Design and Implementation of a Pair of Integrated, Autonomous Underwater Vehicles: *Deb* and *Flo*

James Deacon, Julia Deacon, Brendan Holloway, Sophia Liu, Cormac O'Neill, Henry Sun

California Institute of Technology, 1200 E California Blvd., CA 91126

July 2, 2018

***Abstract* – The Caltech Robotics Team's pair of autonomous underwater vehicles (AUVs), *Deb* and *Flo*, are designed for precise maneuvering at the 2018 intercollegiate RoboSub competition. Designed and manufactured by 25 undergraduates at the California Institute of Technology, *Flo* is the submarine we brought to last year's competition. She is an iteration on 2016's vehicle *Dory*. Improvements include a stereoscopic vision system, a refined pressure hull design, seven new thrusters to provide extra power and reliability, a servo-driven gripper, and magnetically-actuated torpedoes and markers. With these improvements, the vehicle can maintain an arbitrary orientation and velocity, navigate the obstacle course under many lighting conditions, locate an ultrasonic pinger, fire torpedoes and markers, and grip various objects. In addition, this year we built a new sub *Deb*. She has the same 7-thruster arrangement as Flo, but she is designed in the shape of a plus. She also has detachable battery pods and a gimballed camera. *Deb* and *Flo* represent the state-of-the-art for low cost, actively stabilized autonomous underwater vehicles.**

I. INTRODUCTION

The Caltech Robotics Team's mission is to help its members develop engineering, communication, and collaboration skills in an exciting and hands-on environment. We do this by designing and building autonomous underwater vehicles for research, education, and outreach. Our vehicles, *Deb* and *Flo*, were built to complete the challenging RoboSub competition course, cosponsored by the Association for Unmanned Vehicle Systems International (AUVSI) and the Office of Naval Research. The competition takes place annually in July/August at the Space and Naval Warfare Systems Center Pacific's San Diego Transducer Evaluation Center (TRANSDEC). The competition simulates real-world underwater vehicle tasks such as manipulation, navigation, and visual inspection. Competition elements include firing torpedoes at small targets, locating sonar pingers, manipulating small objects, and dropping markers into bins. All of these tasks must be completed autonomously with no human in the loop. To complete these tasks, our subs must have accurate visual and acoustic sensors, precise navigation, control and guidance, and robust object recognition and manipulation.

II. DESIGN OVERVIEW

Deb and *Flo* are autonomous underwater vehicles designed for reliable operation and precise control. The pressure hulls consist of two custom-designed anodized aluminum shells that were machined on a five-axis CNC. An acrylic dome allows us to clearly see the electronics. Flo's electromagnets are positioned in strategically located regions of the hull to actuate external torpedo launchers without requiring openings for pneumatics lines. Powerful servos are positioned outside the hulls to actuate 3-DoF grippers. Seven thrusters give the vehicles control over all six degrees of freedom, and redundancy in certain axes. For Flo, we replaced the gimbaling camera on *Dory* with two cameras, one wide-angle and one narrow-angle, to allow us to identify objects both near and far, swapping between the two as the task progresses. Deb uses a gimballed camera again.

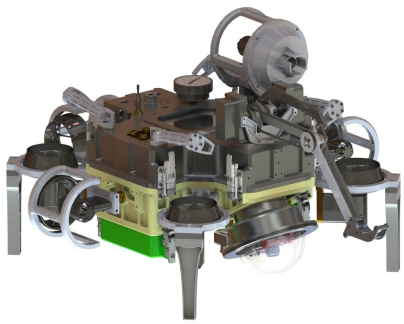

Figure 1: Computer-aided design model of *Deb*.

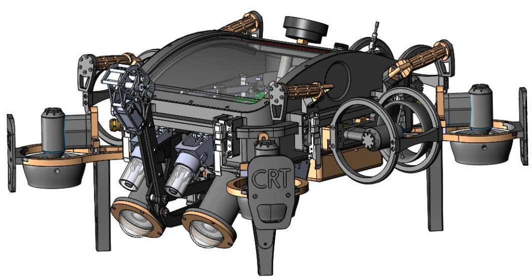

Figure 2: Computer-aided design model of *Flo*.

The electronics are mounted to a two-stage enclosure that swings open on a 4-bar linkage to ensure easy access to all components. The vehicle is powered by two lithium-polymer batteries, giving our subs a runtime of over two hours. The electronics have separate power and battery systems for the actuators and for the computer and sensors. The vehicles' software runs on a smaller Intel NUC™ computer system, which saves mass and volume. The subs use a custom inter-process communication library to implement their vision, actuation, and autonomy systems. This year we have begun using machine learning to solve some vision tasks.

III. MECHANICAL SYSTEM

Deb and Flo's mechanical systems consist of a main pressure hull, external electrical equipment, and task-interaction hardware. Some external equipment (the camera modules and battery pods) are integrated into the main pressure hull; their enclosures are sealed to prevent water ingress and mount to the main pressure hull. All parts were designed by team members using the SolidWorks computer-aided design software. Several pressure and structurally critical parts were simulated using ANSYS finite element analysis software. Parts were machined by hand and on 3-and-5[1]-axis CNC machines. All in-house machining used HSMWorks CAM software, and all 5-axis machining used MasterCAM software. By only performing necessary mechanical upkeep and maintenance on Flo, while making modifications as necessary to prepare her for the changes in this year's competition, hardware could be effectively re-used and allow for precious manpower to be diverted to developing new hardware for Deb.

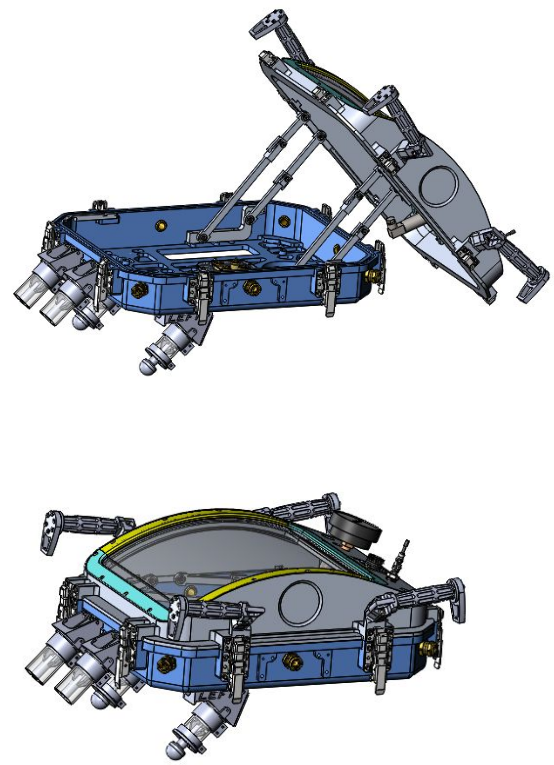

Figure 3: **Flo's main pressure hull.** The four-bar linkages (shown open at top) keep the top connected to the bottom when open. Some parts not shown to display hull structure.

[1] All 5-axis machining was contracted out to GCC (Glendale Community College), which has a professional machine shop.

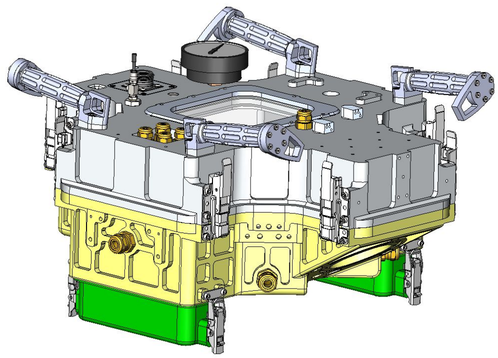

Figure 4: **Deb's main pressure hull.** The four-bar linkages is not shown.

Main pressure hulls

Deb and Flo are designed around central pressure hulls that act as the hub for all of the connected pressurized enclosures while safely protecting the vulnerable electronics of the vehicles. The decision to re-use Flo's pressure hull greatly reduced the manufacturing cost both in terms of money and manpower invested. However, re-using last year's design also limited the options available to modify Flo to successfully complete the reworked tasks for the 2018 competition. The curved acrylic dome in Flo's pressure hull, while allowing for easy viewing of the state of internal electronics of the vehicle, limited the amount of external real-estate for task hardware. The dome consists of a custom-molded sheet of acrylic that seals using a 3D O-ring seal and a series of curved clamps that ensure a good seal without sacrificing aesthetics. Although this system ensures that the acrylic can be quickly and easily replaced in the event of any damage, it essentially prevents the use of the entire upper portion of the vehicle for attaching additional components. The design of Deb's pressure hull sought to overcome this design constraint by replacing the curved dome with a flat surface. A small acrylic window replaced the dome in order to allow visibility to the key electrical components, while the flat aluminum upper hull could be used for mounting the gripper and suction mechanisms.

The use of acrylic on both vehicles also aided in the extensive weight-reduction process which all of our designs must undergo. Both Deb and Flo are extensively ribbed during the machining process, however it was found that the internal ribbing of Flo - while being aesthetically pleasing - provided pockets for oil leaking from the thrusters to accumulate over time. As such, Deb was ribbed externally. This had the added benefit of making it easier for the pressure hull to be machined, since smaller tools could be used in the process.

The halves of each pressure hull seal together using a single O-ring. The latch-based sealing mechanism used in Flo was modified for use in Deb, although new latches had to be found and tested to ensure that they would perform as desired with the new hull geometry. These new latches were found via an extensive testing program where the clamping force of a variety of latches was measured. The latches were integral to the overall design of the pressure hull and the layout of external accessories on both vehicles. This is a consequence of the user needing to be able to access every latch with ease, which proved difficult given the tight corners in Deb's hull design. By flipping the latches upside down relative to last year, we could ensure sufficient clearance at the cost of making the pressure hull more difficult to close manually. Both hulls also have their two halves connected via a 4-bar linkage that allows for the vehicles to be left open while work is being done on their internal hardware.

The interior of both pressure hulls consist of electronics mounted to acrylic plates. This system has been found in previous years to work effectively while remaining light-weight and enabling sufficient air circulation for convective cooling. These packages consist of all non-sensor electronic systems, such as processing, power distribution and electromagnetic actuators. The two-level system used in Flo was once again used in Deb, since it allowed for efficient use of the limited internal volume. The ability of the upper level to swing away from the lower level on straight linkages when the pressure hull is open also proved to allow easy access to all of the electronics when needed.

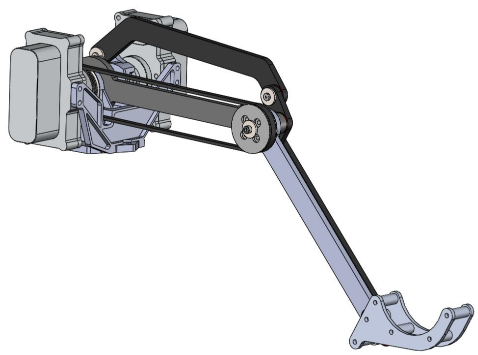

Figure 5: **Common gripper design.** The tuning belt shown transfers the rotational motion of one servo to a joint further down the gripper. The mechanism is designed to keep the angle between the end effector and the vehicle's reference frame fixed.

External Accessories

The external accessories of the vehicles include all parts whose functionality requires them to be outside of the main hull. This primarily includes the cameras, the thrusters, the DVL, the hydrophones, battery pods and task-interaction hardware (markers, torpedoes and gripper). In both Deb and Flo, the camera systems and the batteries are kept in separate modules that are attached to the main hull and form a single pressurised environment. While in Flo, the cameras were fixed in place within hemispherical acrylic domes, a gimballed system was used for Deb's vision system. This was intended to allow for the vehicle to remain stationary while it searched for tasks or objects. This would be especially useful in object manipulation tasks, which Deb was being designed to prioritise. While both vehicles use a set of hydrophones for searching for the pinger task, Deb was equipped with an additional hydrophone for transmitting data to Flo as well. This allows for one-way inter-vehicle communication while retaining the redundancy of having four receiving hydrophones, although Flo can also be made to transmit data by reducing the number of its receiving hydrophones to 3. To prevent Deb from occluding Flo from any transmissions when Flo is operating closer to the surface, an additional set of hydrophones were mounted to the top of Deb for redundancy. The use of latches for Deb's battery pods was an additional innovation over last year's design. While the battery pods have always been removable in our vehicles by unscrewing them, in Flo the process was slow and inefficient, meaning that the batteries were in practice accessed via the main hull. By attaching the battery pods by latches to the "wings" of Deb, they could be quickly and easily accessed without exposing the critical electronics of the vehicle. This is useful both from a practicality standpoint as well as from a vehicle safety stance, since it limits the risk of water getting into the pressure hull. In order to seal the battery pods, two latches and an O-ring are used on each "wing" of Deb.

In Flo, the torpedoes and markers are designed in such a way as to not require through-holes in the pressure hull. This is accomplished by using solenoids inside the vehicle to attract magnets to release the projectiles. The torpedoes themselves are propelled by a spring system, while the markers are simply dropped. Regardless, both are designed in such a way as to allow for them to be triggered with only a small force from the internal solenoid. However, while this design improves vehicle safety, it greatly increased the complexity of the design of the firing mechanism and limited the materials which it could be 3D printed out of while maintaining the required strength. Furthermore, we found that the solenoids could adversely impact the performance of Flo's sensors. As such, Deb is designed to use servos to drop its markers instead of a solenoid. However, this does mean that Deb was not designed to fire propelled torpedoes, limiting the interchangeability of the two vehicles for different tasks. The gripper design for both vehicles is quite similar, with both using the same core design driven by two servos attached to the main pressure hull. While one servo is directly attached to its rotary joint, the other manipulates a joint further along the arm through the use of a timing belt. This decision was made so that a heavy servo did not need to be placed midway along the gripper mechanism, which would have added weight and reduced its effectiveness. However, while Flo's gripper is purely designed to interact with levers and as such has a simple end-tooling design, Deb's need to interact with golf balls meant that a suction mechanism was necessary. This design uses a Videoray M5 thruster as a pump to draw water through a flexible tube attached to the end of the gribber. By creating suction at the end of this pipe, golf balls can be collected and stored in the end tooling. The balls are released by reversing the flow of the pump motor. Both Flo and Deb also re-use the same iconic 7 thruster configuration that we have used on previous vehicles. By having the same configuration on both AUVs, we are able to have a common navigation and mobility code base. The use of seven thrusters allow the vehicles to actively control all six degrees of freedom underwater.

However, Deb’s vertical thrusters do have to be flipped upside down in order to allow access to its battery pod latches. While this is not a problem once the vehicle is running, it introduces the problem of the thruster props being higher up on the vehicle. In order to ensure that the props would be in contact with the water while Deb floats on the surface of the water, special care had to be taken when considering its weight and buoyancy.

Thrusters

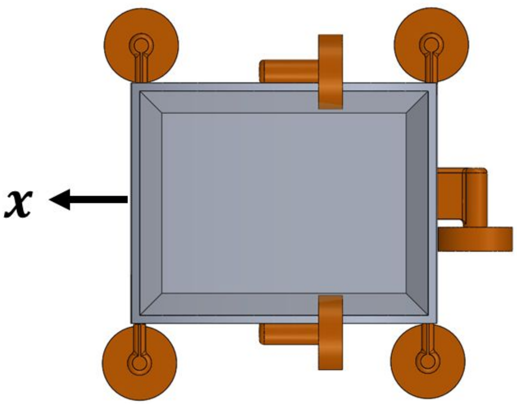

Figure 6: Thruster Configurations. The basic configuration for the vehicles. Notice that there is redundancy in the vertical thrusters, which enables the vehicle to have a “thruster-out” operating mode.

Deb and Flo have inherited Dory’s iconic thruster configuration, utilizing seven brushless VideoRay M6 brushless thrusters. Because of the linearity of newtonian forces and torques, this configuration can apply any force and torque in any direction. This allows the subs to actively control all six degrees of freedom. Although six-axis control theoretically only requires 6 thrusters, the seven-thruster configuration has excellent handling characteristics without restricting the physical layout of the vehicle (see Figure 3, bottom). Notice that the more traditional 6-thruster layout requires thrusters to be mounted on every surface of the vehicle. By contrast, our system keeps the top, bottom, and front of the vehicle available for other hardware and also adds a redundant vertical thruster.

IV. ELECTRICAL SYSTEM

Flo’s electrical system powers and provides a communication platform for all sensors, actuators, and computational elements onboard in a robust and efficient manner. Electronics are implemented modularly; each subsystem is managed by a specific PCB. This minimizes the chance of an error in one part of the system causing failure within the rest of the system. The electronics package is shown in Figure 7. For Deb's electrical system, focus was placed on consolidating most of the subsystems into one main system to remove unnecessary costs and complexity. Both Deb and Flo's electrical systems will be discussed in greater detail below.

Figure 7: Flo’s electronics package outside the pressure hull.

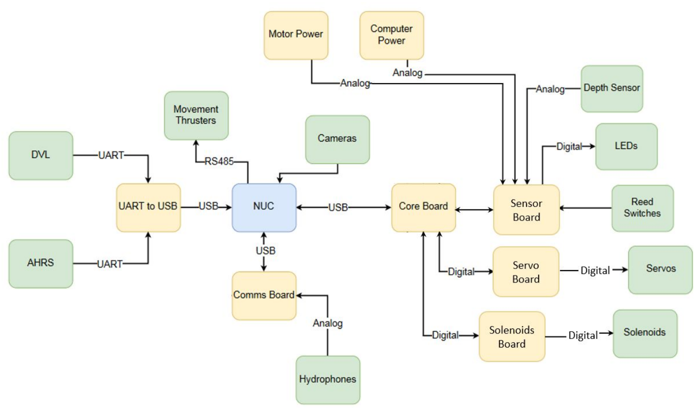

Figure 8: Data flow of Flo’s electrical system. The main computer, the NUC, communicates with the various sensors through the UART to USB and core board. The core board communicates with the sensor, servo, and solenoid boards. The power boards send kill switch and power usage data to the core board. The comms board receives data using the hydrophones.

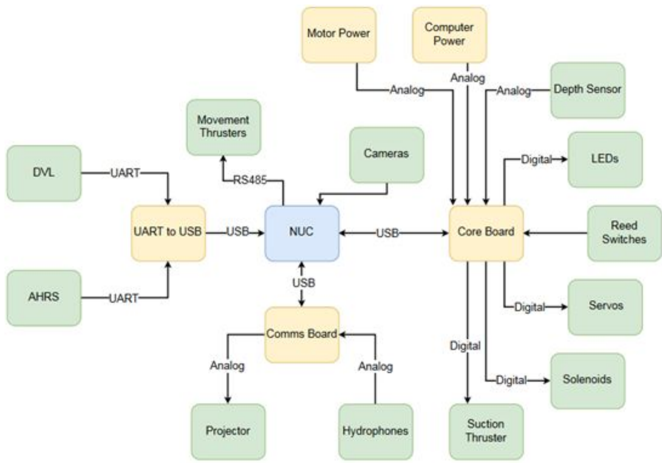

Figure 9: Data flow of Deb's electrical system. The modular boards from Flo have been replaced by one consolidated core board.

A. Mechatronics System

The thrusters and pneumatic solenoids use a separate power system from the microcontroller system to avoid noise and interference with the rest of the electronics. The motor system is responsible for powering the thrusters, solenoids, and gripper, while the microcontroller system is used to power the computer, sensors, and the sensor control and communications.

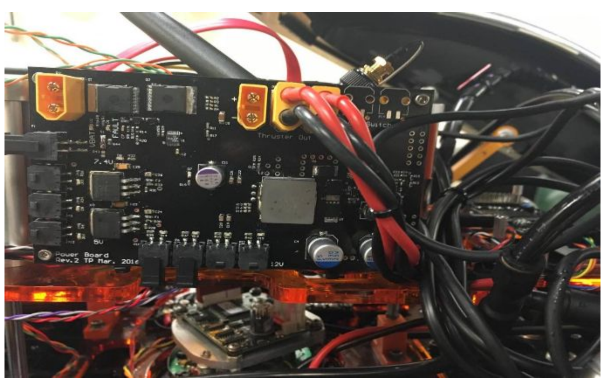

Figure 10: Microcontroller power board. The upper left contains the protection circuitry while the rest of the board contains voltage regulators.

B. Power System

Flo uses the same overall circuit topology for both the motor and computer side electronics. The circuitry consists of a hot swap controller at the input which provides over/undervoltage, reverse voltage and overcurrent protection. The hot swap uses two external power MOSFETs and sense resistors to respond to fault conditions. The controller features a capacitive timer to adjust the timing that triggers a fault condition, and series sense resistors to set the overcurrent fault. The motor power and computer power boards have two different overcurrent set points. The motor power has a current limit of 40A, largely dominated by the 36A thruster current at full throttle, and the computer power board has a current limit of 20A.

During an over/undervoltage fault condition, the hot swap controller will cut power to the system by driving the gates of the external MOSFETs low. An overcurrent fault is triggered if the system draws more current than the current limit for a period of time that is greater than the timer timeout. This type of fault requires a power system reset in order to clear. The hot swap controller has an active low shutdown pin that is used to cut power to the electronics without removing the batteries. The board has a solder jumper that configures the behavior of the kill switch. The computer power board remains on until its kill switch is closed, while the motor power board will remain in shutdown until its kill switch is closed.

Additional protection and monitoring is included for the motor power board, as the motor system uses significantly more power than the microcontroller system. Both power systems contain 50A fuses at the main battery input. Each thruster output also includes a 10A fuse. Deb also includes additional monitoring of the temperature near the main power MOSFETS, the current drawn from the battery, and the current drawn from each thruster output, all of which is serialized and sent to the computer system.

The power boards have 12V, 7.4V and 5V outputs. The 12V and 5V power rails are produced using efficient switching buck regulators to minimize power loss. The 12V power rail can handle a 15A output current, while the 5V rail can handle a 3A output current. On the motor side, the 7.4V power rail is used to power the gripper, and can handle a 15A output current. Both power systems are powered by a 5000 milliamp-hour, 25.9V lithium-polymer battery.

C. Microcontroller system

The microcontroller system provides low-level communication to sensors and actuators that cannot be controlled directly by the main computer. Deb's microcontroller system consists of five boards: the core board, the sensor board, a solenoids board, a servo board, and the hydrophone board. Each board is equipped with a STM32F407VG microcontroller as the STM32F407 provides many integrated peripherals. These boards communicate over an

RS485 bus network in order to efficiently pass messages between subsystems.

The core board (Figure 11) is responsible for handling communication between the Intel NUC™ computer and the rest of the microcontroller system. Messages between the computer and the core board consist of a message ID and five bytes of data. Computer to core messages are parsed by the core board and handled by sending the data to another microcontroller board as an RS485 data packet. Data packets received by the core board are sent to the computer through a USART.

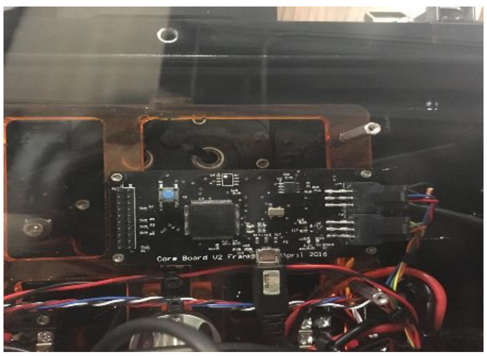

Figure 11: Core board. This standalone board handles communication between the Intel NUC™ computer and the rest of the microcontroller system.

The sensor board (Figure 12) uses an ADC to poll the Omega PX-319 depth sensor and a GPIO to poll the motor kill switch in order to provide continuous updates to the computer about the state of the vehicle. The sensor board is also used to control the two WS2812B LED strips that is used to provide visual feedback about vehicle.

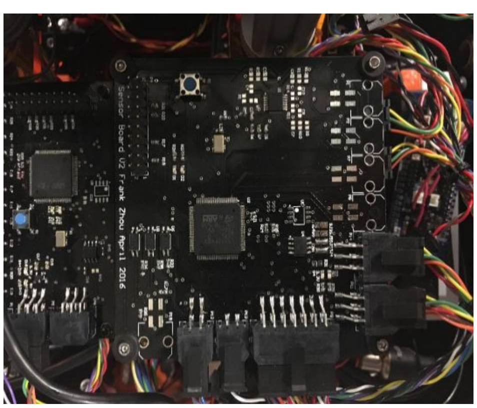

Figure 12: Sensor board. The sensor board polls the depth sensor, motor kill switch, and controls the feedback LED strips.

The servo board drives the three servo motors which control the gripper. It converts angular position information to pulse widths and asserts the PWM signals.

The solenoids board (Figure 13) uses optoisolated GPIO signals to control four electromagnets, which are used to actuate the two torpedos and two markers.

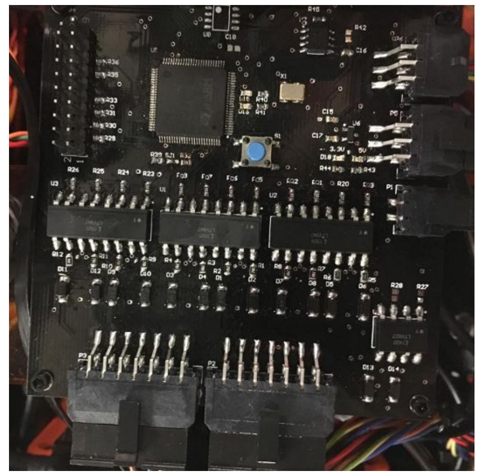

Figure 13: Solenoids board. Optoisolated GPIO signals control solenoids which actuate the torpedoes and markers.

Flo's microcontroller system consists of one core board that combines the sensor, solenoids, and servo board, and a separate hydrophone board for communications. The main core board consists of one STM32F407VG microcontroller and the previously mentioned peripherals necessary for the sensor, servo, and solenoid functions, which includes inputs for the depth sensor and kill switches, PWM outputs for the servos, and optoisolated signals for controlling the solenoids.

D. Sensors

Deb and Flo use a variety of sensors to gather the crucial information it needs to navigate and understand the underwater environment.

1. Depth Sensor

An Omega PX-319 pressure sensor filtered using a second order Bessel filter and processed by an ADC on a microcontroller measures the vehicle's depth to within an accuracy of 1-2 inches.

2. Attitude and Heading Reference System (AHRS)

A VectorNav VN-100 Rugged measures the vehicle's heading and attitude using a 3-axis accelerometer, gyro, and magnetometer. The magnetometer is calibrated with a custom hard-soft-iron calibration routine. Orientation measurements help Flo actively maintain its underwater orientation.

3. Doppler Velocity Log (DVL)

A SonTek Argonaut-M DVL on Flo and a Teledyne Pathfinder on Deb measures an accurate 3D velocity vector, which is used to estimate and control the vehicles' position in the water.

4. Camera

Flo has two AVT GuppyPro cameras, one with a 120° wide-angle lense, and the other with a 90° narrow-angle lense. This dual-lense system allows us to detect objects from further away using the narrow-angle lense, and then switch the wide-angle lense when close to the target to prevent losing sight of the target. The cameras are positioned 45 below the horizontal axis to provide full coverage of obstacles in front of and underneath the vehicle. All task accessories are positioned parallel to the cameras to maximize efficiency. Deb has one gimbaled AVT GuppyPro camera, which allows greater tracking of targets that is separate from the movement of the vehicle.

5. Hydrophones

Four Teledyne TC4013 piezoelectric transducers mounted in a square pattern form an array that listens for the sonar signals. The current from the transducers is converted to a voltage across a 100 M resistor. The signals then pass through a JFET input stage preamplifier with a DC servo to increase the signal amplitude. A 4th stage differential active filter is used to attenuate signals above 200 kHz to prevent aliasing. A quad channel simultaneously sampling 16-bit ADC is used to digitize the differential analog signals at 400 kHz / channel. The signals are then processed on a STMF407VG and the information is sent to the computer via UART. A Fourier transform is used around the pinger frequency to extract the phase information. The heading is computed using the phase differences between the four sensors.

Communication between the two vehicles will be performed using this receiving hydrophone array on both vehicles, and an additional hydrophone for transmitting from Deb. Deb has receiving arrays on both the top and bottom, to avoid situations where there is no line of sight between hydrophones on the two vehicles if one is beneath the other. Differential Quadrature Phase Shift Keying (DQPSK) will be used to encode inter-vehicle transmissions using relative phase differences, as many other analog techniques are infeasible underwater. With QPSK, two bits are encoded per symbol, with a 90 degree phase difference between each value. Differential QPSK was chosen so that only the relative phase shift is necessary, eliminating possible synchronization issues. To encode transmissions, a quadrature oscillator will be used to produce four 90 degree phase-shifted signals. The STMF407VG microcontroller on the communication boards will be responsible for controlling the quad oscillator. The inter-vehicle communication system is currently untested, but in the future we expect to achieve a bandwidth of up to 40 kbps, using DQPSK with a communication frequency of up to 50 kHz to avoid aliasing and with an ADC sample rate of 400 kHz.

V. SOFTWARE

Deb and Flo's autonomy relies on their robust C++ software architecture, which consists of the five processes diagrammed in Figure 14. These processes communicate with one another using a custom inter-process communication (IPC) library based on the publisher-subscriber data model, where processes can publish data to topics (illustrated as rounded rectangles) and subscribe to receive all data on certain topics. This makes the system modular and scalable throughout the lifetime of the code. The team chose to write its own IPC library instead of using an existing framework such as ROS in order to minimize the use of external black-boxes and reduce the learning curve for new members.

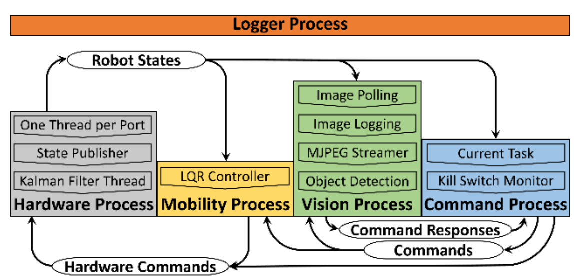

Figure 14: Software architecture. The five main processes that run on the vehicle and the interprocess communication channels are shown. Each process is illustrated by a large box, with threads inside. Each circular box represents a communication channel.

These five processes run on an Ubuntu Linux environment. While testing, we can communicate with them over Ethernet using Secure Shell (ssh).

A. State Estimation

State estimation involves determining the vehicle's orientation and position within a set of world-fixed coordinates, and it is done in the Hardware Process. Orientation is provided by the AHRS. Position can be estimated by integrating the DVL velocity measurements over time. However, the DVL only operates at 1Hz, and even then, only when there is at least one meter of water underneath it. So if we just integrate raw measurements, the update rate would be slow and the vehicle would lose control near the pool floor.

To deal with both of these issues, we implemented a custom 11-dimensional extended Kalman filter (EKF), whose state includes the vehicle's translational velocity, position, translational drag coefficients, inverse virtual mass, and net buoyant force. The prediction step uses an algebraically-linearized Newtonian dynamics model with linear drag.

A Kalman filter is well suited to this task because it can first estimate the drag coefficients of the vehicle by comparing the applied thrust to the measured velocity. Then it can use these drag coefficients to estimate the vehicle's position based on the known thrust being applied.

The output of the Kalman filter is a unified state estimate that includes the vehicle's 3-D position, orientation, translational velocity, and angular rates, all of which are periodically published to the Robot State IPC channel.

B. 6-DoF Control

The Mobility Process receives the current state and the target state from the Hardware and Commander Processes respectively. It then runs a 6-DoF controller, which decides what power level to send to each thruster in order to move the vehicle from the current state to the target state. Most RoboSub teams (and even commercial vehicles) do this using PID loops. However, on our subs we use a Linear Quadratic Regulator (LQR). LQR is an optimal control scheme for multidimensional linear systems. Unlike PID control, a LQR is particularly well suited to situations like ours where the systems have modelable, but coupled dynamics. This is because LQR requires a model of the system, but once given that model, it takes full advantage of the system dynamics.

Our 18-dimensional LQR controls the six translational and angular errors, the six integrals of those errors, and the six rates of change of those errors. Since the vehicle's dynamics are non-linear, we must locally linearize them around the target state before applying LQR. The resulting linearized system, computed anew every time the target state changes, is given to the LQR solver from MIT's Drake Library

Experimentally, this controller was far superior to even our best-tuned PID control systems. In addition, tuning the cost matrices, Q and R, is borderline trivial; it took only twenty minutes to tune our LQR controller compared to PID controllers, which took many months on previous vehicles.

The final step is gracefully addressing thruster saturation. We address this by imposing a prioritization scheme. The output of the LQR controller is decomposed into four components: (1) forces required to keep the sub static, (2) other vertical forces, (3) all other torques, and (4) all other forces. Given this breakdown, the software sums up, in this order, as much as possible of each component that can be added while staying below the thrust caps.

Finally, we profiled our thrusters using a strain gauge to determine the mapping from motor power level and battery voltage to physical thrust. Inverting this relation allowed us to generate the exact thrusts requested by the LQR controller. The entire controller is diagrammed in Figure 15.

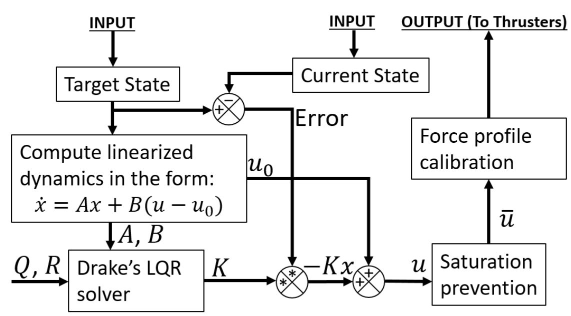

Figure 15: 6-DoF controller diagram. The dynamics are linearized about the target state to compute the LQR gain K, which is applied to the error term to compute the control output.

C. Visual Object Detection & Machine Learning

The vision process converts images into actionable information. It synchronously reads new frames from both Allied Vision Guppy Pro cameras using libdc1394. Each frame is corrected for distortions and then fed through a custom detection pipeline. For example, for the Play Slots challenge, we take the union of several thresholded images - one using a static threshold on HSV values, and the others using a dynamic threshold selecting for pixels that are more red, and less white, than their neighbors over different-sized neighborhoods - and then find rectangles within the resultant image. We then use a novel algorithm based on Gauss-Newton minimization to determine the 3D orientation and position of any rectangle given the pixel positions of its corners. The remaining detectors use an ad hoc mix of edge detection, color thresholding, contour analysis, and Haar features.

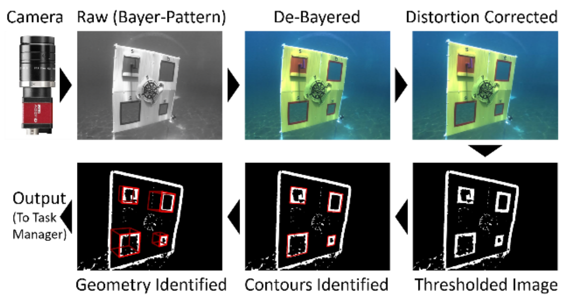

Figure 16: Typical vision processing pipeline. Applied to 2016's torpedo task to clearly illustrate the derived geometry.

New to our pipeline this year is the use of a fast, deep convolutional neural network, the you-only-look-once network, created by Joseph Redmon, to aid in detection of hard-to-track objects. We use this network to predict bounding boxes for objects such as the floating dice, see Figure 17, which we then use to narrow our field of search and better identify relevant features of the object. At time of writing, this feature is not fully integrated with the codebase, but it is expected to be completed prior to competition.

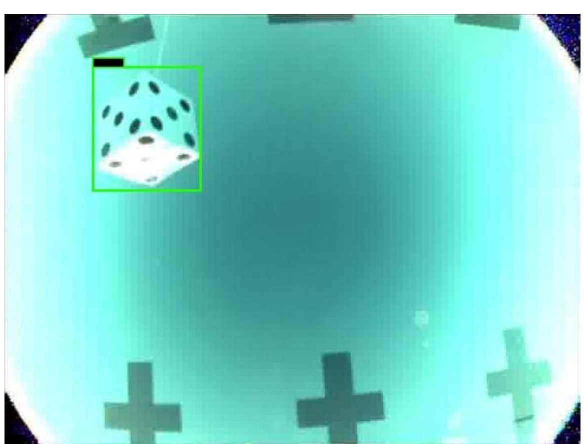

Figure 17: The you-only-look-once model's bounding box prediction. The green box is the model's prediction of the region-of-interest, i.e. where the object is thought to reside within.

D. Python Integration

While most of the software is written in C++, developing in C++ is slow, making software iteration times long. Additionally, the need to recompile and relink detectors and subsystems written in C++ when modified forces the system to be rebooted when the software changes. To mitigate these issues and speed up development, we have developed a framework by which vision detection on the submarine can be performed from Python with state and camera information as well as detections being marshalled transparently between the C++ and Python representations. This new system integrates with the existing vision and camera systems cleanly, with the C++ components exposing the same interface exposed by the existing C++ detection modules but instead forwarding detection requests to the Python backend with the necessary C++ objects exposed in a Pythonic manner.

The performance of C++ and Python detectors is expected to be comparable due to the use of the same libraries, namely OpenCV, and other libraries that perform computation in highly optimized native code, namely Eigen and NumPy, from within native C++ and within emulated Python detectors.

E. Navigation and Mission Planning

The Commander Process is responsible for knowing which task we are on, how to accomplish it, and how to get to the next task. It directs the other processes in order to accomplish this. Each task is encoded as a state machine, which makes development modular and the behavior more intelligent.

We address the problem of navigation by manually driving the vehicle through the course once to measure and save the coordinates of each task. Still, the system is only accurate enough to arrive within visual range of each task, at which point the vision system takes over.

F. Logging and Playback

In developing an autonomous system, we cannot prevent the vehicle from ever making a mistake; however, we make sure that we are always able to piece together what went wrong after the fact. This is made possible by the Logging Process, which records every message sent send over every IPC channel, in both a binary and a human-readable JSON format. The binary allows us to replay logged messages into the original C++ code, while JSON allows us to parse and analyze the logs in a scripting language (Figure 18).

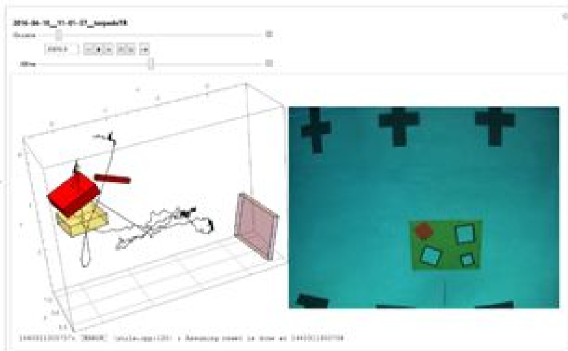

Figure 18: Logging and playback GUI. This Mathematica utility shows what the vehicle is thinking at any user-specified time, which includes the latest camera frame, the current vehicle state (red), the vehicle's target state (yellow), the vehicle's position history (black), and any object detections (pink-gray).

G. Future Architectural Changes

We have also been working on architectural changes to the software systems that have taken more than a year to develop, and thus are not currently in use. These new systems improve many aspects of the current software system, including improvements to the message passing system (making it easier to use and more flexible), the addition of a new configuration system (enabling us to use the same software base on multiple systems with different hardware and electrical configurations and capabilities), a newer, more lightweight and modern, threading system that is highly integrated with message passing system, and a remote procedure call system that enables developers to query information and enable controlling the system on-the-fly.

These changes will address continuing issues present during current development of the system, including a large message passing overhead, difficulty of controlling the scope of message passing, and the lack of fine control over the behavior of the submarine during development

VI. VEHICLE TESTING

Over the course of the year, pool tests were performed weekly at the Braun Pool at the Caltech Athletics Centre. During the course of the academic year, these tests were performed with Flo in the water, allowing for the overhaul of the software to be tested in realistic conditions in the field. These tests take advantage of replicas of the competition pool elements, which were built and designed by members of the mechanical subteam. These pool elements are often simplified to test for the specific requirements of our own strategy at the competition, allowing for them to be made more quickly and easily. This is an important consideration, since it allows for more time to be focused on the design and development of the vehicles themselves. This year, however, we found that the radical changes in the competition tasks meant that we spent more time developing the pool elements than initially expected, which hamstrung our early-year pool tests and limited them to primarily testing vehicle mobility and navigation. Furthermore, delays in machining meant that no pool tests have been conducted as of yet with Deb, and it is looking unlikely that it will be ready in time for the competition. As such, the team decided to focus our efforts on ensuring that Flo is as prepared as possible for its own tasks, reallocating time that otherwise would have spent readying Deb.

The Summer break is often a period of intensive pool testing for the team, however a unique combination of circumstances this year has seen the majority of the team - including the entirety of its leadership - moving abroad to pursue internships. This has limited our ability to perform Summer tests to two a week over the weekend. In an effort to mitigate this issue, the team has been flying down available members on weekends in an effort to provide additional manpower and expertise when needed.

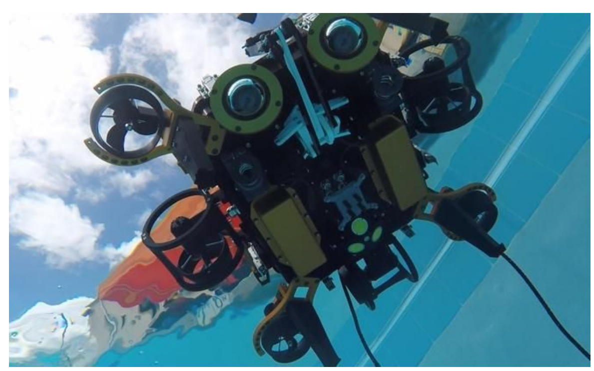

Figure 19: Pool Testing. Flo takes a close look at our camera on her way to the red buoy.

VII. CONCLUSION

Deb and Flo are integrated, performance autonomous underwater vehicles designed to be a cost-effective tool for research and education. The vehicles can perform a number of real-world tasks, such as such as visual inspection and underwater manipulation, efficiently, reliably, and completely autonomously. Further work includes adding additional autonomy, logging, and debugging capabilities to further enhance and tune the vehicles' autonomous capabilities.

VIII. ACKNOWLEDGEMENTS

CRT would like to thank Caltech for all its support during the past academic year, particularly within the Engineering and Applied Science division. We would like to specially thank our faculty advisor and mentor, Professor Joel Burdick, who has given us direction through our many hours of development and testing. Additional thanks goes out to Perry Radford, Jannah Maresh, Matt Burkhardt, John Van Deusen, Professor John Dabiri, Heather Morgan, Betsy Mitchell, Joe Jordan, John Leichty, Glen George, Richard Murray, Tom Mannion, and all the gym and pool staff that have helped us through numerous hours of testing. Finally, we express our immense gratitude to all of our gracious sponsors for the 2017-2018 academic year.

Ocean Level: Caltech Moore-Hufstedler Fund and Caltech Housner Fund, Rhonda MacDonald, Glendale Community College, VideoRay, Rigol, Northrop Grumman
Sea Level: Caltech Student Investment Fund, Danco Anodizing, Teledyne Impulse, Edward Fouad, Jeffrey Orenstein
Lake Level: David Berg, Prue Flicker, Justin Koch