principles in the lower dimension apply, the matrices could track all regions of interest over a threshold probability, including for objects of identical size, but different location.

In summary, the algorithm provides a highly parallelizable method that determines if a specific object exists in the frame, and extracts a region of interest to store object size and increase efficiency for analysis of other parameters of the object, like color. In the future, this algorithm will be modified to match different types of masks, which will be correlated to different types of physical objects, such as lines or targets.

Identifying Color

It is highly likely that a given task will make it necessary for the sub to differentiate color. There are many different color scales that computers use to define colors. Perhaps the most well known is the RGB scale. OpenCV is very powerful image processing software. Some of its capabilities are that it provides useful functions that can convert color scales of images, and use a user defined RGB range to create a spectrum color that is functional in practical application. Once again, the fact that programmers are teaching the sub to be able to recognize certain things comes to the surface. On an RGB scale, pure red is defined as [255,0,0]. However, there are many different shades of red. This must be taken into account because the object that needs to be identified will more than likely fall out of that absolute red scale, and the colors can be distorted from the water and light refracted in the water. This can be addressed by first defining what certain colors are, then creating a threshold of that color. This allows the sub to be able to locate an object in the frame based off of its color. Figure 15 has a variety of colored circles. Using OpenCV, a user defined color (red) can be highlighted, as seen in figure 16.

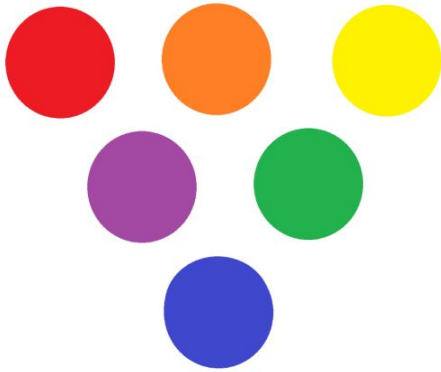


Figure 15: Unprocessed Image

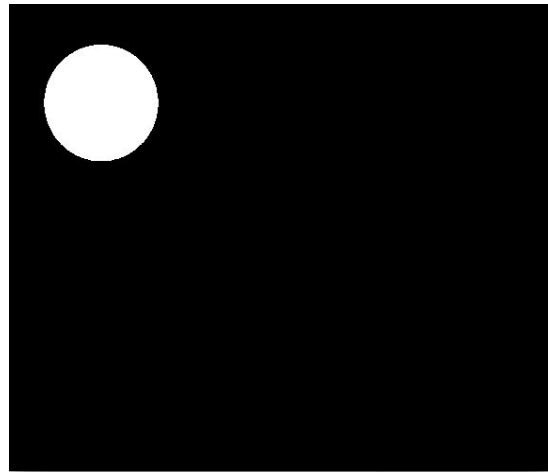


Figure 16: Processed Image

This technique can be used for as many colors as the programmer wishes. Red is simply an arbitrary color, as it is nothing but a number on the RGB scale. Any color can be detected using this method. Another thing that makes this algorithm so useful is how the desired color is turned to white and everything else is turned to black. However, this information must be made useful to the sub. There are many other image processing methods such as fourier transform and gradient detectors. These methods detect drastic changes in contrast, which is very evident in figure 16. This provides another way to calculate the probability that an object exists in the frame, bases on the color the sub is looking for.

The functions that OpenCV can utilize can be used in conjunction with python to calculate the average RGB scale color of a desired frame. The major benefit to being able to calculate the average color in a frame is how the color of the water and the associated light reflection and refraction is “diluted” into the color that takes up the majority of the frame. It can also be used to establish a baseline color that describes the color of an environment given a normalized frame (i.e. there is nothing in the frame worth noticing).



Figure 17a



Figure 17b

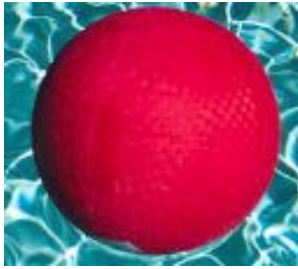


Figure 18a



Figure 18b

Consider Figure 17b, which is the average color of 17a. If this color is established as the ordinary color, Figure 18b, which is the average color of Figure 18a, looks far out of the ordinary. The color averaging algorithm can be combined with the cross-correlation object detection method to assign a color to an object that has already been assigned a shape (mask). Location, shape, and a frame size are returned from the cross-correlation algorithm, and if the average color of that frame is taken, there is enough data to classify an object.

Passing Useful Information to Controls

It is one thing to be able to gather information about everything that was previously explained, but utilizing that information to provide useful information to the control systems of the sub presents a new set of challenges. The defining attributes of the object need to be packaged in a way that can be sent to and interpreted by the control systems so information about corrections or tasks can be sent to the motors, no matter what that task may be. Tasks can include tracking colored buoys, following a line, or shooting a target.

IMU

There are three main goals for the IMU this semester, orientation, feedback, and dead reckoning. Knowing the orientation of the AUV at all times is critical for any movement along with feedback. Dead reckoning, however is not needed for proper AUV movement. It will be an auxiliary function that will hopefully be used to gain more points on the competition.

About the IMU

The IMU that was decided to be used on this AUV, was a military grade IMU that Sparton graciously donated. They also donated a development board along with the IMU and what separates this one from others is the amount of drift associated with the acceleration and gyroscope measurements. The entire setup can be seen in figure 19 .



Figure 19: Spartan IMU with Development Board

IMUs are greatly affected by drift, which is the result of manufacturing, and basic their operating principles. There are better accelerometers and gyroscopes on the market but, are very expensive and often use completely different methods of measuring the acceleration and rotation. Most of these instruments would be infeasible due to size and cost greater than \$100,000. The development board came several different forms of communication including serial over usb. Most modern computers have the capability of serial communication through the USB port which is why this the ideal port to connect to the IMU with. In addition to all of the development board's communication pathways, it also came with several different commands and built in filters. All of these filters, however, incorporate the magnetometer data which will be heavily corrupted by the very large motors surrounding the AUV and therefore cannot be used on the final design..

Feedback

There will be several forms of feedback from the motors and motor commands as previously mentioned. The IMU, however will be able provide a unique type of feedback to the system that others will not. The IMU will be able to detect if the motor command is sufficient to move the AUV in extreme conditions. That is, if there were to be a current pushing the AUV in the opposite direction of desired movement, most other feedback loops will not be able to sense that there is a problem. They will only register that the proper amount of power is being supplied to the motors and the motors are executing the

command. However the AUV will not be moving or will be moving in the wrong direction. The IMU will be able to measure this problem and will be able to set off flags to controls.

Orientation

As previously mentioned, the orientation of the AUV needs to be known at all times. This was achieved with the data gathered solely from the IMU. There are three accelerometers in the device, each representing an X, Y, and Z axis. There is a gyroscope along each axis that measures the angular velocity. The accelerometer and gyroscope data is combined using Mayhony's AHRS algorithm, [4], which produces a rotation matrix. The acceleration data provides a “best guess” of the effect of gravity and the gyroscope data is used to measure any changes in rotation. This rotation matrix contained all of the information that is necessary to keep the AUV upright. The last column in the transpose rotation matrix was the unit vector for the Z axis of orientation with respect to the world axis. The second column was the Y unit vector and the first column was the X unit vector with respect to the world axis. Only the Z unit vector was used in preliminary testing as the other two were not needed at the time. The source code for the algorithm was initially in matlab and was then converted over into python after extensive testing. The rotation matrix was also crucial for dead reckoning.

Dead Reckoning

Dead reckoning is knowing where you are with respect to where you have started. It becomes very complex when implemented with an IMU. This task was done by first changing the acceleration axis into the world axis with respect to gravity using the rotational matrix from Mayhony's AHRS algorithm [4]. The corrected acceleration is then filtered and integrated to obtain velocity. Filtering at each step is necessary to minimize noise in the IMUs measurements. If the data was not filtered before integration, the noise would be amplified and compromise velocity and position data. The code used to do this was based upon xioTechnologies, [1], open source code code in matlab, which only works with full data sets and programmed filters in matlab. The code was then extrapolated to work with a continuous data set and was then transferred over to python.

Chapter 6 - Controls Subteam

Controls Designs and Decisions

Power and Propulsion Side

Part of working with the other teams was creating infrastructure needed for controls to communicate with the propulsion sub team's work. Much of what is about to be described will overlap with the propulsion's description of the system. This however is just focusing on the Controls implemented processes. The propulsion subteam chose to control all of their motors with an Arduino Mega. The Arduino Mega is being used to relay PWM signals to the ESCs as well as sense whether the motors are drawing too much current. The Mega is also responsible for sensing the temperature of the AUV. All of these readings are then interpreted and communicated back to our Raspberry Pi.

With all this information transfer back and forth we needed to make a reliable system to relay communications between our Raspberry Pi and their Arduino. The decision was made to use serial communications to transfer control instructions from the Pi to the Arduino. Serial communications over USB, for our purposes were more reliable, and faster than other communication methods that we pursued.

When working with these serial communications it was important to make a reliable system for which the Pi to send information to the Mega. A system that would ensure accuracy and safety for the AUV and the human operators that would be interacting with it. This is why three states were defined for which the Arduino communications to exist. These states on the controls side are Armed, Disarmed, and Error.

The armed and disarmed states are mutually exclusive. In the armed state, serial communications will consist of motor PWM information. In order to not have the AUV have unpredictable movements if serial communications fail, the mega is designed to timeout and go into its error state if serial communications halt in the armed state. In the disarmed state no serial communications are being sent over serial. When disarmed the arduino is waiting for a command to go into the armed state only. The error state is a little more complex and will be covered next.

Our systems have not developed to a point where there is a need for large amounts of feedback from the Mega to the Pi. Knowing that this would be the case we decided a digital GPIO pin would be used to communicate that there was an error. To elaborate, as the Arduino is reading thermal and current information it checks to see that the values are in an acceptable range. If the values ever exceed what is acceptable the arduino writes to a digital pin which is read by the Pi. The Pi then recognises the error and returns to a control state where it knew that there were no errors. Once the Pi has corrected itself it then clears the error from the Mega returning the entire system to the Armed state.

Vision Sensors Side

Throughout this semester we have worked on implementation of a hierarchy of how our AUV will react to different sensors inputs. The majority of this semester was spent doing research and familiarization with the sensors components. Working with the sensors team we were able to develop a code that showed the output of each one of our sensors. This is a vital piece of the control structure because now we know what package the information delivered to us is going to look like. This piece of

code will allow us to develop the integration of each individual sensor and develop a good functioning test model. Whether the sensor is up and functioning as designed or not, we can test with hard coded data if need be to ensure that our integration of systems is working.

Another part of this semester was used to consider different feedback systems and how we will know if the AUV is actually performing the task that we want it to complete. Depending on the structure of the competition design, we are going to have a various number of different tasks that are recommended until the competition guidelines are set. I recommended to our sensors team to continue working towards the previous competitions tasks because that was where our focus had been all semester already. If we can identify the required objects with vision from a previous year, we should have very little issue modifying the code to recognize the objects required in the next competition. This will help us build a library of performable tasks and identifiable objects. All of these processes will help the AUV, through the instruction of controls, know what tasks need to be done and when it can return to the surface after the mission is complete.

With all of that being said, we are already starting to find the issues that will inevitably show up during competition. Although we are trying to develop ways to correct the possible drift error encountered by the IMU, we are trying to find ways to use our other sensors to correct this issue with input data. For instance; if we use the camera and we see an object directly in front of the AUV, then we should be able to reach the object with a “forward” pulse command. If our forward pulse command puts the target off to the right of the AUV in the next frame, then we should be able to account for the left drift in the AUV’s movement. The other major issue we have thought about is what happens when we detect two different targets? In last years competition there were multiple lines that needed to be followed throughout the competition. This negates my original thought of once the target has been cleared, remove it from the list. So it is going to take a complex algorithm to identify which of the targets is the next one in line.

Chapter 7 - Conclusions and Future Work

Mechanical Subteam

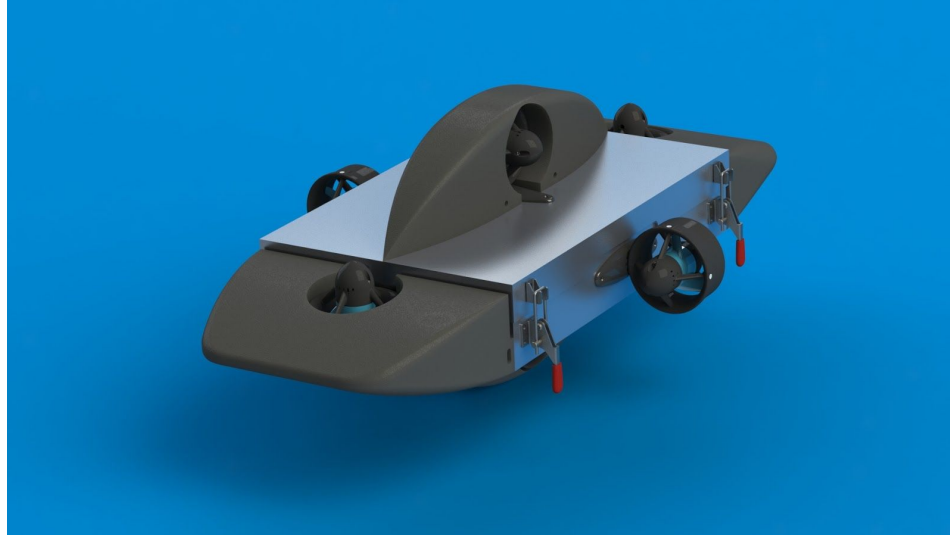


Figure 20: CAD Rendering of Hydrodynamic Fairings

Over the coming semester the mechanical team will first be creating the next revision of the electronics housing/chassis shown in figure 20. The current design still has some microcracks allowing small amounts of water into the housing, and it also has a few unnecessarily large protrusions. Following that the Mechanical subteam will be focusing on designing and prototyping a manipulator arm to meet the demands of the 2018 RoboSub challenge, finishing the design of and casting fairings for the AUV body to further reduce the drag of the vehicle, designing and building a ballast system to better control buoyancy, updating and finishing the torpedo launcher design, and ensuring internal electronics mounting is optimized.

Power and Propulsion Subteam

The propulsion side of the work is drawing to a close, as the circuit and components have been selected and are ready to be implemented by the controls team. Going forward, work will be focused on placing most of the wire connections on a perforated board in order to reduce footprint within the chassis. Once this circuit has been assembled and tested, it can then be made into a PCB. Power will be another point to address next semester as the vision and sensors subteam finalizes hardware choices. This will allow the team to purchase additional batteries to power everything in the AUV.

The subteam will also continually work with the controls subteam to fine tune motor speeds to ensure the best and most efficient movement. This will not only help with extending battery life, but help the AUV achieve better times navigating test obstacle courses and eventually the competition.

Vision and Sensors Subteam

There will be three main goals for next semester. The first is to build a hydrophone array. The competition has several different tasks that need to be complete before the AUV can surface. These tasks

signal their location with an underwater ping. The array will need to be able to hear these pings and differentiate between them as well as know their location relative to the AUV. This is why we don't necessarily need an entire sonar system, just the hydrophones. The current plan is to have four hydrophones that are separated by a known distance, which will allow us to locate, in 3D space, the pinger. The second main goal for the sensor team will also focus on vision as it still needs work before it can be used autonomously. Currently it can only find exact shapes under best case scenario images. That is, a large contrast between the image that we wish to determine and its background, which will not always be the case. We will have to push the system to its limits before it can be reliably useful in the competition. The final main goal is to connect all of the sensors together physically via ethernet and work alongside controls to filter them together. This is due to the fact that there is error in every sensor's measurement that can be reused with other sensors' measurements. These are not trivial goals, will easily take all of next semester to complete if all goes well.

Controls Subteam

As propulsion work draws to a close, Controls will be focusing more on working with Sensors to make infrastructure to make decisions and control the AUV. From the basics of stabilizing the AUV, to the more advanced task of navigating an obstacle course it is the controls task to make that happen. In the coming semester we are looking to get the AUV to accurately stabilize itself and start navigating independently from human control.

References

- [1] "Homepage," *XIO Technologies*. [Online]. Available: <http://xiostorage.com/>. [Accessed: 08-Dec-2017].
- [2] Admin, "New products," *Auvideo*, 30-Dec-2016. [Online]. Available: <https://auvideo.com/>. [Accessed: 08-Dec-2017].
- [3] "T200 Thruster," *Blue Robotics*. [Online]. Available: <https://www.bluerobotics.com/store/thrusters/t200-thruster/>. [Accessed: 08-Dec-2017].

- [4] W. by x-io Technologies, “Open source IMU and AHRS algorithms,” – *x-io Technologies*.
[Online]. Available: <http://x-io.co.uk/open-source-imu-and-ahrs-algorithms/>. [Accessed: 08-Dec-2017].

Bibliography

- “RoboSub,” *RoboSub | Robonation*. [Online]. Available:
<http://www.robonation.org/competition/robosub>. [Accessed: 08-Dec-2017].

Appendix A - Abbreviations

AHRS - Attitude Heading Reference System
AUV - Autonomous Underwater Vehicle
BMS - Battery Management System
CAD - Computer Aided-Design
CPU - Central Processing Unit
CSI - Camera Serial Interface
CUDA - Compute Unified Device Architecture
ESC - Electronic Speed Control
GPIO - General-Purpose Input/Output
GPU - Graphics Processing Unit
HDMI - High-Definition Multimedia Interface
IMU - Inertial Measurement Unit
I/O - Input/Output
IR - Infrared
PCB - Printed Circuit Board
PCI - Payment Card Industry
PWM - Pulse Width Modulation
RAM - Random Access Memory
RGB - Red, Green, and Blue
SATA - Serial Advanced Technology Attachment
SD - Secure Digital
USB - Universal Serial Bus
VIP - Vertically Integrated Project

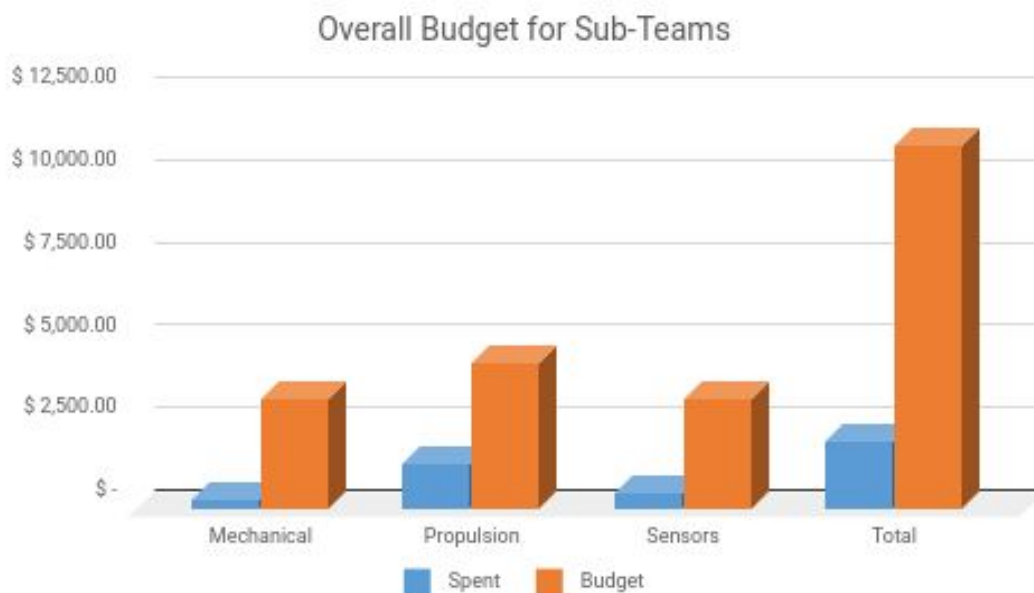
Appendix B - Budget and Finance

Total Funds	\$13,000.00		
Competition Savings	\$2,000.00		
Total Sub Funds	\$11,000.00		
	Spent	Budget	
Mechanical	\$ 231.70	\$ 3,300.00	
Propulsion	\$ 1,311.00	\$ 4,400.00	
Sensors	\$ 447.71	\$ 3,300.00	
Total	\$ 1,990.41	\$ 11,000.00	18%

Note: Each team has a deviation from this total, and will be denoted at the bottom of their respective financial report

*Controls Team expenditures are absorbed into Propulsion and Sensors budgets

The deviation will be used for shipping costs and additional funds for the competition

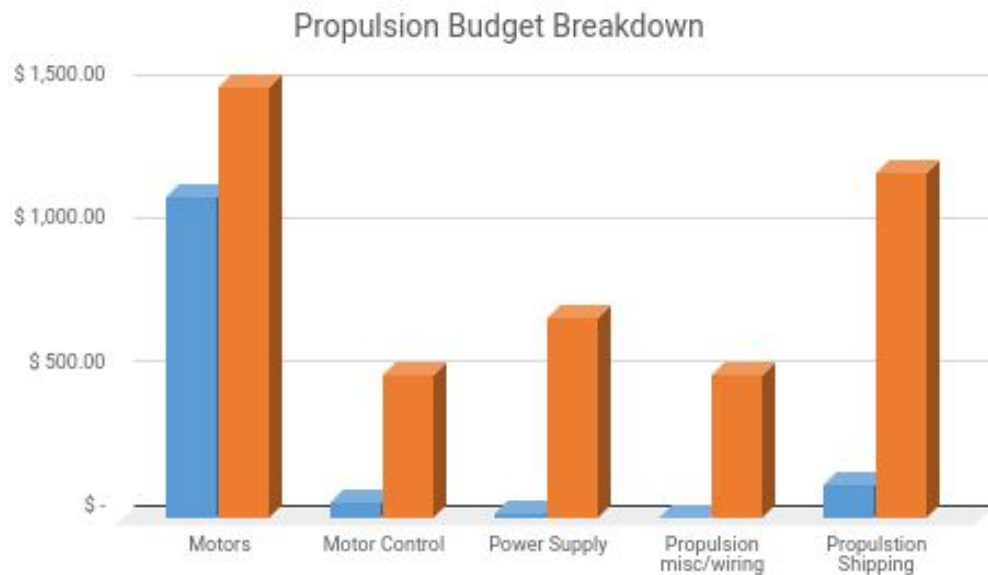


Appendix BI - Budget and Finance - Propulsion Team

	Budget	Spent	Remaining	% Remaining
Motors	\$ 1,500.00	\$ 1,119.30	\$ 380.70	25%
Motor Control	\$ 500.00	\$ 52.98	\$ 447.02	89%
Power Supply	\$ 700.00	\$ 20.47	\$ 679.53	97%
Propulsion misc/wiring	\$ 500.00	\$ -	\$ 500.00	100%
Propulsion Shipping	\$ 1,200.00	\$ 120.38	\$ 1,079.62	90%
Total	\$ 4,400.00	\$ 1,313.13	\$ 3,086.87	70%

Note: Propulsion Deviation is \$1,200.00

Deviation Remaining	\$1,079.62			
---------------------	------------	--	--	--

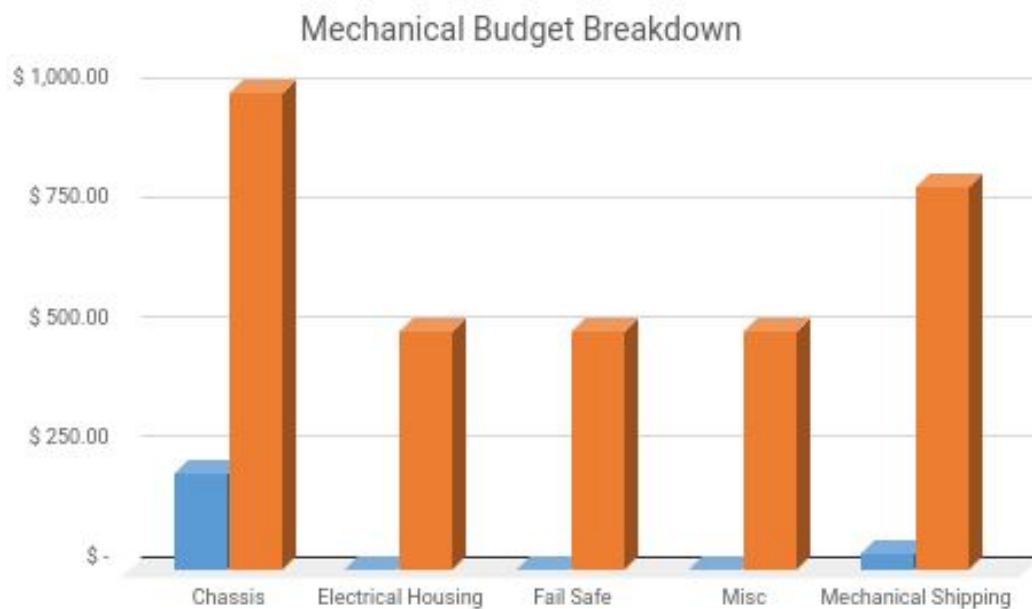


Appendix BII - Budget and Finance - Mechanical Team

	Budget	Spent	Remaining	% Remaining
Chassis	\$ 1,000.00	\$ 199.87	\$ 800.13	80%
Electrical Housing	\$ 500.00	\$ -	\$ 500.00	100%
Fail Safe	\$ 500.00	\$ -	\$ 500.00	100%
Misc	\$ 500.00	\$ -	\$ 500.00	100%
Mechanical Shipping	\$ 800.00	\$ 31.83	\$ 768.17	96%
Total	\$ 3,300.00	\$ 231.70	\$ 3,068.30	93%

Note: Mechanical budget deviation is \$800.00

Deviation Remaining: \$768.17

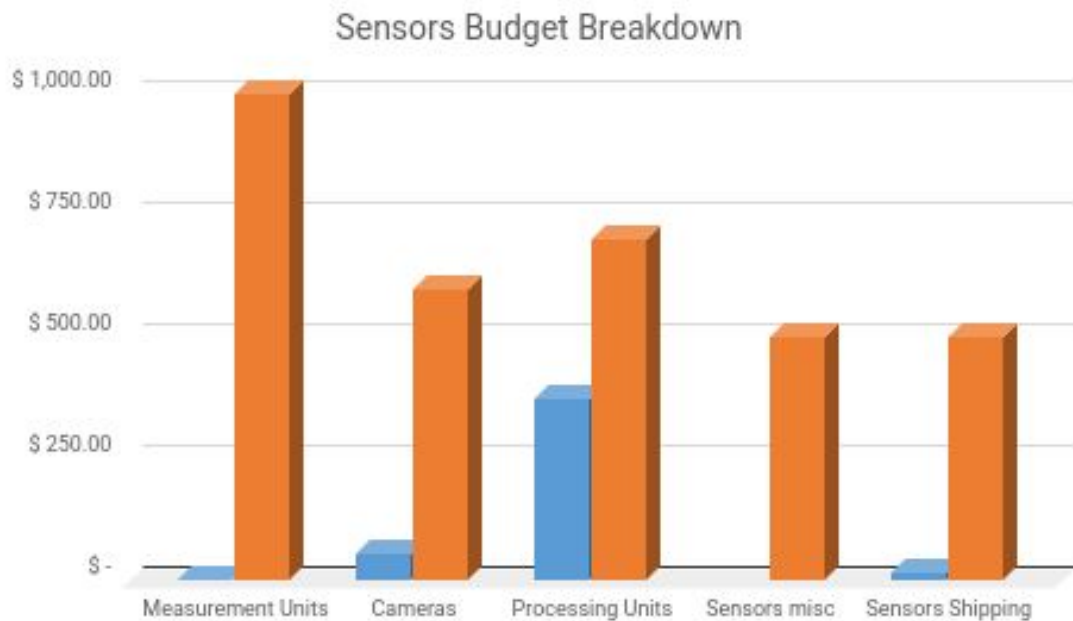


Appendix BIII - Budget and Finance - Sensors Team

	Available	Spent	Remaining	% Remaining
Measurement Units	\$ 1,000.00	\$ -	\$ 1,000.00	100%
Cameras	\$ 600.00	\$ 54.98	\$ 545.02	91%
Processing Units	\$ 700.00	\$ 374.05	\$ 325.95	47%
Sensors misc	\$ 500.00		\$ 500.00	100%
Sensors Shipping	\$ 500.00	\$ 18.68	\$ 481.32	96%
Total	\$ 3,300.00	\$ 447.71	\$ 2,852.29	86%

Note: Sensors deviation is \$500.00

Deviation Remaining	\$481.32			
---------------------	----------	--	--	--



Appendix C - Project Plan Evolution

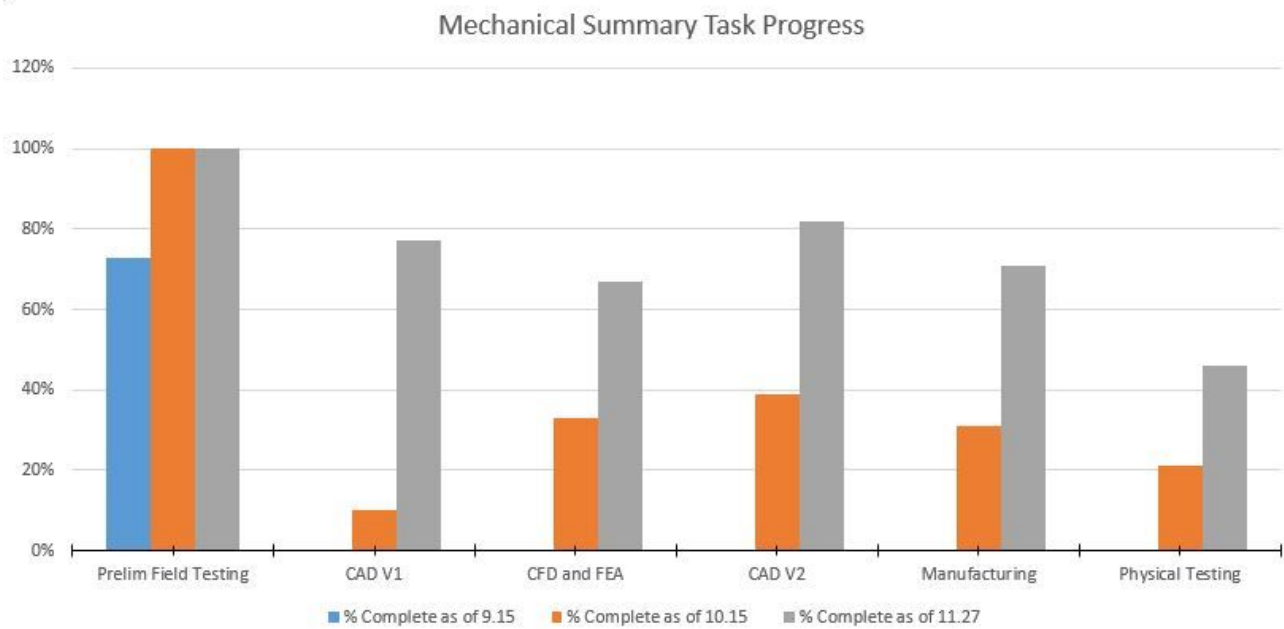
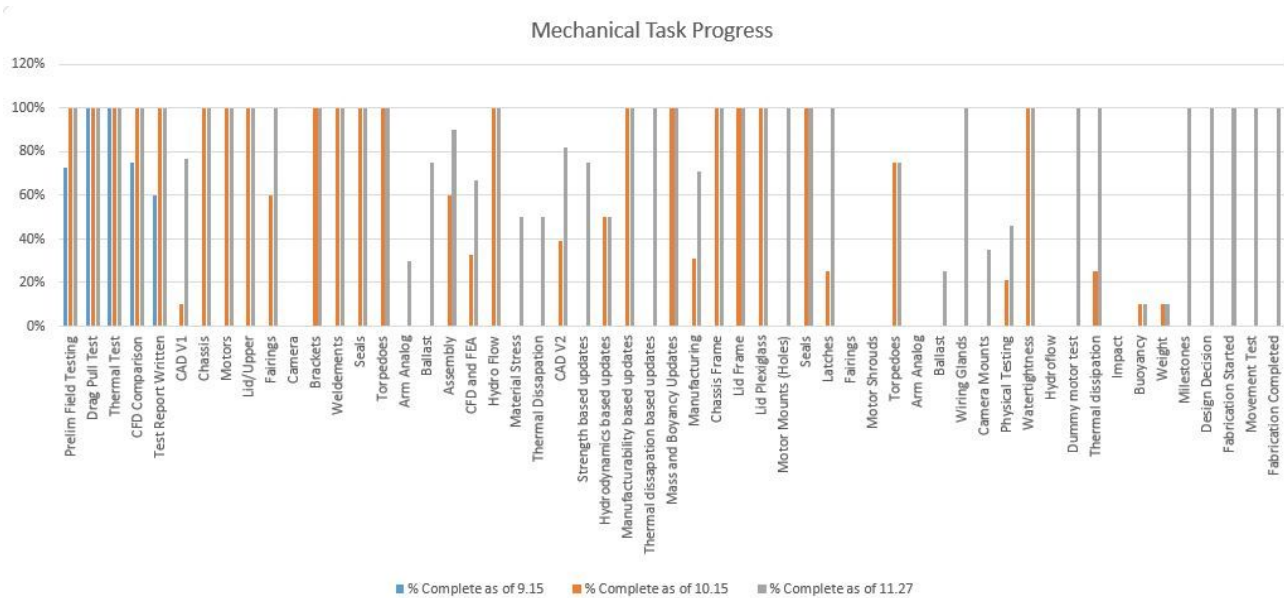
Mechanical Subteam Tasks	Start	Finish	Actual Start	Actual Finish	% Complete
Prelim Field Testing	Tue 9/5/17	Fri 9/15/17	Sun 9/10/17	Thu 9/14/17	100%
Drag Pull Test	Sun 9/10/17	Sun 9/10/17	Sun 9/10/17	Sun 9/10/17	100%
Thermal Test	Sun 9/10/17	Sun 9/10/17	Sun 9/10/17	Sun 9/10/17	100%
CFD Comparison	Tue 9/12/17	Thu 9/14/17	Tue 9/12/17	Thu 9/14/17	100%
Test Report Written	Sun 9/10/17	Thu 9/14/17	Sun 9/10/17	Thu 9/14/17	100%
CAD V1	Wed 9/20/17	Thu 9/28/17	Wed 9/20/17	NA	77%
Chassis	Wed 9/20/17	Tue 9/26/17	Wed 9/20/17	Tue 9/26/17	100%
Motors	Fri 9/22/17	Tue 9/26/17	Fri 9/22/17	Tue 9/26/17	100%
Lid/Upper	Wed 9/20/17	Tue 9/26/17	Wed 9/20/17	Tue 9/26/17	100%
Fairings	Fri 9/22/17	Tue 9/26/17	Fri 9/22/17	Tue 9/26/17	100%

Camera	Fri 9/22/17	Tue 9/26/17	NA	NA	0%
Brackets	Fri 9/22/17	Tue 9/26/17	Fri 9/22/17	Tue 9/26/17	100%
Weldments	Thu 9/21/17	Tue 9/26/17	Thu 9/21/17	Tue 9/26/17	100%
Seals	Wed 9/20/17	Tue 9/26/17	Wed 9/20/17	Tue 9/26/17	100%
Torpedoes	Sun 9/24/17	Tue 9/26/17	Sun 9/24/17	Tue 9/26/17	100%
Arm Analog	Tue 9/26/17	Tue 9/26/17	Tue 9/26/17	NA	30%
Ballast	Fri 9/22/17	Wed 9/26/18	Fri 9/22/17	NA	75%
Assembly	Tue 9/26/17	Thu 9/28/17	Tue 9/26/17	NA	90%
CFD and FEA	Tue 9/26/17	Thu 10/12/17	Tue 9/26/17	NA	67%
Hydro Flow	Tue 9/26/17	Thu 10/12/17	Tue 9/26/17	Thu 10/12/17	100%
Material Stress	Tue 9/26/17	Thu 10/12/17	Tue 9/26/17	NA	50%
Thermal Dissipation	Tue 9/26/17	Thu 10/12/17	Tue 9/26/17	NA	50%
CAD V2	Thu 9/28/17	Fri 10/13/17	Thu 9/28/17	NA	82%
Strength based updates	Thu 9/28/17	Fri 10/13/17	Thu 9/28/17	NA	75%

Hydrodynamics based updates	Thu 9/28/17	Fri 10/13/17	Thu 9/28/17	NA	50%
Manufacturability based updates	Thu 9/28/17	Fri 10/13/17	Thu 9/28/17	Fri 10/13/17	100%
Thermal dissipation based updates	Thu 9/28/17	Fri 10/13/17	Thu 9/28/17	Fri 10/13/17	100%
Mass and Buoyancy Updates	Fri 10/13/17	Fri 10/13/17	Fri 10/13/17	Fri 10/13/17	100%
Manufacturing	Thu 10/12/17	Fri 11/17/17	Thu 10/12/17	NA	71%
Chassis Frame	Thu 10/12/17	Thu 10/19/17	Thu 10/12/17	Thu 10/19/17	100%
Lid Frame	Thu 10/12/17	Thu 10/19/17	Thu 10/12/17	Thu 10/19/17	100%
Lid Plexiglas	Thu 10/19/17	Thu 10/26/17	Thu 10/19/17	Thu 10/26/17	100%
Motor Mounts (Holes)	Thu 10/19/17	Thu 10/26/17	Thu 10/19/17	Thu 10/26/17	100%
Seals	Thu 10/19/17	Thu 10/26/17	Thu 10/19/17	Thu 10/26/17	100%
Latches	Thu 10/19/17	Thu 10/26/17	Thu 10/19/17	Thu 10/26/17	100%
Fairings	Thu 10/26/17	Thu 11/9/17	NA	NA	0%
Motor Shrouds	Thu 10/26/17	Thu 11/9/17	NA	NA	0%
Torpedoes	Thu 11/9/17	Fri 11/17/17	Thu 11/9/17	NA	75%

Arm Analog	Thu 11/9/17	Fri 11/17/17	NA	NA	0%
Ballast	Thu 10/26/17	Thu 11/9/17	Thu 10/26/17	NA	25%
Wiring Glands	Thu 10/12/17	Fri 11/17/17	Thu 10/12/17	Fri 11/17/17	100%
Camera Mounts	Thu 11/9/17	Thu 11/9/17	Thu 11/9/17	NA	35%
Physical Testing	Mon 11/27/17	Thu 12/7/17	Mon 11/27/17	NA	46%
Water tightness	Mon 11/27/17	Wed 12/27/17	Mon 11/27/17	Wed 12/27/17	100%
Hydro flow	Mon 11/27/17	Wed 12/27/17	NA	NA	0%
Dummy motor test	Mon 11/27/17	Wed 12/27/17	Mon 11/27/17	Wed 12/27/17	100%
Thermal dissipation	Mon 11/27/17	Wed 12/27/17	Mon 11/27/17	Wed 12/27/17	100%
Impact	Mon 11/27/17	Wed 12/27/17	NA	NA	0%
Buoyancy	Mon 11/27/17	Wed 12/27/17	Mon 11/27/17	NA	10%
Weight	Mon 11/27/17	Wed 12/27/17	Mon 11/27/17	NA	10%
Milestones	Fri 6/1/18	Fri 6/1/18	Wed 9/20/17	Fri 11/17/17	100%
Design Decision	Wed 9/20/17	Wed 9/20/17	Wed 9/20/17	Wed 9/20/17	100%

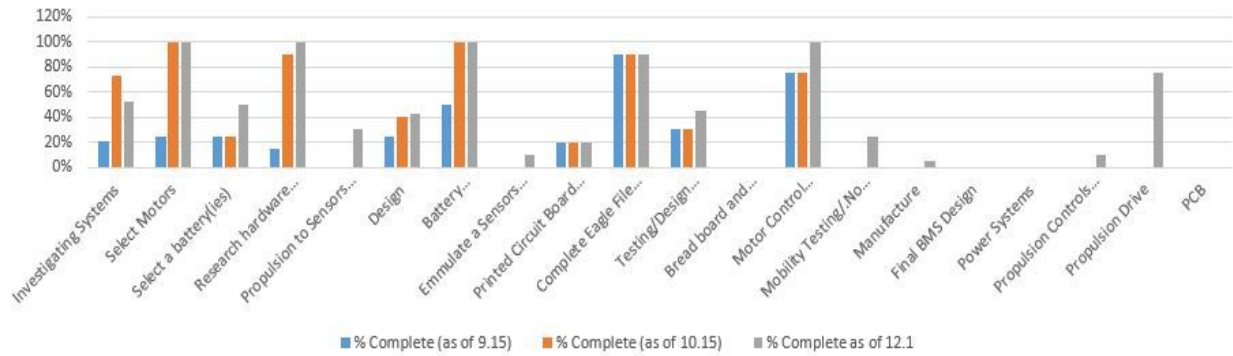
Fabrication Started	Sat 10/7/17	Sat 10/7/17	Sat 10/7/17	Sat 10/7/17	100%
Movement Test	Sat 11/4/17	Sat 11/4/17	Sat 11/4/17	Sat 11/4/17	100%
Fabrication Completed	Fri 11/17/17	Fri 11/17/17	Fri 11/17/17	Fri 11/17/17	100%



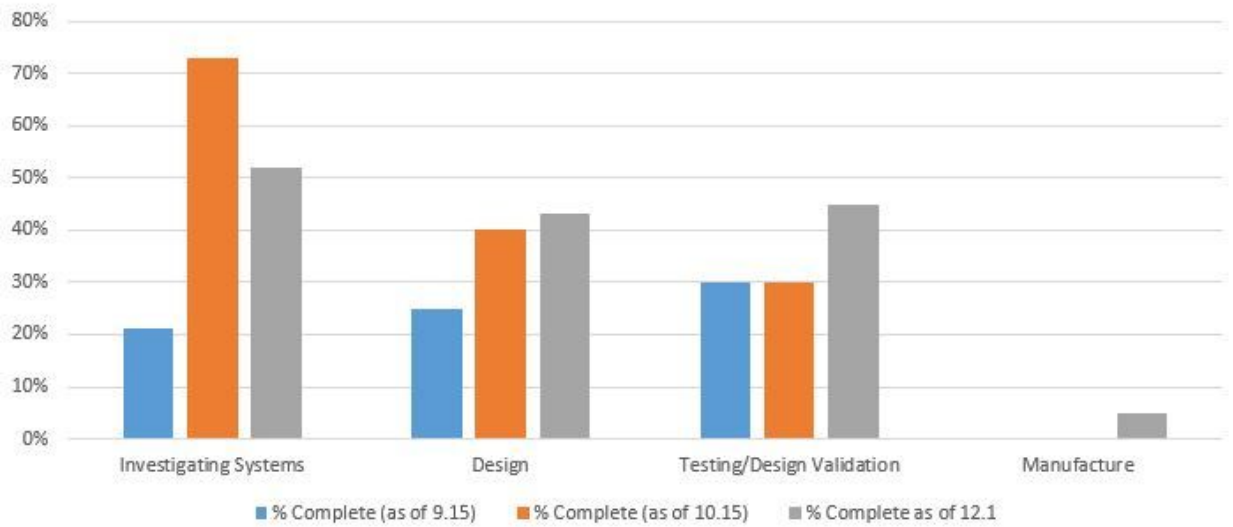
Propulsion Subteam Tasks	Start	Finish	Actual Start	Actual Finish	% Complete
Investigating Systems	Fri 9/1/17	Fri 12/1/17	Fri 9/1/17	NA	52%
Select Motors	Fri 9/1/17	Wed 9/20/17	Fri 9/1/17	Wed 9/20/17	100%
Select a battery(ies)	Fri 9/1/17	Wed 9/20/17	Fri 9/1/17	NA	50%
Research hardware interfacing	Fri 9/1/17	Mon 9/25/17	Fri 9/1/17	Mon 9/25/17	100%
Propulsion to Sensors Communications	Fri 9/1/17	Fri 12/1/17	Fri 9/1/17	NA	30%
Design	Mon 8/28/17	Sun 4/15/18	Mon 8/28/17	NA	43%
Battery Management System Design	Fri 9/1/17	Wed 11/1/17	Fri 9/1/17	Wed 11/1/17	100%
Emulate a Sensor's signal for motor control testing	Fri 9/1/17	Wed 11/1/17	Fri 9/1/17	NA	10%
Printed Circuit Board Design	Tue 1/30/18	Sun 4/15/18	Tue 1/30/18	NA	20%
Complete Eagle File for BMS	Mon 8/28/17	Fri 9/1/17	Mon 8/28/17	NA	90%
Testing/Design Validation	Wed 11/1/17	Wed 11/1/17	Wed 11/1/17	NA	75%
Breadboard and test Battery Management System	Wed 11/1/17	Tue 11/14/17	NA	NA	0%

Motor Control Testing (Mock sensors signal to motors)	Wed 11/1/17	Tue 11/14/17	Wed 11/1/17	Tue 11/14/17	100%
Mobility Testing/.No Communications (Motors mounted to Chassis)	Tue 11/21/17	Sat 11/25/17	Tue 11/21/17	NA	25%
Manufacture	Sat 4/1/17	Tue 5/1/18	Sat 4/1/17	NA	6%
Final BMS Design	Thu 11/30/17	Sat 12/30/17	NA	NA	0%
Power Systems	Fri 12/1/17	Sun 12/31/17	NA	NA	0%
Propulsion Controls and Feedback	Sat 12/2/17	Mon 1/1/18	Sat 12/2/17	NA	10%
Propulsion Drive	Sun 12/3/17	Tue 1/2/18	Sun 12/3/17	NA	75%
PCB	Sat 4/1/17	Tue 5/1/18	NA	NA	0%
Milestones			Fri 6/1/18	NA	0%
Test BMS Design	Tue 11/14/17	Tue 11/14/17	NA	NA	0%
Test Motors on Chassis	Sat 11/25/17	Sat 11/25/17	Sat 11/25/17	Sat 11/25/17	100%
Manufacture BMS	Sat 12/30/17	Sat 12/30/17	NA	NA	0%
Complete PCB Design	Tue 11/14/17	Tue 11/14/17	NA	NA	0%
Manufacture PCB	Sat 4/15/17	Sun 4/15/18	NA	NA	0%

Propulsion Task Overview % Completion
September, October to December



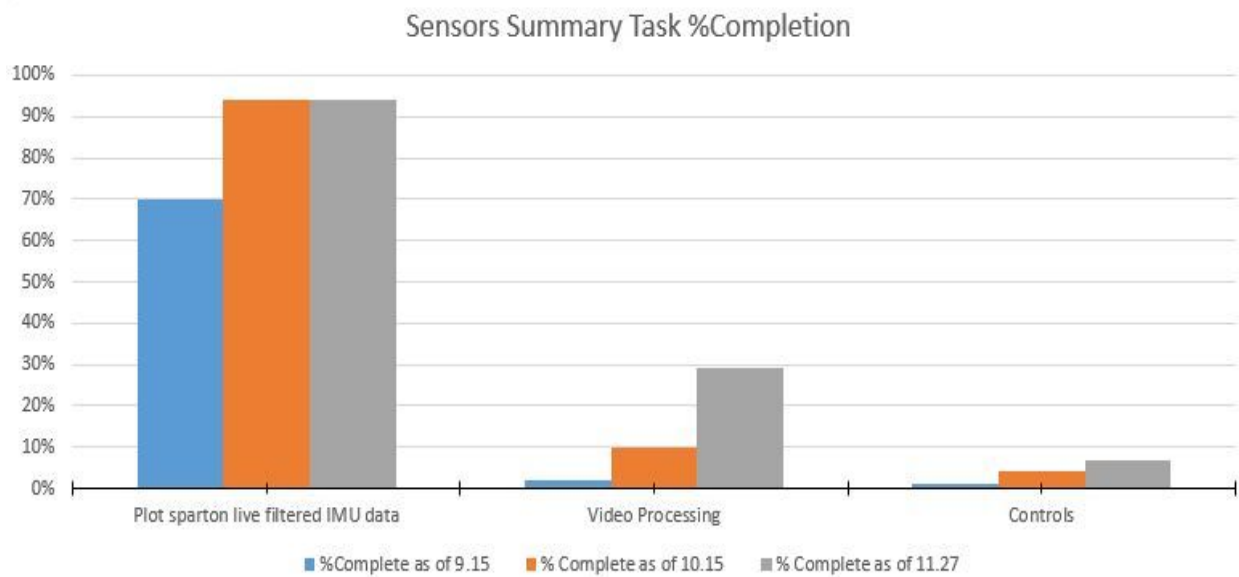
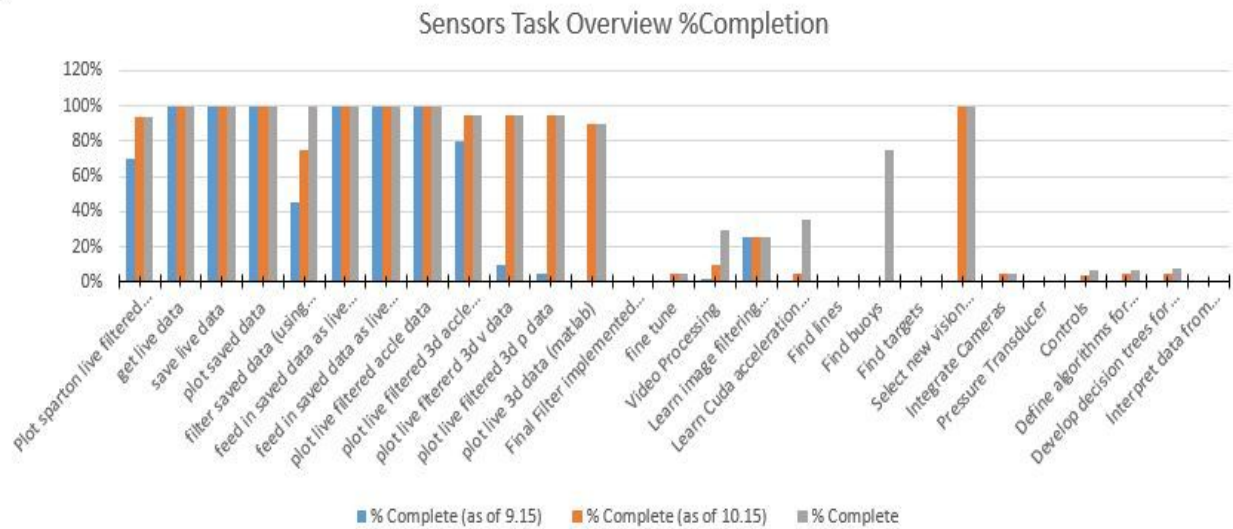
Propulsion Summary Task %Completion
September, October to December



Sensors and Vision Subteam	Est. Start	Est. Finish	Actual Start	Actual Finish	% Complete
Plot sparton live filtered IMU data	Wed 3/15/17	Wed 11/1/17	NA	NA	94%
get live data	Wed 3/15/17	Wed 11/1/17	Wed 3/15/17	Wed 3/22/17	100%
save live data	Wed 3/15/17	Wed 3/22/17	Wed 3/15/17	Wed 3/22/17	100%
plot saved data	Wed 3/22/17	Wed 3/22/17	Wed 3/22/17	Wed 3/22/17	100%
filter saved data (using and comparing different filters)	Thu 3/23/17	Thu 3/23/17	Thu 3/23/17	NA	100%
feed in saved data as live data to a filtering function (python)	Tue 3/28/17	Wed 11/1/17	Tue 3/28/17	Thu 3/30/17	100%
feed in saved data as live data to a filtering function (matlab)	Thu 3/30/17	Thu 3/30/17	Wed 8/30/17	Wed 9/13/17	100%
plot live filtered accel data	Wed 8/30/17	Wed 9/13/17	Wed 8/30/17	Wed 9/13/17	100%
plot live filtered 3d accel data	Wed 8/30/17	Wed 9/20/17	Wed 8/30/17	NA	95%

plot live filtered 3d v data	Wed 8/30/17	Wed 11/1/17	Wed 8/30/17	NA	95%
plot live filtered 3d p data	Wed 8/30/17	Wed 11/1/17	Wed 8/30/17	NA	95%
plot live 3d data (matlab)	Wed 8/30/17	Wed 11/1/17	Wed 8/30/17	NA	90%
Final Filter implemented into python	Tue 10/17/17	Wed 11/1/17	NA	NA	0%
fine tune	Fri 10/13/17	Fri 10/27/17	NA	NA	5%
Video Processing	Wed 9/6/17	Sun 12/10/17	NA	NA	29%
Learn image filtering techniques	Wed 9/20/17	Fri 10/20/17	Wed 9/20/17	NA	25%
Learn Cuda acceleration for processor stress relief	Sun 9/24/17	Tue 10/24/17	Sun 9/24/17	NA	35%
Find lines	Mon 10/2/17	Sun 12/10/17	Mon 10/2/17	NA	0%
Find buoys	Mon 10/2/17	Sun 12/10/17	Mon 10/2/17	NA	75%
Find targets	Mon 10/2/17	Sun 12/10/17	Mon 10/2/17	NA	0%
Select new vision processor unit	Wed 9/6/17	Wed 9/27/17	Wed 9/6/17	Wed 9/27/17	100%

Integrate Cameras	Wed 9/27/17	Wed 11/1/17	Sun 10/15/17	NA	5%
Pressure Transducer	Sun 12/10/17		Sun 10/15/17	NA	0%
Controls	Wed 8/30/17	Sun 12/10/17	NA	NA	7%
Define algorithms for processes	Wed 8/30/17	Wed 10/11/17	Sat 10/14/17	NA	7%
Develop decision trees for algorithms	Thu 8/31/17	Sun 12/10/17	Sat 10/14/17	NA	8%
Interpret data from hydrophone array	Wed 9/6/17	Wed 10/4/17	NA	NA	1%



Acknowledgements

Project Advisor: Dr Anthony Maciejewski

Graduate Student Advisors: Chris Robbiano, Megan Emmons

Industry Advisor(s): Dr. Jacob Sauer, Torie Hadel and Ball Aerospace