University of Riverside California Autonomous Underwater Vehicle Design and Implementation of SeaGoat

Aaron Chan, Adam Anunciation, Andrew Olguin, Brian Du, Ignancy Chorazewicz, James Luong, Jeremiah Alias, Jorge Gonzales, Juson Hoo, Kevin Tang, Marco Rubio(Leader), Merie Cha Tan, Raymond Lo, David Foster, Steven Herzberg

Abstract- **The SeaGoat is an autonomous underwater vehicle (AUV) developed by a team of undergraduate students at the University of California Riverside (UCR). This will be the first underwater AUV developed at UCR for the AUVSI RoboSub competition. The vessel is comprised of a custom built aluminum frame that supports 8 motors and houses a modified waterproof pelican case. The motors are configured to provide 6 axis of freedom and the pelican case encloses all water sensitive components.**

I. INTRODUCTION

The UCR RoboSub team SeaGoat is competing in the RoboSub 2016 Competition hosted by the Association for Unmanned Vehicle Systems International (AUVSI). The competition is held in San Diego California at the TRANSDEC pool from July 25 thru 31, 2016. As the first team from UCR to compete in RoboSub, SeaGoat's primary goal is to establish an organization and framework for future teams to utilize in future competitions.

The team is broken down into three sub teams electrical, software, mechanical. Constant meetings were still require

II. DESIGN OVERVIEW

The submarine uses a powerful intel NUC computer for autonomy, an STM32 for low level control and sensor input, has eight blue Robotics Motors for thrust, a Pelican Case to enclose sensitive electrical components, and an aluminum frame to hold it all together. Low cost PVC camera enclosures were inspired by team Nautilus who competed in RoboSub 2015.

The vehicle's dry weight is approximately 40 pounds and measures 33.5 inches in length, 21 inches wide, and 18.5 inches tall. The submarine has 6

degrees of freedom and is capable of moving at top speed of about 3 feet per second. Maximum Tested depth before competition is 12 feet.
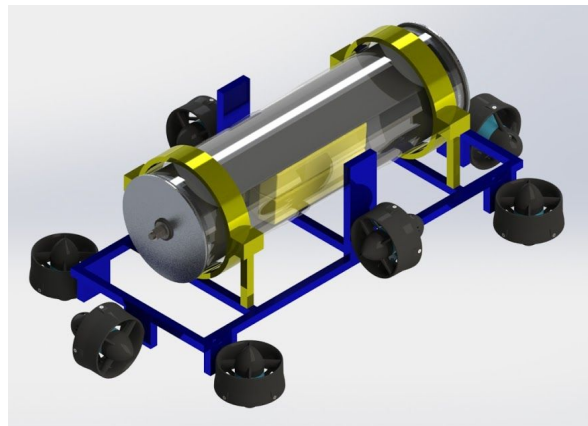
III. MECHANICAL SYSTEMS

Fig 1. Original Design

The SeaGoat mechanical system is comprised of the main electronics enclosure, the sensor enclosure, thrusters, and the frame. Original design incorporated an acrylic cylindrical tube. Due to water leaks caused by a cracked endcap redesign was needed to meet time deadlines. To meet time constraints the main hull was switched to a pelican case since it has worked for several teams in the past.

The Pelican case reduces development time by allowing quick and reliable waterproofing. A secondary enclosure is used to hold the IMU away from power components and reduce the effects on noise. The frame has been designed using SolidWorks and provides reconfigurable mounting points for optimal positioning of the thrusters.
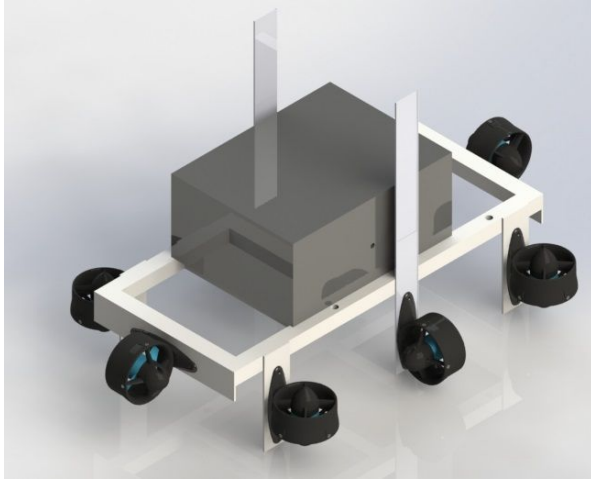
Fig 2. Redesigned Configuration with Pelican case

A. *Frame*

1 by 1 inch hollow square tubes are used to construct the frame. Aluminum was favored over steel as its lightweight and easy to work with. Fastening was achieved with steel braces instead of welding to reduce work time. Water is allowed to enter the reduce reduce buoyancy and to prevent the effect of air if a leak would occur. In addition sliding brackets have been used to create repositionable motor mounts for the forward thrusters allowing for optimal positioning. Quick adjustments allowed for simpler and faster testing. The SeaGoat's frame was designed with ease of assembly, ease of use, and cost effectiveness in mind.

B. *Motors and configuration*

Propulsion is provided by a set of 8 Blue Robotics motors, that includes, 6 T100's and 2 T200's. BlueRobotics motors were selected because of the highly favorable cost to power ratio. The Company provided useful documentation allowing for easy system integration and design.

of PVC pipe with a Lexan faceplate to provide outside vision. Blue Robotics connectors are used to connect through the backside of the housings. Generic 2.52mm headers are used to connect detachable wires inside the housing for easy detaching.

IV. ELECTRICAL SYSTEMS

A. *Power System*

Power is stored using two 14.8 volt, 4 cell, 12000

Fig 5. Claw mechanism

2) Claw: Claw is custom 3D Printed and is controlled using a servo. This was to be attached on the frame and controlled via waterproof servos.

E. *Enclosures*

1) Main Electronics and Sensor Housings: Modified Pelican cases will be used for both the main electronics enclosure and the sensor electronic enclosure. After testing it was found that the cases would take on water below 8 feet in depth. In order to improve this a layer of silicone sealant was added below O-rings. After further testing it was shown that the enclosures are watertight to a depth of 12 feet. Unfortunately we could not test any deeper as this was the maximum depth of our testing pool. In order to pass the wires through the enclosures BlueRobotics cable penetrators were used for the motors cables.
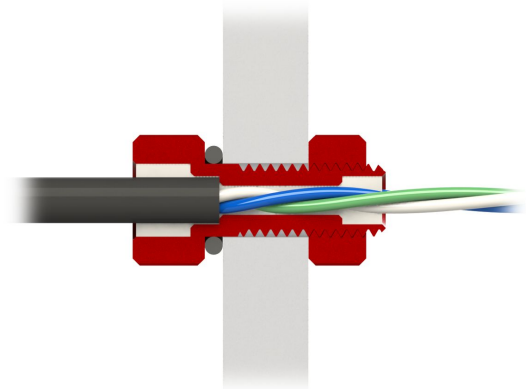

Fig 6. Blue Robotics Connectors

2) Camera Housings: Camera housing are made out

mAh, 30C VENOM Lithium Ion Batteries. They are capable of 177.6 watt hours each which gives enough time to run during the 15 minutes needed for a round. Power is distributed via the power rail using bullet connectors.
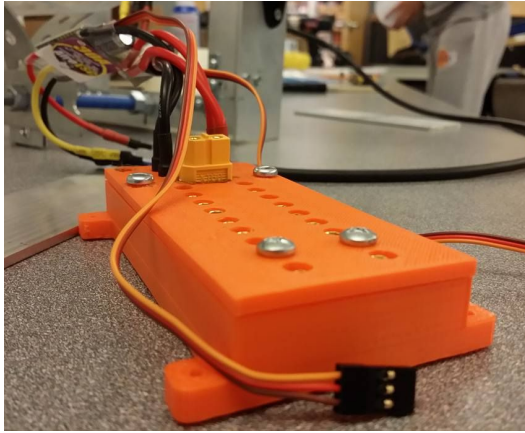

Fig 7. Power Rail

B. Actuator Control

All actuators are controlled by a generic servos. Position is controlled using PWM signals that dictate the servo's position. Servos are supplied 5 volts via the STM32 shield.

C. Communication

Serial communication is used to transmit information between the intel NUC and the STM32. Each packet contains 32 bytes of information including the vessels attitude (expressed as quaternions), linear acceleration, and a checksum for error correction. To communicate with the OLED display an Arduino nano is used as a display driver.
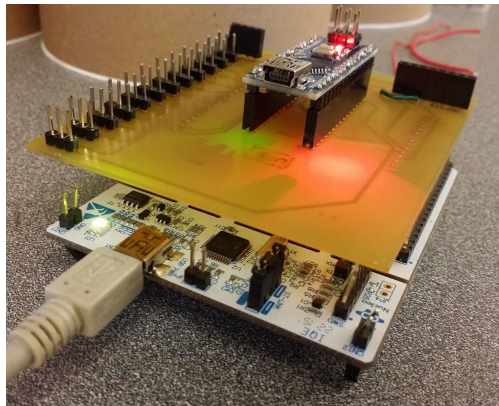

Fig 8. STM32 and control shield

I2C Protocol is used to communicate between Arduino nano and the OLED, and between the STM32, pressure sensor, IMU, and Temperature Sensor.

V. SENSORS

A. Inertial Measurement Unit (IMU)

For sensors the MPU6050 Inertial Measurement unit(IMU) breakout board is being used. It has a 3-axis gyroscope, 3-axis accelerometer, and a digital motion processor(DMP).
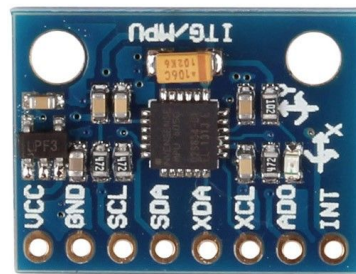

Fig 9. MPU6050 Breakout Board

B. Pressure Sensor

Blue Robotics Bar30 Pressure sensor is mounted to the case to provide high resolution pressure information.

Fig 10. Blue Robotics Bar30 Pressure sensor

The pressure sensor has 0.2 mbar of resolution equivalent to 2mm resolution. Sensor operates at 3.3v capable of accepting up to 5.5v. Comes with Standard 4 pin DF13 connector. Sensor also comes with an accurate temperature sensor ± 1 degrees. To communicate with sensors I2C protocol is used.

C Cameras

The Logitech HD Pro Webcam C910.provides the video feed used to detect objects at a distance.

Fig 11. HD Pro Webcam C910

The Logitech camera is capable of capturing video at up to 1080p resolution. Two cameras are used to gather most information needed to complete the competition. One is mounted on the front of the submarine and one on the bottom. Both are contained in their individual housing to protect them from any water.

D. Hydrophones

The hydrophone element we use is the teledyne Reson tc4013 hydrophone.

Fig 12. Teledyne. Hydrophones

The tc4013 has a useable frequency of 1Hz to 170 kHz, which capture the pingers frequency ranges. The tc4013 is connected to a non inverting amplifier using a LT6200 op amp. The LT6200 is a low noise op amp with the same noise as a 56 ohm resistor. The amplifier is then connected to a 4 stage bandpass filter, using the ADA4841-1 op amp. The ADA4841 is a low power, low noise, and low distortion op amp making very useful in hydrophone applications. Originally, we planned to use 3 or 4 hydrophone elements but we were stuck with only using one hydrophone. To utilize a single element we are having the hydrophone take a

circular path while taking samples and calculate phase shift for direction.

VI. SOFTWARE

The software system is responsible for mission planning, computer vision, guidance, navigation, and control. The mission planning and computer vision software will be running onboard the Intel NUC computer due to heavy processing power requirements of the vision software, and because of how closely the mission planning needs to be to tied to the vehicles perception system.

The STM32 microcontroller is utilized as the data acquisition, navigation, and low level control computer. The real time nature of these processes and the nature of how closely they are tied to the sensor hardware makes a MCU the ideal platform.

A. Vision

The vision system is vehicles primary means of perceiving the surrounding environment. As such the reliability of the vision software is crucial to the success of the mission. The vision software has been written entirely in C++ using the OpenCV's open source computer vision libraries.

In order to detect the mission tasks each obstacle requires its own unique set of algorithms. The object detection techniques are as follows:

1) Path marker: First, color based image thresholding is used to segment out potential orange colors using the Lab color space representation of the image. Simultaneously, the gray scale image is sharpened and edges are detected. Next, the contours are extracted from each binary image. If the area of a contour overlaps within both the color segmented image and the edge image then we say we have a potential detection. Next, the edge contours are filtered out if they do not meet the thresholds for rectangularity and area.

2) Buoys: To detect the buoys the same color segmentation method is used as described previously with the exception that three different colors are being searched for. Next, a Hough Circle transformation is applied to the edge contours within the gray scale image to find potential circles. Once again the overlaps between the contours within the color segmentation and the edge image are found.

3) Navigation gates: The same algorithms are used from the ground marker detector with the exception that the minimum contour area is smaller and two orange posts are expected to be seen. Note: In the case of a reflection on the water's surface we will take the lower of the two posts.

Once the objects detected and the contours have been extracted the center points are found and a fitted line is found. The line provides the orientation of the contour relative to the camera. This data is then sent to the low level controller to be used as feedback information.

B. Mission Planning

The mission planner is implemented as a finite state machine. Each state performs a simple high level behavior e.g. searching, centering, lock heading. By utilizing inheritance and by creating abstraction layers for the vision code each state is composed of simple high level function calls.

C. Control

After the mission had been analyzed two main mission phases were identified. The phases are the transit phase and the task phase. The two mission phases are defined as follows:

1) The Transit phase: occurs when the vehicle is moving between tasks. It begins once the system has completed task phase and ends when the vision system detect and begins tracking the mission task. During the transit phase the craft will simply maintain a constant heading, depth, and forward velocity.

2) The task phase: occurs when the vehicle is attempting a mission task. Hence, the mission task object needs to have been detected and tracked. Within the task phase a mission objective will be completed and the vehicle will position itself such that it is in the best position to enter the transit phase. For example, when a path marker is found the vehicle will position itself over the marker, match the direction that the marker is pointing, and move to the approximate depth that the next task marker is located then enter into the transit phase.

By analyzing the mission phases assumptions can be made in order to simplify the design of the control architectures.

1) The transit phase controller: designed to maintain a constant heading, depth, and linear velocity along the surge axis. Because of the inherent stability of the system, controllers for both pitch and roll angles are unnecessary. Basic closed loop PID feedback control is used to for depth as well as heading. Where feedback is provided by the AHRS and depth is found using the pressure sensor. Due to the limitations of the onboard sensors linear velocity is unobservable so a closed loop controller is not possible. Instead an open loop controller is used.
2) Task Phase controller: This control architecture is the same as the the transit controller with the addition of two position controllers for aligning the vessel with the target. These controllers use the a simple PID controller where feedback is provided by the vision system.

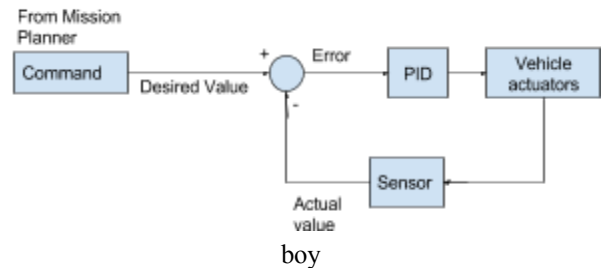All PID controllers will be tuned empirically during pool tests.

boy

Fig 13. Simple PID controller example

D. Navigation

The navigation system is responsible for providing real-time feedback of the vehicle's state. Where state is defined as the 3D position of the vehicle, the Euler/ platform angular rates, and lastly linear velocity.

However, due to the limitations of the vehicle's on-board sensors linear velocity and position, along the surge and sway axis, are unobservable when the vision system has not detected a task marker and is unable to provide visual feedback. The description of how the sensor data is utilized is provided below:
1) Attitude: In order to collect attitude information a simple AHRS has been created by using a Magdwick filter to combine IMU data together to find the euler angle of the sensor frame relative to the earth frame.
2) Depth: Within the vehicle's operating depths the Sauders and Fofnoff (1976) relationship between pressure and depth is linear. Therefore, only sensor bias and offset must be eliminated to find the depth of the vessel.

Currently, there is experimentation being done with using flow sensors to provide linear surge velocity feedback.

VII. STATUS AND TESTING

A. Waterproofing

To determine where water leaks were coming from paper towels were using as lining to visually inspect leak location.
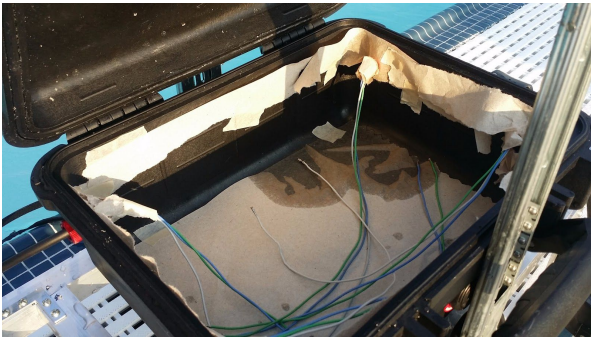

Fig 14. Pelican Case Water Test

At the time of this writing a full system test was yet to be achieved.

B. Vision Testing

The computer vision algorithms have been tested on a series of datasets that includes images taken during previous RoboSub competitions. This dataset has been generously donated to our team by SDSU Mechatronics club. The results generated by that dataset are sufficient to satisfy the necessary required for our vehicle to successfully navigate the course.
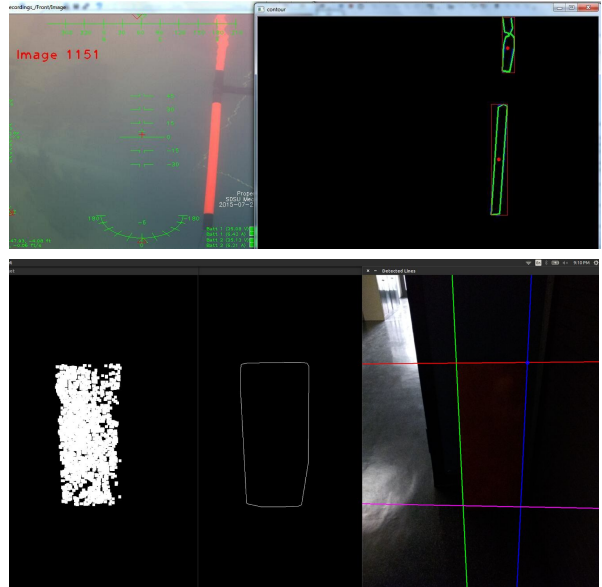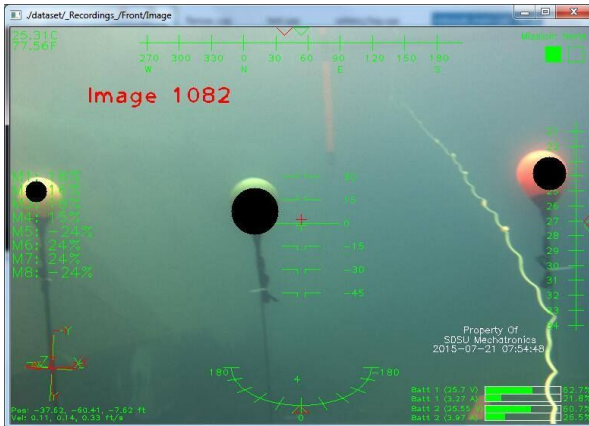
Fig 15. Computer Vision results for navigation gate, buoys, and simulated path marker

VIII. Acknowledgement