California Institute of Technology RoboSub Technical Design Report

Brendan Hollaway, Irene Crowell, Cormac O'Neill, Mohith Manohara, Elaine Lowinger, James Deacon, and the rest of the Caltech Robotics Team

***Abstract*—The Caltech Robotics Team's autonomous underwater vehicle (AUV) Flo is designed for precise maneuvering at the 2019 RoboSub competition. Designed and manufactured by 25 undergraduates at the California Institute of Technology, Flo is the submarine we brought to last years competition, with a new gripper, and redesigned marker and torpedo systems which replace magnetic actuation with servo motors. New algorithms for strategy and machine learning based vision, along with improved robustness of electrical systems make Flo more intelligent and reliable than ever. Her systems together allow for careful navigation of the TRANSDEC course and precise interaction with game elements.**

I. Competition Strategy

Based on an analysis of the point values of various tasks, and our experience at previous years' RoboSub Competitions, our team selected tasks that we believed we could feasibly do, while maximizing points. Our main task sequence, with stretch goals in italics, consists of

(1) Passing through the gate
(2) *Grabbing a garlic element*
(3) Following the path
(4) Hitting buoys (flat and called triangle side)
(5) Swapping the bin lid, *dropping the garlic element* and markers
(6) Following the random pinger to one of (7) or (8), and then the other
(7) Sliding the bar on the torpedo board, and *shooting torpedos in the ovals*
(8) *Opening the casket, grabbing the vampire*, and surfacing

Because the gate, path, and marker bins are recurring competition elements, we felt confident in our ability to repeat our performance on these tasks from last year and quickly get those points. In addition, we had experience following the pinger accurately, and based on the large point value of the random pinger task, we felt it was important to include this as part of our efforts. In order to receive random pinger points, it would be necessary to score points at the torpedo task and the surfacing area. We decided that simply surfacing above the pinger and pushing the torpedo board bar would be the simplest ways to make sure we could do this. Together, these tasks were those we felt could contribute a good source of points, while requiring a reasonable amount of time, mostly on the part of our software team, since they required no new electrical or mechanical systems.

Based on the results of last years competition, we knew that in order to effectively perform the buoy task, a new machine learning based approach to vision was required. This meant that the software team would need to put a large amount of time into this project, but that time spent would help with both the buoy and torpedo tasks, and be invaluable in future years. Based on this, we deemed the point value and other benefits worth the added effort, and made sure to have a reliable image classification framework, as discussed in the Software section.

Finally, the tasks that we felt would require a much larger amount of effort, but that we still wanted to attempt, included shooting torpedos and using manipulating objects including the garlic, the casket, and the vampire. Doing this would require two new mechanical system in the form of a gripper, and a more robust torpedo system. One important difficulty faced here was a lack of ports to power actuators on our vehicle. Since we were confident in our ability to score points using our marker system, and had seen in previous years how difficult accurate torpedo shots are, we decided to dedicate our effort and remaining ports to creating a gripper. As a final stretch goal, we decided to research non-actuated ways to place objects through the torpedo holes, with a lower priority placed on this task, in order to better dedicate team time to getting a working gripper.

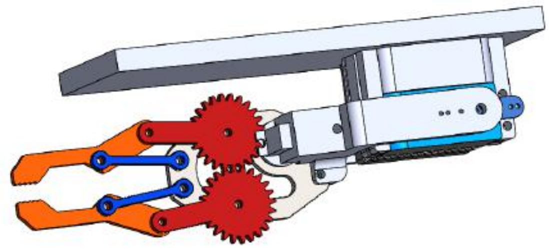
Another resource management decision the team made related to dedicating efforts towards creating a new vehicle (Deb), versus maintaining and improving our current vehicle (Flo). Having an existing, working, vehicle proved immensely helpful for the software team at the beginning of the year, allowing them to test new algorithms in the pool without waiting for any manufacturing, while the mechanical and electrical teams could dedicate their time to the new vehicle. However, close to competition, it became clear that the issues that continued to arise with Flo's aging systems, which required fixing for software testing to continue, were taking too much time away from the construction of Deb. The decision to halt work on Deb was a tough one, but it meant that the team could feel confident they would have a working robot at competition, and avoid the need for a large software time sink in transferring working code to a new platform. This decision helped us maximize our in-water test time, allowed us to develop new components like the gripper and torpedos, and provided increased systems reliability.

II. Vehicle Design (Novel Aspects)

A. Mechanical

While designing the gripper for our vehicle, we emphasized simplicity above all else. In order to effectively pick up the cylindrical handles on the garlic and the crucifix, a two-pronged claw was developed that could be opened and closed by a single waterproof servo. The use of a waterproof servo removes the need for a heavy pressure vessel to protect it from the outside environment, reducing the amount of torque which is needed to actuate the gripper. In an ideal world, a simple gripper might only need to consist of the end effector itself, but Flo's crowded underside and the requirement not to obstruct our cameras mandated that we introduce an additional degree of freedom into our design. This allows for the gripper to be moved in and out of the cameras' frame, thereby removing the obstruction to Flos vision during regular operation, while also allowing vision to be used to verify an effective grab during gripper tasks. Finally, the introduction of this degree of freedom has the added benefit of allowing the gripper to extend below the vehicle's "feet" during operation, while stowing itself safely in front of the vehicle at other times.

Fig. 1. CAD of 2018-19 gripper design.

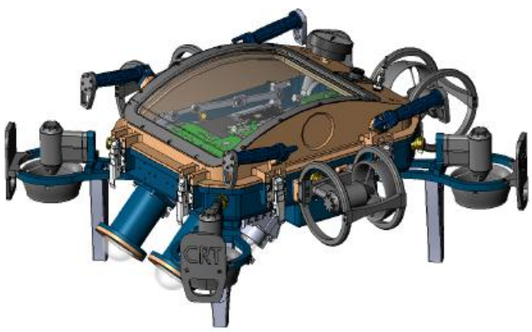


Last year, it was determined that the use of solenoids and permanent magnets to actuate our markers and torpedoes led to localization and navigation issues for the vehicle. We hypothesize that the varying magnetic fields from the movement of the magnetic components affected the calibration of our magnetometer. In order to mitigate this problem, the torpedo and marker launchers were redesigned to be triggered by waterproof servos. Unfortunately, the limited number of through-holes built into Flos pressure hull restricted us to 3 servos, with 2 being set aside for the gripper. As such, we leveraged our earlier work with golf balls as markers to design a servo-actuated marker dropper that can fire both markers at once with a single servo. This allows us to use our servos in the most efficient way possible while also reducing the magnetic interference affecting our vehicles navigation capabilities. This lack of servos also lead to a redesign of our torpedoes. Instead of our previous active magnet system, the new torpedo design uses permanent magnets further from the Flo to hold the torpedoes in place until hitting the torpedo board to dislodge and drop them.

B. Electrical

The design of the electrical system aims to allow the computer to communicate with the various sensors and motors on the sub while also isolating potential problems in the sub. The electronics are soldered on PCBs, which have various functions ranging from supplying power to the other boards to interfacing with the sensors and servos.
The sensors and servos communicate to the computer through an intermediary core board. The computer sends a UART signal to the core board, which

Fig. 2. CAD of Flo in 2019.



then processes the request and relays it to the appropriate board to handle. These intermediate boards include a sensor board to send depth information and handle shutdown requests, and a servo board to send signals to the gripper and marker actuators. The signals are transmitted using an RS485 bus. Because each board communicates separately with the core board, a problem on one of the boards does not affect any of the other boards, making debugging easier. Each of these boards has an STM32F4 microcontroller that is programmed for that board's specified function.

The sub is powered by two 26 volt LIPO batteries. There are two boards that take the battery input and supply power to the rest of the sub. One of them powers the computer and board microcontrollers, and the other powers the thrusters and servos. We decided to isolate the computer and motor supplies to prevent surges from one supply affecting the performance of the other supply. The motor power board supplies 26 volts directly from the LIPO battery to the thrusters. It also uses a switching buck regulator to convert the 26 volts to 5 volts, in order to power all of the servos on the exterior of the sub. Meanwhile, the computer power board supplies 12 V to the computer and 5 V to each of the boards. Because the STM32F4 microcontroller is powered by 3.3 V, every board has a voltage regulator that converts the 5 V provided by the power board to 3.3 V.

C. Software

1) State Estimation: Using our DVL velocity measurements and AHRS orientation data as inputs, we craft an 12-dimensional extended Kalmann Filter (EKF) to estimate our sub's pose in the water. This EKF leverages the sub's dynamics to predict the sub's motion through the water, even between important sensor updates such as the DVL, which only fires once per second. An EKF is well suited to this task, as the sub is moving slowly enough that many effects are approximately linear (thus fitting the EKF's conditions) for the timesteps we are working under.

2) Control: Rather than use standard PID control to guide our robot through the water, we implemented a Linear Quadratic Regulator (LQR) with the use of Drake's LQR solver [1]. An LQR is a provably-optimal control scheme for multidimensional linear systems. It achieves this optimality by leveraging a model of the sub to take 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 vehicles dynamics are non-linear, we must locally linearize them around the target state, discretizing our system, before solving for the optimal LQR gain K. Then, we analyze our computed controls with an eye towards gracefully addressing thruster saturation. To do so, we impose the following the 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 thrusters' thrust caps.

Experimentally, this controller was far superior to even our best-tuned PID control systems. In addition, tuning the cost matrices for the state errors and the controls, 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.

3) Visual Object Detection: Our object classification algorithms rely on leveraging a mixture of classical higher order features such as colours, contours, and edges, along with point features such as SIFT descriptors, machine learning approaches such as the Convolutional Neural Net (CNN) You Only Look Once (YOLO) [2], and a novel Gauss-

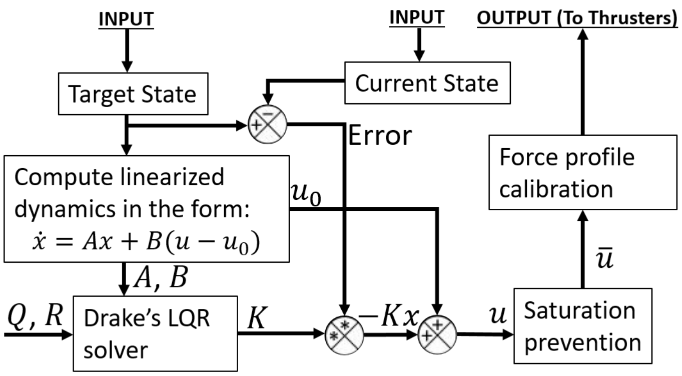


Fig. 3. 6-DoF controller diagram. The dynamics are linearized about the target state to compute the LQR gain, which is applied to the error term to compute the control output.

Newton minimization algorithm used to localize and verify the structure of rectangular detections.

While we are approaching tasks from a great distance, or tasks that have very complex features, such as the vampire buoys, we have found the YOLO CNN to be the most effective. However, it is unable to extract orientation information from the target, and it is also very slow (1 FPS on the Intel Nuc), meaning that other approaches are necessary for fine-grained approaches closer to the target. Once we are close to the target, colors and other high-level features such as circular and rectangular shapes become more clear, allowing us to use different static or adaptive thresholding techniques to identy them. Once identified, if we are able to see any rectangular shapes, such as the outline of the Drop Garlic bins, we can find the 3D orientation of the object. This is done using multipoint Gauss-Newton minmization to find how skewed and rotated the object is, thus allowing us to figure out where a normal vector facing out of the object would lie. With this information, we are able to align to the target more cleanly, and complete up-close tasks. We can also use this reprojection to identify and reject outliers; for example, if we find a detection that claims that the bins are facing sideways, we know (or at least certainly hope!) that this is a misdetection and should be ignored.

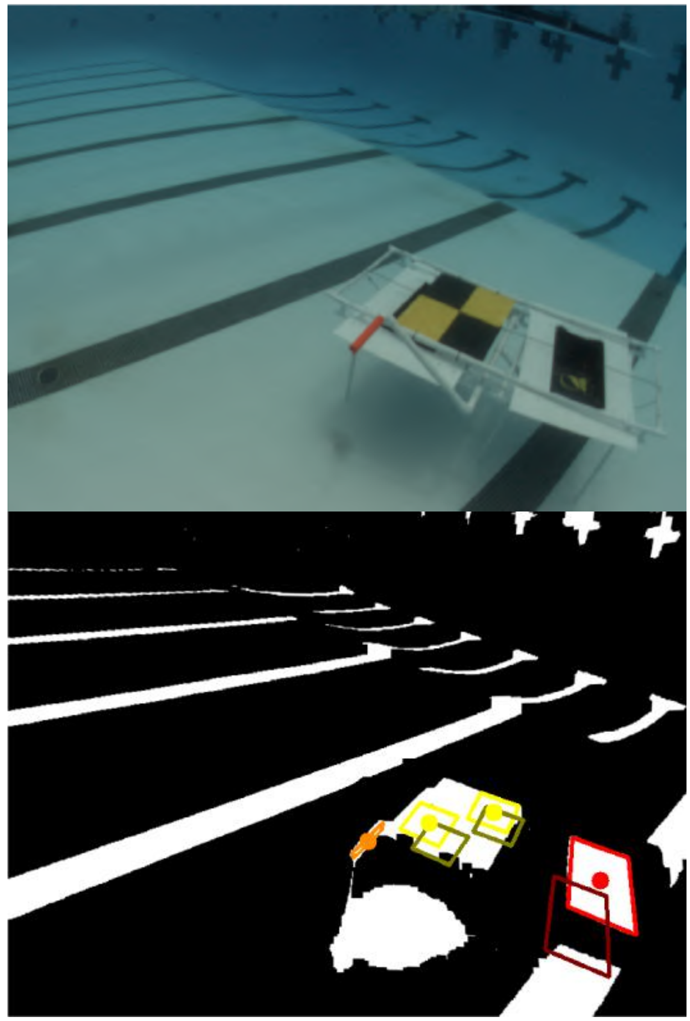


Fig. 4. Example of detector using Gauss-Newton minimization to extract the orientation of the marker bin and the marker bin cover. The dots represent the center of each validated contour, and the brighter rectangles represent the detected rectangular region. The darker rectangles are the projection of the estimated pose; they are what is ”behind” the detected rectangle. Here, the red rectangle is the black inside of the bin, the yellow rectangle is the yellow rectangular portions of the cover, and the orange is the bin handle. The white regions are potential areas of interest.

III. Experimental Results

In-water testing of our vehicle takes place in Caltech's Braun Pool. Reserving two lanes of the 25 yard long outdoor facility, we are able to test Flo's systems and perform code changes on the fly through an ethernet tether connected to pool-side equipment. Sharing the pool with other users does limit our ability to string multiple tasks together in a single test, but the opportunity to debug the vehicle in the water has proven invaluable during our design process. In particular, experience during testing helped us make design decisions that allow for for safer vehicle operation, and faster identification of failures before they can become catastrophic.

A primary example of this is our use of positive pressure inside the AUV. Even though the changes in relative pressure between the inside of the hull and the surrounding environments put more wear on the O-ring seals versus if we were to negatively pressurise the vehicle, the positive pressure allows us to monitor Flo for any bubbles that would warn us of an ongoing leak.

Even more valuable than the hardware lessons learned during in water testing are the software lessons. Due to limited access to pool time, our software team saves logs from each run in the pool, including all visual footage. Because objects look so different underwater, it is important to write vision algorithms designed around accurate photos, and train our machine learning models on the same. Capturing footage in advance and debugging later saves invaluable amounts of in water time. The four hours we spend on average each week at the pool can then be used to debug the strategy and motion of the submarine.

Acknowledgment

The Caltech Robotics team would like to thank all of our sponsors for their support

Ocean Level Sponsors:
Caltech
Glendale Community College
Northrup Grumman
Video Ray
Rhonda MacDonald

Sea Level Sponsors:
Dropbox
Danco
Team Grandma

References

[1] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019. [Online]. Available: https://drake.mit.edu

[2] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[3] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[4] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[5] G. Guennebaud, B. Jacob *et al.*, "Eigen v3," http://eigen.tuxfamily.org, 2010.

Appendix A: Expectations

TABLE I
Expected Scores at competition

Subjective Measures			
	Maximum Points	**Expected Points**	**Points Scored**
Utility of team website	50	35	
Technical Merit (from journal paper)	150	130	
Written Style (from journal paper)	50	45	
Capability for Autonomous Behavior (static judging)	100	90	
Creativity in System Design (static judging)	100	80	
Team Uniform (static judging)	10	10	
Team Video	50	45	
Pre-Qualifying Video	100	100	
Discrectionary points (static judging)	40	20	
Total	**650**	**555**	
Performance Measures			
	Maximum Points	**Expected Points**	**Points Scored**
Weight	See Table 1 / Vehicle	10	
Marker/Torpedo over weight or size by $<$ 10%	-500 / marker	0	
Gate: Pass through	100	100	
Gate: Maintain fixed heading	150	150	
Gate: Coin Flip	300	300	
Gate: Pass through 60% section	200	0	
Gate: Pass through 40% section	400	400	
Gate: Style	+100 (x8 max)	400	
Collect Pickup: Crucifix, Garlic	400 / object	800	
Follow the Path (2 total)	100 / segment	400	
Slay Vampires: Any, Called	300, 600	600	
Drop Garlic: Open, Closed	700, 1000/ marker	4000	
Drop Garlic: Move Arm	400	400	
Stake Through Heart: Open Cover, Cover Ovel, Sm Heart	800, 1000, 1200 / torpedo (max 2)	0	
Stake Through Heart: Move lever	400	400	
Stake through Heart: Bonus - Cover Oval, Sm Heart	500	0	
Expose to Sunlight: Surface in Area	1000	1000	
Expose to Sunlight: Surface with object	400 / object	400	
Expose to Sunlight: Open coffin	400	400	
Expose to Sunlight: Drop Pickup (crucifix)	200 / crucifix	200	
Random Pinger first task	500	500	
Random Pinger second task	1500	1500	
Inter-vehicle Communication	1000	0	
Finish the mission with T minutes	Tx100	500	
Total	-	**12,460**	

APPENDIX B: COMPONENT SPECIFICATIONS

Component	Vendor	Model/Type	Specs	Cost (if new)
Bouyancy Control	n/a	n/a	n/a	n/a
Waterproof Housing	Glendale Community College, DANCO	Custom Hull	6061-T6	Re-used
Waterproof Connectors	MacArtney	SubConn MCLPBH3F	3 pin	Re-used
Thrusters	VideoRay	M5		Donated
Motor Control	Built into the thrusters			
High Level Control	LQR controller, uses (in part) Drake's LQR solver [1]			
Actuator 1	HiTec	HS-5086WP	IP67 50oz-in	52.89
Actuator 2	Savox	SW0250MG	69.4oz-in	30.99
Propellors	Videoray	M5	Max. Thrust (nominal): 10kg	Donated
Battery	Turnigy	LI-PO	129.5 Wh, 5000 mAh, 25.9 V	Re-used
Convertor	Custom, built into the boards			
Regulator	Custom, built into the boards			
CPU	Intel	NUC	8 core processor	Reused
Internal Comm Network	-	UART	-	
External Comm Network	-	Ethernet	-	roughly $150
Programming Language 1	-	C++	-	-
Programming Language 2	-	Python	-	-
Compass	VectorNAV	VN-100 Rugged	800Hz data rate	Re-used
Inertial Measure Unit (IMU)	VectorNAV	VN-100 Rugged	800Hz data rate	Re-used
Doppler Velocity Log (DVL)	Teledyne	Pathfinder	12Hz data rate	$15,000.00
Camera(s)	Allied Vision	Guppy Pro F-046	62fps	Re-used
Hydrophones	Teledyne	RESON TC-4013	1Hz-170kHz	Re-used
Manipulator	n/a	n/a	Custom design, 3D printed	Free (Free printing)
Algorithms: vision	See Software section on vision. Wide variety of tools.			
Algorithms: acoustics	Using phase-angle to find the direction of the accoustic pinger			
Algorithms: localization and mapping	Waypoint map of course. Localize using pinhole approximation.			
Algorithms: autonomy	Overall system is a series of unidirectionally linked finite state machines.			
Open Source Software	ROS [3], OpenCV [4], YOLO [2], Drake [1], Eigen [5]			
Team Size (number of people)	25			
HW/SW expertise ratio	12 programmers, 10 mechanical engineers, 2 electrical engineers, 2 Business team members			
Testing time: simulation	200 hours (vision algorithm simulations)			
Testing time: in water	140 hours			

Appendix C: Outreach Activities

We strive to spread the love for Robotics and STEM within our local area. This year, our two biggest outreach events were collaborations with some local Girl Scout troops, and with students from Escondido Charter High School's FRC team, team 2839 the Daedalus Project. More photos of both events available upon request.

At a local STEM and Robotics expo organized by the Girl Scouts, which was targeted at young girls and Girl Scouts in the area, we hosted a booth that was designed to give very young girls (4-8 years old) an introduction to what robotics can be like. We helped them craft cup-bots, which use a miniature motor and a slightly-off-center popsicle stick to "dance" across the table, as the off-center popsicle stick jerks it around. They decorated the cups to their hearts' contents, adding pipe-cleaner arms and googly-eyes (and in one case a very demonic expression). The older girls then were able to help wire up their robot (read: feed a wire up to a metal connector and wrap it around), while we helped the younger girls get theirs set up. Finally, they were able to flip the switch and watch their robots dance! It was truly wonderful seeing how excited many of them were, and I sincerely hope that this shows all of them that they have the option to pursue robotics in the future.

Fig. 5. Helping young girl scouts get their first experience with robotics. As can be seen in the second image, the popsicle stick on top of the cup spins rapidly, allowing the cup robot to "dance" when placed on a table.

We also invited down FRC team 2839 from Escondido Charter High School to tour our lab and learn about our approach to robotics. We gave them a presentation on what RoboSub is, and thus what one example of robotics they could look forward to in college would be, as well as to how our sub worked. They were able to ask us questions about how various sensors worked, and our rationale behind various component designs and task strategies, so that they could then take those lessons back to their own team and use them to grow and develop. Overall, they learned a lot, and we sincerely hope that we were able to encourage them to continue to pursue robotics in the future!

Fig. 6. Giving a lab tour and presentation about robotics and RoboSub to Escondido Charter HS students.