Palouse RoboSub: Development of the Cobalt AUV

Brandon J. Kallaher, Brian T. Moore, Christian Ziruk Medina Pina, James M. Irwin, Matthew S. Godmere, Ryan M. Summers

***Abstract*—This paper aims to convey the design of the 2015-2016 Palouse RoboSub AUV, Cobalt. Included is the theory behind the design process and the implementation from electrical, software, and mechanical engineering perspectives. The electrical aspect of the paper goes over the increased power requirements, how those requirements were satisfied, and the new sensor suite. The mechanical section covers the design decisions made to accommodate the need of a stable vehicle. In terms of software, the paper goes over the new features and tools created to help increase development speed, as well as the control systems and sensor filtering put in place.**

***Keywords*—journal, LaTeX, paper, RoboSub, Palouse, WSU, UI, AUVSI.**

I. Introduction

The Palouse Robosub team consists of 25 active members ranging from freshman to graduate students across five disciplines from two universities. The team spent this year redesigning the vehicle mechanically and electrically from the ground up. The main goal of this year's design was to be extensible and to maintain passive vehicle stability. To accomplish this, a well designed submarine and a robust control system were developed. Substantial modifications were done to the existing software package to create a control system that was comprehensive enough for the submarine's needs. Others improvements include expanding the Sensor Suite, and constructing a Physically realistic Simulator that can be used for module testing, and mission script validation. The outcome of this process is Cobalt, Palouse RoboSub's 2015-2016 AUV.

II. Electrical

A. Overview

The electrical team focused on a number of key goals this year including the replacement of thrusters with newer models and different specifications, the proper management of power distribution, and the creation of robust embedded systems for Cobalt. The electrical team focused primarily on a number of issues relating to last years design, specifically revolving around the integration of new systems to support the new BlueRobotics thrusters, the addition of firmware to allow for remote code updates, the creation of complex sensor reading firmware, and the management of new increased power demands.

B. PCBs

Cobalt uses completely custom PCBs that were designed by team members using KiCad, an open-sourced PCB design tool. All boards include a PIC32MX250F128B as a microcontroller to interface with the computer and all the required peripherals. These PCBs are responsible for controlling every part of Cobalt, including thrusters, sensor communication, and visual feedback. The boards were sent out for printing by OSHPark and were populated with components by team members. Communication from these boards is done with the help of a USB-UART bridge to the Intel NUC computer powering Cobalt. All microcontroller UART communications follow a robust packetized protocol that is lightweight enough to provide minimal packet overhead, but also allows for framing of arbitrary data. This format has been used extensively in previous years and has suited the submarine well in maintaining data framing.

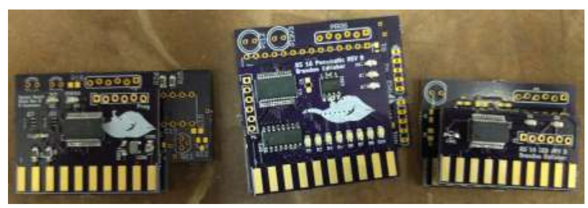

Fig. 1: Populated PCBs

C. Thrusters

This year the submarine transitioned to BlueRobotics T200 thrusters from our previous Seabotix thrusters. The T200 thrusters have nearly twice the maximum thrust at 4.5kg up from 2.4kg. Additionally, the AC BlueRobotics thrusters's Electronic Speed Controller (ESC), which uses battery power to drive the thrusters, is embedded in the thruster itself. This is in contrast to the DC Seabotix thrusters, which require large H-bridges inside the submarine hull. This feature significantly reduces heat generated within the submarine, and allows us to avoid running any thruster power through the hull. Finally, they are significantly cheaper, at as little as half the cost, and require less maintenance. This has permitted us to upgrade to an 8 thruster configuration for a cost well within our budget.

There are trade offs involved with these new thrusters, however. While potentially twice as powerful, the BlueRobotics thrusters are half as efficient and operate at a reduced nominal voltage of 16V, down from 19 Volts. Adding in the additional two thrusters in our configuration, this change increases our potential surge power draw on the submarine from approximately 800 Watts to over 2700 Watts, and our surge current from 40 amps to over 200 amps. As a result, our power system needed to be drastically overhauled, reducing the battery voltage and significantly increasing wire thickness. Currently, our battery system is incapable of delivering this surge power for more than a few seconds, and current draw is deliberately limited in software to prevent dangerous conditions. Future work will include constructing a more roboust power distribution system to enable utilizing the full capability of these new thusters.

D. Thruster Controller

The new thrusters are controlled completely differently as well, as they now take all communications over an I2C bus protocol. In the past, the computer was able to directly control the thruster H-bridges through a serial port, but that is no longer possible. To create a communication bridge between the computer and the thrusters, a relatively simple packaging scheme with a microcontroller to act as the I2c-serial translator was developed. Additionally, the microcontroller can be configured to automatically repeat commands to the thrusters to prevent them from automatically shutting down. Two bytes determine the message to command the thruster to a specific speed and the microcontroller interprets and forwards on data. Additionally, the microcontroller acts as an excellent source of timing for automatic data acquisition from each of the thrusters. With the new thrusters, data such as the running RPMs, current draw, and voltage across the motor can be queried. This information is vital for integration into the control system to allow for correction of magnetometer distortions.

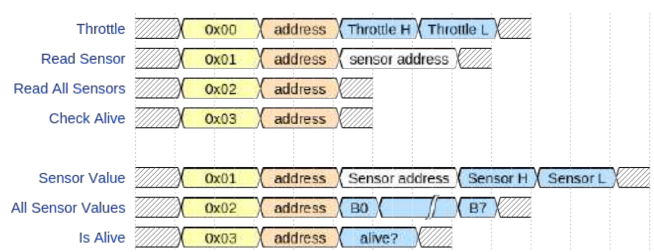

Fig. 2: Thruster Command Packets

E. Battery Monitoring

Monitoring of battery health is also a very critical part of this years' design. Because of our recent change in thrusters, current draw is a much larger problem than it has been previously. Now the cell voltages of the LiPo battery must be carefully monitored to ensure that the battery is being drained and the load spread appropriately. To accomplish this, a small microcontroller has been placed within the power management compartment to monitor individual cell voltages. This microcontroller is vital for safety shutdown in the event of dangerous battery cell balance conditions.

F. Sensor Data Acquisition

This year, an emphasis was put on redundant sensor information to ensure that control system parameters were as accurate as possible. To accomplish this, four separate 10 degree of freedom inertial measurement unit (IMU) packages were implemented at different points within the submarine. These IMUs were placed in a configuration to minimize potential noise from the thrusters and were placed in such a way that distortions could be easily calculated out. Additionally, four high-accuracy depth sensors were implemented to help validate the derived pitch and roll. The depth sensors implemented on the submarine have a resolution of 2mm and precision of roughly 1cm. By placing a number of these sensors at various points across the submarine, a three dimensional tilt plane can be created with relatively high accuracy. This information can be used to double check the derived pitch and roll control system parameters from the IMUs.

All sensors are read through a single microcontroller, which poses some strain on data acquisition timing. Currently, the IMUs must be sampled at a rate close to 100Hz for use in the control system. However, due to a fault in the microcontroller silicon that was only discovered after board bring-up, all IMUs and sensors had to be routed through a single I2C bus. This causes a large strain on the bus because all of the IMUs have identical I2C addresses and are put through an I2C mux chip to help define which device is the intended recipient. Because of this, commands must be first sent to the mux chips to select which device and then communications may be conducted between the microcontroller and IMU. With all of this communication overhead, the I2C line reaches close to the maximum transfer rate of each sensor at roughly 115Hz. Additionally, there are issues relating to sensors holding the bus lines active, which blocks all communication. The microcontroller has been automatically configured to recognize this and will cycle the I2C line until it is cleared. However, this action additionally reduces the transfer rate of the devices. In future years, it would be extremely beneficial to spread the devices across multiple microcontrollers or use an FPGA.This allows data to be efficiently read from all IMUs.

G. Microcontroller Bootloader

The bootloader was designed as a solution for all of our microcontroller reprogramming needs. Many of the microcontrollers are placed deep within the electronics bay and are hard to work on once the submarine has been completely assembled. The bootloader was designed as an application that would reside permanently in the memory of each microcontroller and run on power-up. If desired, a binary image can be uploaded to the microcontroller via a serial communication protocol and the bootloader will rewrite the microcontrollers flash while powered. This allows us to deploy remote code updates to all microcontrollers without physically opening up the submarine, which helps to remove issues relating to wear on o-ring seals and problems that can arise when enclosures are not properly resealed. Additionally, because the submarine no longer has to be unsealed to be worked on, the chance of accidentally disconnecting wires is drastically reduced. With the use of a bootloader, the number of times that the hull needs to be opened up is reduced nearly to zero.

H. Power Management

Because of the increased power requirements imposed by the new thrusters, the team decided to move main power distribution from the main hull to a ”Power” compartment. The power compartment is a separate, water-tight enclosure where all high-power distribution is managed. This design ultimately increased complexity from a mechanical engineering perspective, as they now needed to create, test, and verify another water-tight enclosure, but it was deemed necessary as a result of the large currents required by the T200 thrusters. With the addition of a power distribution compartment, all high current routing can be isolated away from the rest of the submarine. To accomplish this, the power compartment routes all thruster power out to thrusters from this case and then forwards on additional electronics power to the rest of the submarine. Power relays are also contained in this case. A high-current relay switches power to the thrusters, while a smaller relay controls power to the rest of the electronics. A PCB housed within the power case implements our power control logic in hardware, but information also needed to be gathered from some of the analog hardware to report on battery health and current consumption to know how much longer the submarine would be able to run for. For this reason, a microcontroller was placed within the power compartment to report information on the current draw of the thrusters as well as the voltages on all of the cells of the LiPo battery. Because of this implementation, more communication wires needed to be routed between the power compartment and the main hull. Wiring between these two compartments is at a premium because each water-proof connection is a potential fault point in the submarine.

III. Software

A. Overview

The focus for the software team this year was to build upon the core run-time developed last year, enhance subsystem robustness, and improve testing tools and general quality of life. The biggest issue faced last year at competition was the lack of a capable movement and sensor stack to allow the vision and AI subsystems to perform their functions. Although core run-time and debugging features were sound, the inability to obtain confident sensor readings, thruster balancing issues, and inefficient testing methods made it difficult use the platform's full capability.

The plan this year was to attack this problem at all levels of abstraction. First, a new sensor suite was created along with strategic filtering methods in order to have reliable sensor data for the software system. Second, a powerful control system stack was developed to abstract movement management away from the AI. This not only made it simple to create and test missions, but also allowed missions to be independent of the submarine, allowing us to compensate mechanical changes more quickly and easily. Lastly, a simulation platform was developed using the Unity Engine for verification testing to ensure functionality before time sensitive testing within a real pool.

B. Sensor Suite

Cobalt has undergone a significant upgrade in terms of sensor quantity and processing methods over previous Palouse submarines. In previous years, a single depth sensor, accelerometer, gyroscope, and magnetometer served to enable submarines to hold a desired depth and orientation. Functioning cameras existed, but were not integrated with the localization modules. While much could be done with this simple system, it was vulnerable to significant magnetic distortions and accelerations which would interfere with maintaining headings, and possessed no methods by which to estimate X and Y locations within the pool. Cobalt's new array of sensors and algorithms serve to address these issues.

Cobalt possesses four depth sensors, four IMUs each consisting of a 3-axis gyroscope, accelerometer, and magnetometer, four hydrophones, two front-facing stereoscopic cameras, one bottom-facing camera, and current, voltage, and rpm sensors embedded in each of the 8 thrusters. All sensor data is integrated using an Unscented Kalman Filter (UKF), with a particle filter acting as an intermediary for the hydrophone and camera data.

1) Gyroscopes: The four gyroscopes provide accurate roll, pitch, and yaw velocity measurements. Offset from drift is calibrated away on start up. The fact that rotational velocity is constant regardless of position on a rigid body is exploited for aligning the gyroscopes. Each raw data feed is rotated to common axes to remove orientation error, and is scaled to degrees/sec, with a resolution of 1/14.35 degree/sec. These measurements directly inform the UKF for rotational velocity, and are used to help correct accelerometer data.

2) Depth Sensors: The four depth sensors are mounted on the four corners of the submarine, on two different z planes. In addition to providing accurate depth readings with a resolution of 2mm, the differential depth readings provide pitch and roll measurements. These measurements directly inform the UKF to provide an accurate depth, pitch, and roll measurement, and are used to help correct the accelerometer.

3) Embedded Thruster Sensors: The thrusters each are equipped with several internal sensors which include voltage, current draw, and rpm sensors. The voltage and current sensors are used to help track battery life, and all three are used to help correct magnetometer data.

4) Accelerometers: The four accelerometers collect the sum of translational, rotational, centripetal, and body (gravitational) accelerations along each axis. The static gravity vector is used to align the separate accelerometers onto the same plane. In typical applications, the accelerometer is used to estimate the gravity vector in conjunction with a magnetometer for orientation. The four depth sensors instead provide this information, allowing for accelerometer data to be corrected and used for dead reckoning. By subtracting away an estimate of the gravity vector, as well as the centripetal force based on the rotational velocity as measured by the gyroscopes, the rotational and translational terms remain, and can separated based on a linear least-squares solution between the four accelerometers. These rotational and translational accelerations are fed directly into the UKF as the process update, removing the need for a finely tuned physical model of the system.

5) Magnetometers: The four magnetometers require the most correction to raw data to acquire useful information. The goal of these calibrations is to isolate the magnitude and direction of the Earth's magnetic field by removing the many local magnetic influences of comparable magnitudes.

Hard iron, Soft iron, and inter-sensor quadrature error is calibrated away by performing an exhaustive pin test, forming an ellipsoid of raw data points. Hard iron effects are removed by identifying the x,y,z offset and centering the ellipsoid to the origin. Principle component analysis provides the eigenbasis and eigenvalues to properly rotate and scale the ellipsoid into a sphere, removing the soft iron and quadrature errors.

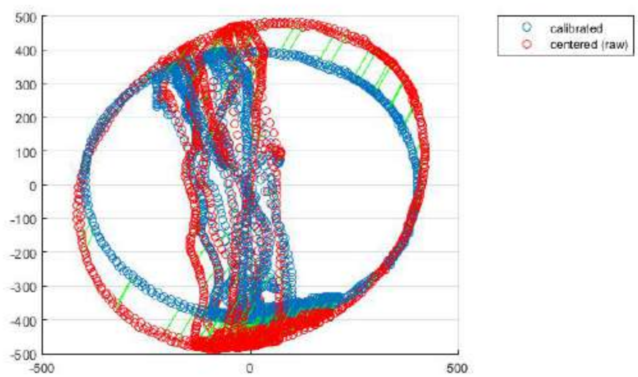

Fig. 3: Magnetometer data from Spin test.
Red points represents raw data with the Hard Iron offset removed. Blue points represent the data corrected for Soft Iron and Quadrature error correction.

Finally, dynamic, oscillating magnetic fields generated from thrusters must be adjusted for. The frequency of the oscillation is dependent on the rpm of thrusters, and the average magnetic field strength is closely correlated with the current. A total of 96 fourth-order polynomial equations, one for each axes of each magnetometer paired with each individual thruster, are used to properly correct the average static offset caused by each thruster utilizing the measurement from their embedded current sensors. While targeted IIR notch filters have been experimented with successfully to remove oscillations, a simple FIR Low pass filter is sufficient, and provides a consistent time delay which is more preferable. Together, these calibrations serve well to isolate and estimate the Earths magnetic field.

6) Smoothing and time synchronization: After calibrating using individual measurements, all calibrated sensor streams are sent through boxcar filters to remove jitter and mitigate false measurements. For the accelerometers, magnetometers, and gyroscopes on the IMUs, reporting at 100Hz, a 19th order FIR filter is utilized for an approximate delay of 100ms. The depth sensors sampled at 40Hz are run through a 9th order FIR

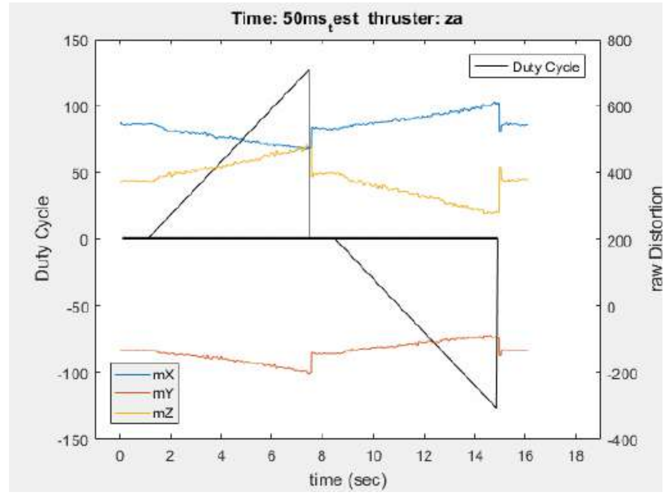

Fig. 4: Magnetometer Distortion Samples
3-axis magnetometer measurements from fixed submarine during thruster ramp test.

filter for a similar delay. From these filtered values, the derived measurements are sent to the UKF. This includes the roll and pitch estimations provided by the depth sensor streams, the yaw estimation provided by the average of the four magnetometer streams, the average of the rotational and translation acceleration estimations derived from the four accelerometer streams, and the average of the gyroscope measurements of rotational velocity.

7) Hydrophones: As cost constraints prevent us from purchasing a DVL or similar tool, our submarine must acquire measurements to estimate x and y locations by comparing the detection of landmarks against known positions. Towards this end, our hydrophone system is constructed to fully localize the relative location of the pinger in 3D space.

Cobalts hydrophone suite consists of four hydrophones. Due to cost constraints, these hydrophones are sufficient for our needs, but physically too large to construct a phase array for the 20KHz-50KHz range. Instead three time-delay arrays are constructed using one central reference hydrophone and three placed along orthogonal axes. Each hydrophone signal is conditions with an analog 8th order bandpass filter, which strongly rejects all frequencies outside of the 20KHz-50KHz range, before being sent to a 10bit 1MHz ADC.

To acquire the desired sub-wavelength precision (+/-12us) with a time-delay array requires significant considerations. The apertures of the hydrophone pairs are made as large as possible, and the signal is grossly over-sampled beyond Nyquist at a high bit resolution. Additionally, assuming the ping is a pure tonal signal with a roughly rectangular envelope, most of the information lies in the rising and falling edges. Thus, sampling the entirety of the 200ms pulse is desirable.

The size of this pulse requires either a raw transfer speed of over 8MB/s, or a buffer size of over 2MB, neither of which are common for affordable DAQs. Towards this end, the conditioned digitized signals are sent to a Digilent Zybo board for processing. The Zybo is a board with an ARM

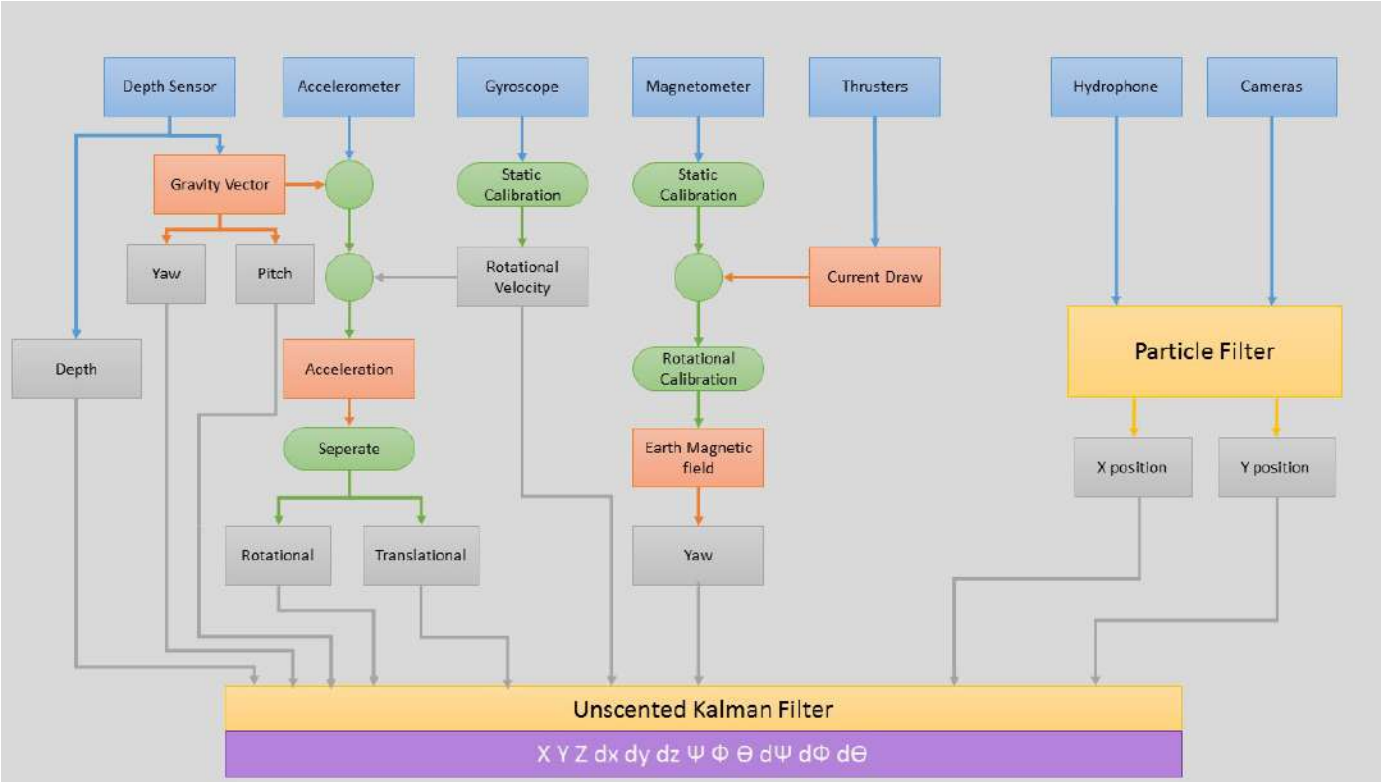

Fig. 5: Diagram of Sensor Processing
Blue = Sensor, Green = Calibration / Correction, Orange = Derived information,
Gray = State variable measurement, Yellow = Filter, Purple = State variables

processor wrapped in a large FPGA fabric with a high-speed bus connecting the fabric and DRAM. This enables us to do complex parallel processing on the signal and have an effective transfer speed more than sufficient to log the data.

Each incoming hydrophone channel is continually placed into a circular buffer of 2000 elements which for approximately 20ms of data. The data in each of these four buffers is processed with a high order FIR filter that looks for signal activity in a specified frequency band. If signal beyond a minimum threshold is detected, this indicates a ping. Once a ping is detected on any of the four hydrophones, the entirety of each buffer is dumped into a file, and new entries to the buffer are continually streamed to the file for 210ms to ensure the sampling of the entire ping across all four channels.

Once the ping has been recorded on all four channels, a cross-correlation algorithm is utilized to detect the time delay between the reference hydrophone and the three other hydrophones. Geometrically, each measured time delay between each hydrophone-reference pair defines a one-sided hyperboloid along which the pinger may lie. Finding the intersection of the three hyperboloids provides a good estimate of the pingers relative location.

As one would guess from a system of trilateration compared with the more accurate triangulation, the directional measurement towards the pinger will be highly accurate, while the distance measurement will be significantly less so beyond of the near-field. However this still provides the submarine with a useful, consistent measurement that can help bound x,y drift in addition to providing a heading with which to accomplish pinger-related tasks.

8) Stereoscopic cameras: The details of the cameras and their stereoscopic set-up are covered in the Software section. It is worth noting here, that like the hydrophones, the front cameras can accurately estimate the relative position of objects of known locations. While this information is typically used by the AI to directly control the position of the submarine, the information is also sent to the Localization module to help inform the x and y estimates.

9) Particle filter: The hydrophone measurements and front camera measurements are both integrated into the system by way of a particle filter. This avoids the necessity of running multiple asynchronous Kalman filters to resolve sensor data coming in at significantly different rates (100hz vs $<$ 5hz), and removes the inaccuracy involved with translating polar coordinate variances into Cartesian coordinate variances and vice-versa. Using 3-dimensional particles consisting of x,y,z positions, relative measurements of known landmarks are reversed to provide likelihood estimations of particles given their proximity to these landmarks. Particle positions can freely be converted to polar coordinates centered at arbitrary locations,

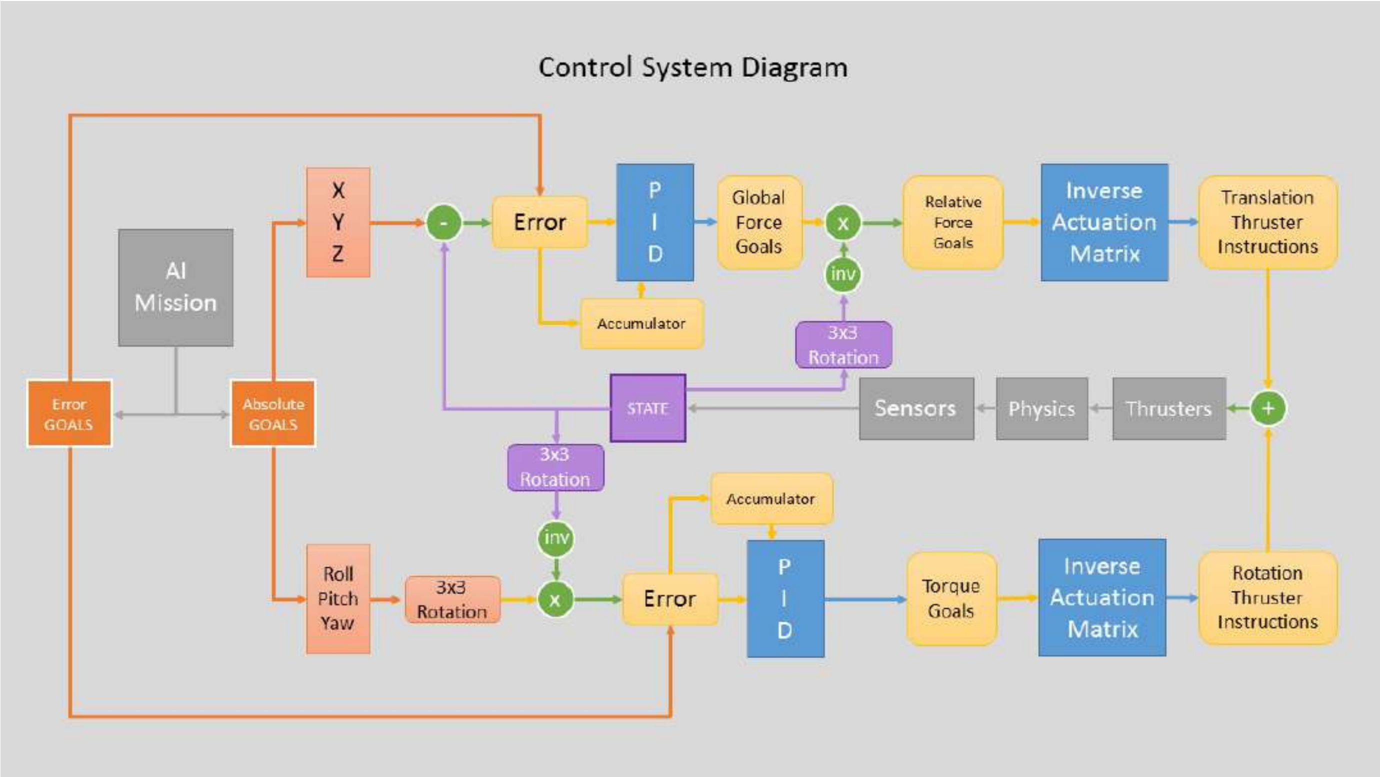

Fig. 6: Diagram of Control System
Gray = external influence, Orange = Goals, Purple = State information, Yellow = calculated values, Blue = Control module functions, Green = mathematical operations.

and have the probability of their positions with respect to angle and radial measurements considered separately and without linearization, before being converted back to Cartesian coordinates. The particle filter is re-sampled and updated whenever a hydrophone or camera measurement is made, using the results of the UKF for modeling the kinematics experienced between measurements. At the conclusion of this update, the most likely cluster of particles is identified, and its mean and variance are calculated and delivered to the UKF as a noisy measurement. The number of particles used is a dynamic function of how strained the computer is, though the number remains quite high as the particle filter will tend to update at rate under 5 hz in line with the camera fps.

In the future, this particle filter can be expanded with a SLAM stage in the update loop to help automatically map landmark locations, and to automatically resolve ambiguities such as identifying an active pinger from a list of possible nominal positions.

C. *Control System*

The control system is a modified PID-based 6-axis positional control system with special consideration for rotational slerp, windup, and hysteresis, and feedback substitution. The goal is to allow for independent, reliable control for translational and rotational goals. The system is fully parameterized, and only requires a new settings file, and PID parameter tuning to operate any on submarine with any arrangement of thrusters.

1) Translational Goal: Desired X, Y, and Z positional goals are sent to the control module. The control module compares these goal values with the current X, Y, and Z states provided by the Sensor Module. An error is generated from their difference. This error represents a scaled desired force to be exerted to push the submarine and correct the error.

This error is multiplied by both a corresponding Proportional coefficient, and an Integral coefficient. The Integral products are each added to a private accumulator, whose total values are then added to each respective Proportional product. The velocity states dX, dY, and dZ are multiplied by Derivative coefficients and added to the respective sums. These summations equate to the desired (X,Y,Z) force vector. Finally, the submarine's Roll, Pitch, and Yaw states are used to rotate the force vector to account for the submarines intrinsic coordinate system being misaligned with the global coordinate system.

2) Rotational Goal: Rotational torque goal calculations differ significantly from the calculation of the translational force goals, as they are not orthogonal in the Euclidean sense, and must be considered as a whole rather then independently.

The desired Roll, Pitch, Yaw orientation is delivered to the control module. The 3x3 rotation matrix from the origin to this orientation is generated. This is multiplied by the inverse of the 3x3 rotation matrix generated from the Roll, Pitch, and Yaw of the current state. The resulting product is a 3x3 matrix that described the rotation from the current orientation to the desired orientation. This relative-rotation matrix is converted back to its nominal roll, pitch, and yaw components. These values equate to the roll, pitch, and yaw errors. The pitch error is modified as a function of the roll and yaw errors to more smoothly rotate between the current and desired orientations. This is mathematically different, but functionally similar to a 'slerp' performed with quaternion rotation vectors. From here the desired torque goals about each axis are calculated in the same manner as the translational forces with Proportional, Integral, and Derivative components.

3) Actuator Conversion: Upon start-up, a 6xN Actuator Matrix is generated from each thrusters position, orientation, and maximum thrust described in a settings file, where N is the number of thrusters on the submarine. This matrix converts a vector of percent-maximum-thrust instructions into the resultant X, Y, and Z forces and Roll, Pitch, and Yaw torques that would be effected on the submarine. Since the system is over-actuated, Singular Value Decomposition is utilized to generate an optimal Inverse-Actuator Matrix, which can be multiplied by a 6x1 column vector of desired X, Y, and Z forces and Roll, Pitch, and Yaw torques to generate a combination of thruster instructions that will efficiently deliver those forces and torques.

The control system initially separates translational and rotational goals for calculation. Each cycle, after the desired (X,Y,Z) force vector is determined, it is padded with three zeros for Roll, Pitch, and Yaw, and then multiplied by the Inverse-Actuator Matrix. The resultant thruster instructions will produce the desired translational force while exerting zero net torque on the submarine. Conversely, when the desired Roll, Pitch, and Yaw Torques are determined, a set of instructions are produced that will yield those desired torques while providing zero net translational force. The thruster instructions for the translational and rotational goals are then added together and sent to the thruster module for execution by the thrusters.

4) Windup: A limitation is set on the absolute value each of the accumulators may hold. This is common practice and reduces overshoot when resolving a large error.

5) Hysteresis: A modification on the standard PID system; when the absolute value of the error of a state is below a certain threshold, the Proportional coefficient is set to zero, leaving only derivative and integral terms to effect actuation. This permits higher proportional coefficients without the inefficient or jerky responses that would result from small steady state oscillations in error about 0.

6) Error goals: Rather that receiving an absolute state goal, the Control system can receive error goals which directly set error values in the feedback loop. This ignores the current state of the submarine and converts the positional control system into an open-loop fly-by-wire system. This is a convenient, simple way by which another system, such as a human operator with a joystick, or the AI, may substitute its own preferred method of feedback. As an example: to center on an object, the AI may take the feed from a bottom-facing camera, and continually deliver the X and Y pixel positions of the object's centroid as X and Y error-goals. This allows for missions to seamlessly switch between general positional goals when moving about the course, and more finely-controlled maneuvering relative to an object when accomplishing tasks.

D. Simulator

A submarine simulator was developed this year to streamline verification testing since it was found that a lot of time was wasted during pool tests trying to get new functionality up and running. Since the feedback of the software system is so complex, a full simulator was needed instead of a simpler faked data testing method. A way to test new submarine designs before they were finished was also needed since software testing has been hindered by mechanical and electrical delays in the past, making it difficult to verify the software on a new platform.

The simulator was developed using the Unity Engine and was written in C# in Linux using the new Unity Editor port. The overall design of the simulator was somewhat trivial since it matched the real-world system nearly one to one. Since our original architecture was built upon a very modular and asynchronous model via inter-process messaging and shared memory, the simulator fit into the software pipeline seamlessly. The goal was to simulate all real-world subsystems (sensors, thrusters, and cameras) as closely as possible. The rest of the software would be exactly the same and would operate without realizing that the hardware was being simulated.

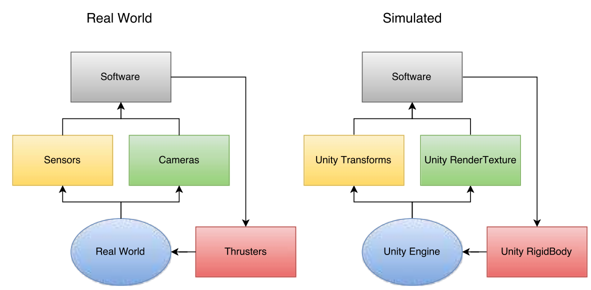

Fig. 7: Simulator Control Loop

1) Sensors: Hardware sensors were not simulated directly due to their complexity, but their outputs were matched using Unity's built in 3D transforms. Every loop, the simulated sensor module calculates the orientation of the submarine and outputs it to the software for processing. Noise can be added to more closely simulate bad sensor readings depending on observed noise within the real system.

2) Thrusters: Thrusters were also not simulated directly since fluid dynamics are too computationally expensive. However, since thrusters were linearized in the real system, it is assumed linear scaling of thruster values to the engine, making thruster simulation trivial. The thrusters are treated as a list of thrust vectors loaded from the submarine configuration file. This loads the offsets of the thrusters to the center of mass, their orientations, and their minimum and maximum thrust.

Additional properties such as ramp up time, delay, and transfer functions to simulate fluid dynamics can also be modified depending on the real-life characteristics.

3) Cameras: Cameras were also not simulated directly due to the complexities of fish-eye projection and lens distortion. Fish-eye projection is also not supported in OpenGL or Unity and must be done manually using multiple rectilinear viewports. To simplify, since cameras are calibrated, undistorted, and rectified for the actual system, the extrinsics can be loaded to place cameras in the same location and projection space as the real submarine to recover the rectified coordinate frame. The vision pipeline then becomes identical as the real one barring re-projection errors within the original calibration routine.

4) Submarine: Simulating the actual physics of the submarine was done much more directly than any other portion of the simulation. Since the Unity engine supports rigid body physics, mesh rendering, and drag, most of the work was importing these components into the engine using approximations from the original mechanical engineering designs. A visually representative rendering was done using a down-sampled mesh and texture overlays from the SolidWorks model. Statics such as the center of drag, mass, and buoyancy were also taken using SolidWorks calculations based on the estimated properties of the submarine. Dynamics were a little more work since Unity does not support tensor drag and rotations directly. Instead, a simple tensor rotational/translational drag and inertial model was made that could closely mimic the dynamics of the submarine in water without the complexities of fluid dynamics.

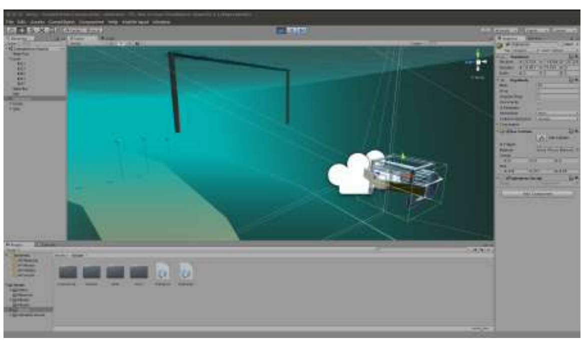

Fig. 8: Simulator Mission Testing

5) Pool: Since the submarine can be tested in many different environments, the pool simulation is just as important as the actual submarine. Pool and obstacle simulation, water reflections, fog rendering, lighting are all very complex problems to simulate properly. The competition pool was made using Google Earth measurements along with documented measurements of the geometry of the bottom. Obstacles were made from documents as well and can be placed and moved as needed once the final layout is measured at competition. All models were made using Google SketchUp and were imported to the engine using a common format. Collision meshes and other scripted elements were generated once imported to allow entity interaction and debugging. To simulate water, the built-in Unity water shaders were used along with an exponential fog layer in the camera rendering pipeline. Lighting is also built into the engine and can be adjusted depending the time of day and lighting conditions. The overall visual quality is still very far from a real-world system, but for the computational benefits and lack of impact on a color based vision pipeline, it is more than adequate for our needs.

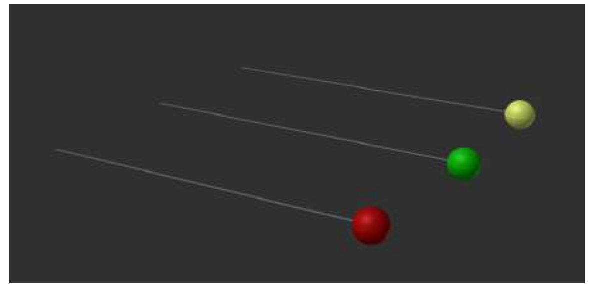

Fig. 9: Buoy Model

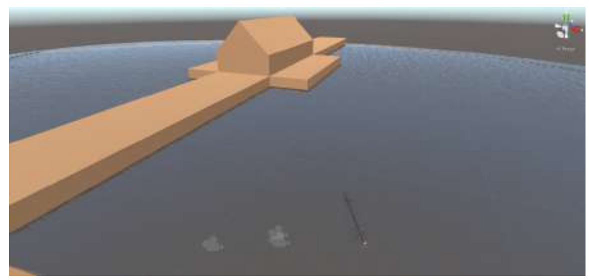

Fig. 10: Pool Model

IV. Mechanical

A. Overview

This year the vehicle was completely redesigned from the ground up, using lessons learned from past designs. This years mechanical system consists of an upper hull, frame, and external sub-enclosures that house our AUV's electronics while the frame provides structure and mounting points for the AUV's external enclosures and thrusters. The entire AUV was designed using SolidWorks for solid modeling as well FEA analysis of the AUV's frame. All Manufacturing was completed in house using Manual machinery (mill, lathe, brake, etc) or using CNC equipment to complete more complicated parts. All programming and operation of the machines was completed by team members with supervision from the shop manager. Programming of parts was completed with Computer Aided Manufacturing (CAM) software Master Cam.

B. Frame

Cobalt's frame is far different from all previous frame iterations our team has constructed in the past. Moving away from the Bosch Rexroth extruded tubing has allowed the design of the frame sub-assembly to complement the rest of the AUV. Important considerations for the design of our frame was

Fig. 11: Rendering of Cobalt

symmetry and modularity. The frame is composed primarily of two sheet metal U-channels that incorporate a hole pattern for easy placement of thrusters and actuators. The struts that tie both of the u channels together allow for sub-enclosures and the AUV's hull to be mounted bringing the entire sub together. The front of the sub includes a hinged door that latches to the frame when in operation. The doors give access to the battery and power enclosures for easy access and allows the primary hull tubes to be removed when the doors are open.

C. Bulkhead

The bulkhead is the central hub for Cobalt's wiring as well as the main hull. As such, it must be robust to support the weight of all the electronics and the hull but not too heavy that it starts to buckle sections of the frame. Different manufacturing and material choices were analyzed before reaching the final version of the hull. The choices for material were between a solid block of aluminum or aluminum tubing with larger outside and smaller inner diameter of which the design demanded. It was selected to use the aluminum tube with a nominal outer diameter of eight inches and an inner diameter of four inches. The price of the tube was approximately 130 dollars where the aluminum block needed for our size requirements was approximately 500 dollars. The choice for manufacturing was between sending it out to a local machine shop or to complete it in house. It was decided to complete the bulkhead in house to take the challenge of learning Mastercam and the requisite manufacturing processes. The wiring panels located on both the port and starboard side of the AUV can mount up to 20 blue robotics wire penetrators or a combination of SeaConwetmate connectors and BlueRobotics thrusters. Using a panel for this application allowed for future manipulation and access to the center of the bulkhead for routine maintenance. The top of the bulkhead is where the switches are located. The switch housing is 3D printed allowing modification or switch addition if necessary. Each switch has a magnetic base that activates functionality using reed switches.

D. Electronics Rack

The electronics rack was designed for modularity in mind. The hole pattern allows for easy manipulation and addition of electronics components when necessary. Due to it being cantilevered the mounting bars are 1/4 inch to increase the web area thus increasing the rigidity of the entire assembly. All electronics components are mounted to a 3D printed fixture which is then mounted to the electronics rack.

Fig. 12: Rendering of the Bulkhead

E. External Enclosures

Secondary enclosures include the camera cases, power enclosure and the battery enclosure. The camera cases this year include domes that are designed to allow the fish eye lens to have 160 degrees of view and is epoxied to the Delrin cap. Inside the camera case is a 3D printed fixture that supports the camera and lens. The support is cantilevered to the rear cap allowing the entire camera assembly to be accessed from the rear cap for maintenance and lens adjustment. The power case was designed using Delrin as the primary enclosure material with an aluminum lid that uses a face seal to keep water from breaching the inside cavity. The battery case uses a polycarbonate tube and Delrin caps to contain the battery. All seals for the battery case are bore seals allowing the battery to be easily accessed if necessary. The design allows for easy repeatability, if more enclosures needed to be made for quick battery swapping the design is kept simple.

Fig. 13: Rendering of a Camera Case

V. Conclusion

Significant strides have been made over the previous submarine iterations that make Cobalt a capable, reliable, expandable test frame moving forward.

An emphasis on parameterizable software and modular features ensure that significant portions of software including the AI, the Sensor suite and the Control system will be used independent of any hardware modifications to the submarine. They are all sophisticated enough to fit our needs for the foreseeable future, and dynamic enough to accommodate any hardware or electronic modifications with minimal adjustment. Future work will include expanding the type of sensors used, to better localize the X and Y coordinates of the submarine, and improving the magnetometer algorithms to reject local, external magnetic distortions that may be found at the Transdec pool and beyond.

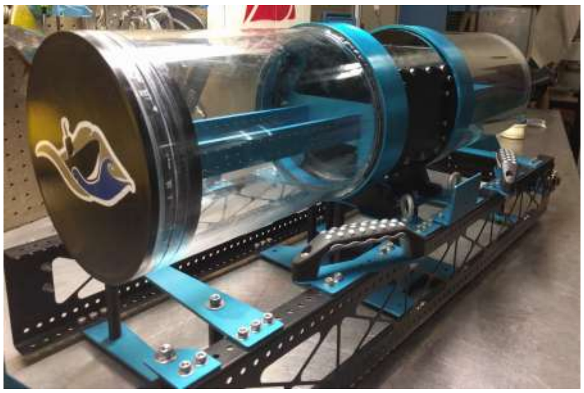

Fig. 14: Complete Mechanical Build

A spacious, open mechanical frame gives significant room for electronics hardware expansion, and a wide selection of mounting holes for attaching future actuators or external enclosures. A design emphasis on ease of manufacturing and inexpensive materials ensures that modifications to future designs can be quickly realized. The creation and testing of actuators for more technical tasks will be a high priority for the following year's submarine.

The electrical system is simple and reliable, but is the most specialized and integrated part of the submarine. Modifications are slower and more difficult with this system. Furthermore, high power requirements with low voltages demanded by our thruster system make safe and efficient power distributions more difficult. Future work will focus on improving the power distribution system and streamlining the electronic design components to make the electrical and electronics subsystems as adaptable and accommodating as the other features of the submarine.