Amador Valley HS RoboSub Team: Development of Marlin

Brody West (P); Mechanical: Kelly Chen (VP), Nicholas Maslennikov, Richard Ho, Trey Cranney, Shawn Zhang, Varun Srinivasan, Calvin Qin, Jonathan Ye; Software: Luke Shimanuki (VP), Brian Kim, Varun Iyer, Michael Wan, Michael Chen, Evan McNeil, Seth Girvan, Jeffrey Chen, Rahul Gupta, Benjamin Liu, Daniel Yang, Jonathan Yang, Surya Ramesh, David Zhang Brandon Beanland; Electrical: Eric Pianin (VP), Wonjoon Lee, Shreyas Krishnaswamy, Sarah Tandean, Timothy Kanarsky, Grant Ye; Business: Brendan Daft, Lucas Relic, Matthew Xie, Andre Londono.

Abstract-- This paper presents Marlin, AVBotz's latest autonomous underwater vehicle (AUV), which was built by a team of 40 students from Amador Valley High School. This year AVBotz built Marlin completely from the ground up, rather than simply upgrading Barracuda as it has done for the past 13 years. As AVBotz's most ambitious project to date, Marlin provides a suite of new features over previous designs, including increased speed, six degree-of-freedom control, a torpedo launcher, a grabber, improved mechanical modularity, and upgrades to every sensory and computing component. Marlin's software has also been completely overhauled, with a distributed architecture, a dynamic mission control system, plug-and-play operation of peripherals and processes, Bayesian modeling, redesigned low-level controls, and newly written vision and signal-processing modules. To fund this extensive project, the business subdivision has conducted the largest fundraising campaign in team history, securing many more sponsorships than in years past. Given these upgrades, Marlin is AVBotz's most capable AUV yet.

I. DESIGN STRATEGY

Marlin is intended to serve as an extensible platform for the next line of AVBotz AUVs and to last for several generations of AVBotz members. Therefore, it was designed to ensure reliability, longevity, and modularity.

Marlin's mechanical structure is robust and modular, consisting of a framework of anodized aluminum panels, each of which can be modified and replaced without having to remachine adjacent pieces. Additionally, extra room in the pneumatics box and frame allow for future modifications and improvements. Marlin's electrical infrastructure is a combination of standard products augmented with customization. Extra connectors on the rear endcap, large electronic shelves, a modular control board, and a highly versatile computer motherboard were chosen for future upgradability. Marlin's software stack is also modular and upgradable. The software team chose a distributed architecture, split up into separate programs for each high level function, such that modules can be easily added or removed, enabling rapid feature development and external device integration. Furthermore, Marlin maintains a dynamically updating probabilistic model of its surroundings, improving its ability to robustly navigate through the course in spite of the occasional bad observation. Extensive documentation for all aspects of Marlin is prepared and archived online for future AVBotz teams to further improve the AUV.

II. VEHICLE DESIGN: MECHANICAL

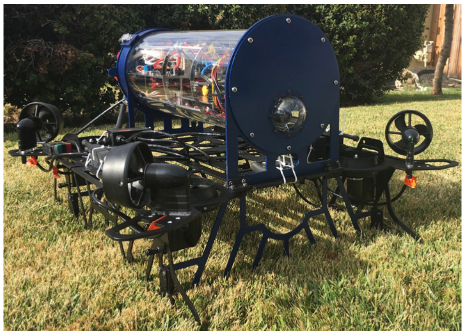

Fig. 1: Marlin

The mechanical team is responsible for all of Marlin's structural components. Its goal this year was to design a vehicle that was more capable than past vehicles, with an emphasis on constructing a modular platform that future teams can improve on. Mechanical uses many tools, such as SolidWorks CAD and Simulation, 3D printers, CNC machines, lathes, and drill presses.

A. Frame

The frame is built from 6061-T6 aluminum components for high strength and relatively low weight. Sheets and L-pieces are machined using CNC machines and standard industrial tooling machines, and then anodized with a type II coating. A horizontal plane provides secure mounting points for the hull mounts, surge thrusters, dropper, torpedoes, and many other components, while four vertical side panels support the weight of the plane and provide mounting points for the vertical thrusters, grabbers, and hydrophones. Additional panels were added in the front and back of Marlin to avoid warping the horizontal plane under load.

B. Electronics Rack

The electronics rack is a structure of machined aluminum sheets which houses internal electrical components and screws into the interior of the back endcap. The mechanical team chose aluminum to dissipate heat through the back endcap and into surrounding water. All of the components supported by the electronics rack are securely mounted, allowing Marlin to operate in any orientation.

C. Hull

Marlin's hull is a 9.5-inch diameter acrylic tube, 25.875 inches long, and sealed by two anodized 6061-T6 aluminum endcaps. The front endcap is sealed to the end of the tube with 3M DP420 epoxy, and the front camera dome is sealed using a face o-ring. An aluminum collar was added to the back of the tube, so that the back endcap seals to aluminum instead of casted acrylic, ensuring seal integrity. The back end cap uses o-rings 3% smaller than normal specification and was given a 10-degree chamfer to ease insertion and extraction of the electronics rack [7]. Four spring-draw latches on the rear of the hull help keep the back endcap seal watertight by preventing the endcap from sliding out of the tube. Thirteen waterproof SubConn connectors interface between Marlin's internal and external electronics, and can be quickly detached and reattached without disrupting the watertight seal. Two aluminum mounts attach the hull to the horizontal plane of the submarine.

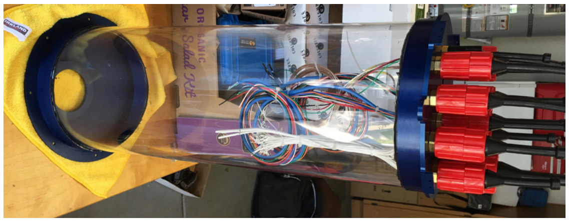

Fig. 2: Hull and Endcaps

D. Thrusters

Marlin maneuvers using eight VideoRay M5 thrusters, chosen for their high thrust and power efficiency. Four vertical thrusters control depth, pitch, and roll. The four angled thrusters are arranged at 45-degree angles and control strafe, yaw, surge movements. This configuration facilitates front camera field-of-view and increased speeds. The current thruster configuration also leaves room for future integration of a sonar and Doppler Velocity Log (DVL).

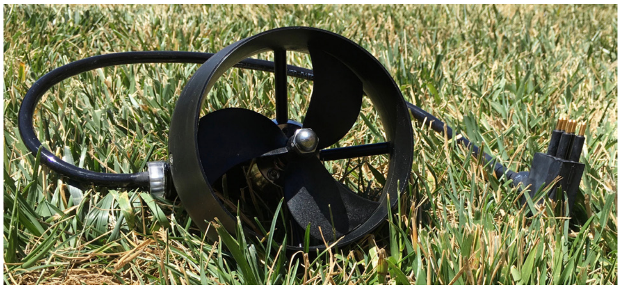

Fig. 3: VideoRay M5 Thruster

E. Pneumatics

Marlin is intended to complete all manipulation tasks through pneumatic actuation. Marlin's solenoids are housed inside an aluminum box with water-tight cordgrips, while its air canister and pressure regulator are housed in a 3D-printed ABS scaffolding enclosed in a Blue Robotics waterproof enclosure.

F.. Dropper

The dropper is an external mechanical structure made of 3D-printed polylactide (PLA). There are two main parts: a container screwed on to Marlin's frame and a sliding piston attached to a double-acting pneumatic actuator. The dropper is loaded with two one-inch diameter stainless steel ball bearings. The first bearing is dropped when the actuator is activated, aligning a hole in the bottom tube. The second ball bearing is loaded when the piston head returns back to its resting position. This design allows us to drop two markers with only one linear actuator.

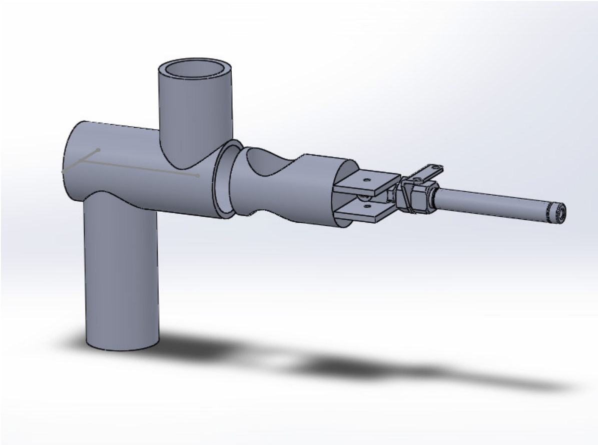

Fig. 4: Pneumatic Dropper

G. Grabber

Each of Marlin's two grabbers is mounted to a side panel and aligned with the down-facing camera. The grabbers are made of aluminum and cut with an electrical discharge machine (EDM). Each grabber consists of a double-acting pneumatic linear actuator connected to two sets of rack gears that mesh with cogs that have arms attached. As the piston extends, it actuates the rack gears, which turns the cogs, thus rotating the arms. This rotation opens and closes the grabber.

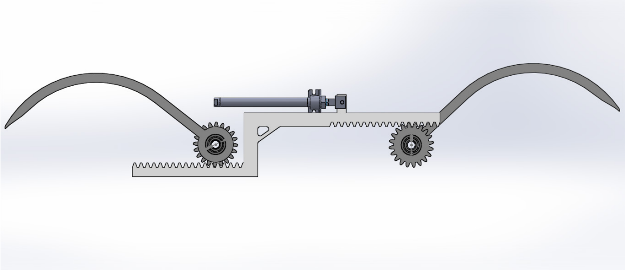

Fig. 5: Grabber

H. Torpedo Launcher

Marlin's torpedo system is also driven by pneumatic linear actuators. The torpedoes rest inside a 3D-printed PLA tube with o-rings that holds them in place before launch. They are launched by being hit by the piston rod. The torpedoes were designed so that their center of mass is located halfway down their length and were 3D-printed from PLA with 100% infill so that they are roughly neutrally buoyant.

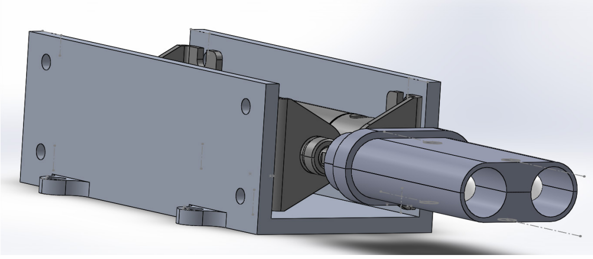

Fig. 6: Torpedo Launcher

I. Kill Switch

Marlin uses a Carling Technologies sealed switch to kill power to the thrusters while underwater. The wiring is placed inside a potting box and is sealed with MG 832C translucent epoxy for waterproofing. The switch is small enough for single-hand operation and is attached near one of the rear handles.

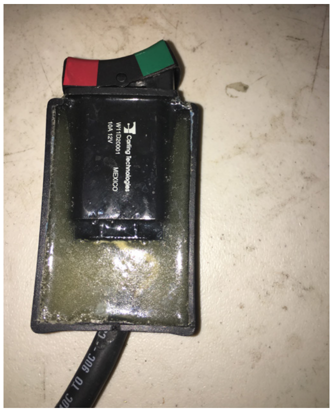

Fig. 7: Kill Switch

III. VEHICLE DESIGN: ELECTRICAL

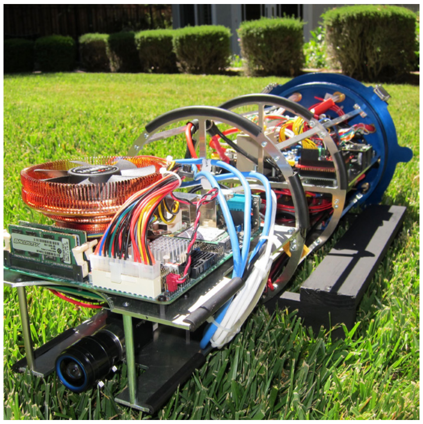

Fig. 8: Marlin's Electronics Rack

This year, electrical developed Marlin's all new electrical infrastructure. This system is intended to be more powerful and capable than any previous system to accomadate this year's mechanical and software changes. To accomplish this, Marlin has larger batteries, a more powerful main computer, and a highly versatile passive acoustics system.

A. Batteries

Marlin is powered by two Venom 22.2V 22,000 mAh Lithium Polymer Batteries connected in series. These batteries were chosen for their high energy density and ability to supply large amounts of instantaneous current. The batteries provide a nominal 44.4V and have an energy capacity of about 975Wh, which allows Marlin to run for 3-4 hours of continuous water testing.

B. Power Supply

Besides the thrusters, which run off of 44.4V, all other components run off of 24V, 12V, or 5V. To satisfy this, Marlin utilizes a two tier system to regulate battery voltage. The first tier utilizes a Cincon DC/DC converter to convert battery voltage to 24V. The Cincon DC/DC converter uses the aluminum electronics rack for heatsinking. The second tier takes the 24V output of the first tier and uses an ATX power supply to provide power to the computer as well as 12V, 5V, and 3.3V rails.

C. Power Distribution

To ensure proper power distribution, Marlin has a series of switches and fuses. A kill relay, which is remotely switched by the kill switch, controls power to the thrusters. A 100A fuse on the thruster power line provides overcurrent protection to the bulkhead connectors. The 24V, 12V, and 5V rails are also individually fused for overcurrent protection.

D. Bulkhead Connectors

Marlin uses MacArtney SubConn underwater connectors to interface with all external devices (*e.g.*, thrusters, hydrophones, and the pneumatics system).The connectors carry all power and signal I/O through the rear endcap. The connectors range from 2 to 21 pins and have a variety of applications such as coaxial and high power. The connectors are easily removable, which contributes to Marlin's modularity.

E. Cable Splicing

Since the SubConn cables are not long enough in some cases, 3M Scotchcast and Polycase potting boxes filled with MG Chemicals 832C compound are used to create neat and compact cable splices. There are two potting boxes on the sub: Thruster Merge and Hydrophone Merge. To accommodate Marlin's eight thrusters, a potting box was used to compactly and neatly splice power and RS-485 lines. Because the vertical thrusters use only half of their maximum power, their connections were stacked using Teledyne Impulse connectors to save space and simplify cabling.

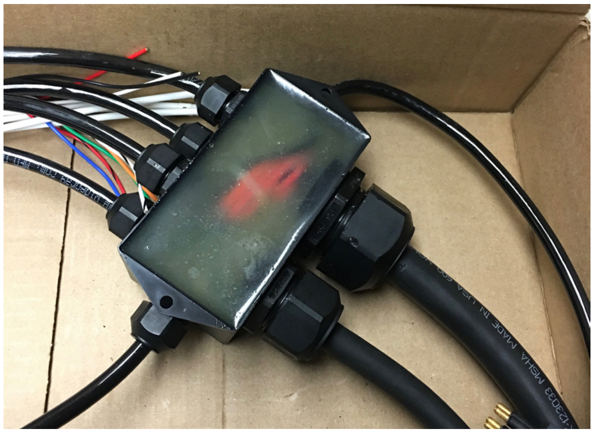

Fig. 9: Thruster Merge Box

F. Main Computer

Marlin's main computer is built on a Jetway NF9J-Q87 Mini-ITX motherboard. It contains a 2.7 GHz quad core Intel i7-4790T Haswell processor, 8GB of RAM, and a 120GB mSATA SSD. The main computer handles high-level functions such as image processing. The NF9J-Q87 was chosen for its large number of USB ports and its onboard RS-232 ports. The i7-4790T was chosen for its combination of high processing power and low power consumption. At full power, the main computer uses just 60W. To handle the heat generated by the main computer, a heat pipe CPU cooler is used.

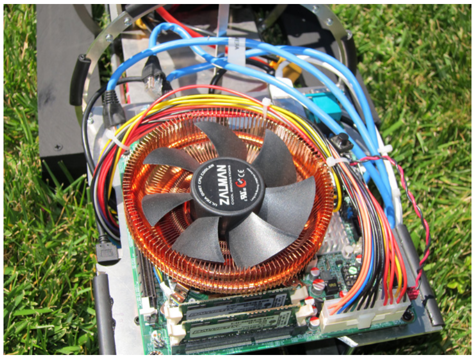

Fig. 10: Main Computer

G. Control Board

Marlin's control board is based on the ATmega 2560. The ATmega 2560 is a 16MHz 8-bit AVR RISC-based microcontroller with 256KB of flash memory. It was chosen for its simple programming environment and wide variety of I/O. The Mega handles all of Marlin's low-level tasks, such as navigation and motor control. Custom serial converters and kill state sense circuits allow the ATmega to interface with all of Marlin's sensors.

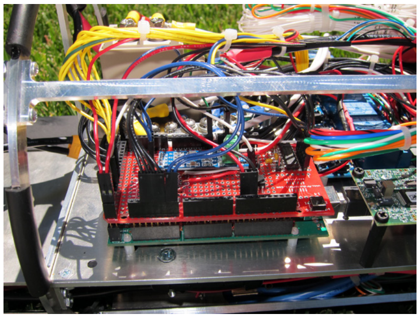

Fig. 11: Control Board

H. Motor Control

The ATmega 2560 controls Marlin's thrusters over a bussed RS-485 interface. The bussed interface allows for a simplistic software setup and saves pins on the rear bulkhead. The motor's electronic speed controller is housed internally, which prevents heat buildup inside of the main hull.

I. WiFi Tether

Marlin's WiFi tether contains a NetGear router and a rechargeable battery pack in a waterproof pelican case. The access point is wired to the main computer through a 100ft Cat6 ethernet cable and a SubConn ethernet connector through the rear endcap. The tether allows us to upload software changes to the main computer and control board without having to interrupt testing.

J. Navigation Sensors

Marlin uses a PNI TRAX attitude and heading reference system (AHRS). The TRAX AHRS has a 3-axis magnetometer, 3-axis accelerometer, and a 3-axis gyroscope. The TRAX has an onboard Kalman filtering algorithm that provides accurate heading under a wide variety of conditions, including the ability to overcome errors normally caused by erratic motion and/or changes in the local magnetic field. The TRAX communicates with the ATmega 2560 over RS-232.

Marlin's pressure sensor is the Ashcroft Model K1 Pressure Transducer/Transmitter. It converts pressure sensor reading into voltage readings. These readings are linearly converted to depth on the ATmega 2560.

K. Cameras

For visual input, Marlin uses front-facing and downward-facing 1.3 MP Point Grey Blackfly machine vision cameras with Theia Technologies SY125M lenses. The SY125M lenses give us a 125° field of view with very little distortion. The cameras interface with the main computer over USB and ethernet.

L. Actuator Control

Marlin uses an 8-channel relay module to control the various actuators on the vehicle. The relays switch 24V to power the solenoids and are controlled via the digital output pins on the control board. An 8-channel module gives us the capacity for future expansion.

M. Signal Processing

To capture audio from the pinger, Marlin uses four Teledyne Reson TC4013 Hydrophones. This year, the National Instruments USB-7855R OEM serves as Marlin's digital signal processor (DSP). A custom amplifier and high pass filter conditions the signals before they are processed on the DSP. Marlin's op-amp is built around the Texas Instruments TLE2074CN and results in a gain of 40dB. After amplification, the signals are pass through a high-pass filter designed around the TLO84CN. The high-pass filter has a cutoff of about 10 kHz and filters out any noise not generated by the pinger. The signal conditioning circuit is located inside of a shielded box and powered by two small 9V batteries to avoid interference.

The DSP processes the hydrophone signals and sends Time Difference of Arrival (TDOA) data to the main computer. The NI USB-7855R OEM was chosen for its 1Ms/s (simultaneous) sampling rate which prevents quantization error. It has an onboard FPGA which allows implementation of real-time digital narrow bandpass filters and cross correlation in hardware. The FPGA was programmed in National Instrument's LabView development environment.

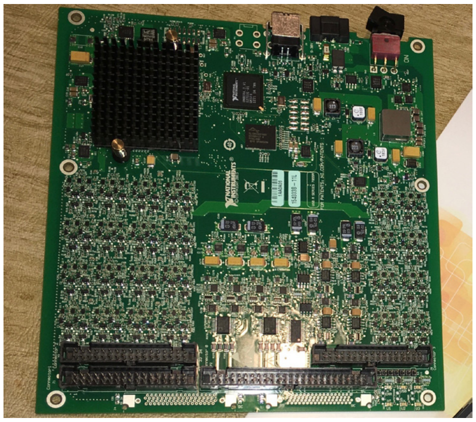

Fig. 12: Marlin's DSP/DAQ

IV. VEHICLE DESIGN: SOFTWARE

The software subdivision architects, implements, and maintains all programmable elements of the AUV. The subdivision strives to create a robust system with the flexibility and power to deal with challenging conditions and perform complex actions. To do this, the subdivision designs and improves algorithms to manage various aspects of the AUV's operation, including mission planning, navigation, vision, and signal processing.

Marlin's software stack runs on the Debian GNU / Linux operating system. It consists of a collection of separate processes that communicate over named pipes. This has the benefit of allowing easy testing of individual components, making each program smaller and easier to maintain, ensuring thread and memory safety (most programs aside from Interface can be single threaded), and preventing the entire system from breaking when a single component stops working. Most software is written in C/C++ and Python.

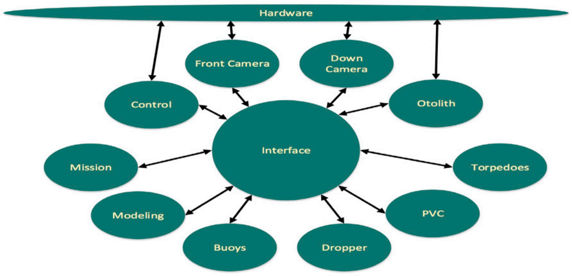

Fig. 13: Software Diagram

A. Interface

Interface is the master program that connects to each process. Each program can issue queries or commands to Interface through the designated pipes. Typically these involve requests for various objects (*e.g.*, images, the current state, and the prediction model) or messages to pass to other processes. Each connection is processed in a separate thread, so each component is plug-and-play; if a component is disconnected, Interface will just stop using that information; if it is reconnected, Interface will resume using that information.

B. Control

Our low-level controller runs on the ATmega2560 microcontroller. It connects to Interface, the AHRS, and the thruster controllers via serial UART pins. It utilizes six proportional-integral-derivative (PID) filters [5], one for each element of the state, to control the bearings of the submarine. The resulting PID vector is multiplied by the thruster matrix, mapping the effect of each thruster on each element of the state, to produce the desired thruster power. The state consists of the horizontal position, computed by integrating the thrust vectors, the depth, determined from the analog pressure sensor, and the attitude, received from the AHRS. Control sends the current state to Interface and receives the desired setpoint. It also activates the relays on command.

Because of the simple serial protocol, Marlin can be controlled manually through a terminal for testing and debugging use. Furthermore, all configuration variables (e.g., PID gains and the thruster matrix) can be adjusted over serial communication to empirically tune the optimal values on the fly. Most of Control, aside from a limited set of AVR specific sections such as input / output, is designed to be platform independent, so it can be tested and debugged offline on any laptop.

C. Modeling

In order to robustly navigate between objects in the pool and account for imprecise positioning and possibly unreliable observations, Modeling constructs a probabilistic map of all objects in the pool. Our custom designed modeling algorithm incorporates recursive Bayesian estimation to update its predictions as it receives new signals from the perception processes. The probability density map is computed as the sum of a set of hypotheses (Gaussian distributions) with varying means and variances. For each observation (given in the form of another Gaussian distribution), each hypothesis is split into two hypotheses, one accepting the observation and the other rejecting it, based on a given certainty. Gaussian distributions are combined by multiplying them, forming a new Gaussian distribution as described in [1]. In order to prevent exponential computational complexity as more observations are added, Modeling trims out unlikely hypotheses, where likelihood is proportional to the integral of the probability density function (PDF) across all dimensions.

This solution was arrived at after much research into alternative systems. It was found that a particle filter [3] would have to be either too slow or too inaccurate, and Kalman filter variants (e.g., Unscented Kalman Filter [4] and Extended Kalman Filter [6]) would not be able to account for bad observations without hard-coding an error rejection filter. The implementation of a multi-hypothesis model whereby each hypothesis is given by arbitrary PDFs was considered, but because of the inaccuracies involved in combining hypotheses and the high power requirements for computing the integral, instead Modeling enforces that all hypotheses are Gaussian, resulting in the solution described above, at the cost of slight approximation error.

D. Mission Control

Mission Control generates a list of goals (*e.g.*, ram the red buoy), each containing a list of primitive actions (*e.g.*, move to location). Each goal is associated with a point value and time to completion (TTC). At the start of a run, and after each goal is completed, Mission Control selects a new goal on the fly based on the point value, TTC, and distance (computed from the most likely hypothesis). This allows for varying task sequences in order to adapt to unforeseen circumstances.

E. Vision

Marlin has one front facing USB camera and one down facing ethernet camera. Each competition element is detected with a separate vision module for efficiency and parallelization. Images are processed by a combination of edge, contour, and blob detection, thresholding, and color mapping using the open source OpenCV library. Each module receives images from Interface and responds with a hypothesis (in the form of a Gaussian distribution) for the location of the object.

F. Otolith

Otolith, written using LabVIEW, runs on the FPGA and reads from the four hydrophones located outside the main hull. The analog-to-digital converter (ADC) receives amplified signals from the hydrophones at 500 thousand samples per second. Using each analog signal, Otolith applies a Butterworth low pass filter [2] to remove extraneous noise, then looks for a ping of the designated frequency. The time at which the ping is detected is recorded for each hydrophone, then the time difference of arrival (TDOA) is computed and sent to the main computer for localization. On the main computer, a small C program receives the TDOAs and computes the location of the pinger using trigonometric ratios between the TDOAs; this method, while introducing a slight built-in error (which decreases as the sub points towards the pinger), is less susceptible to slight errors in physical measurements and ping detection when compared to multilateration (solving the system of hyperboloids), making it much more robust. It sends to Interface the corresponding Gaussian distribution

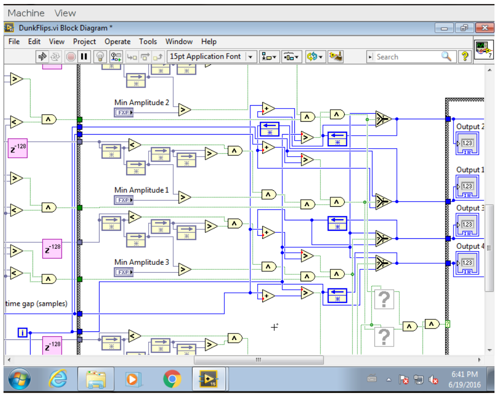

Fig. 14: Otolith

G. Logging

Because essentially all important information is sent over the pipes between processes, Marlin uses the standard "tee" command to copy messages to a file. Therefore, Marlin records comprehensive logs including the current state, the estimated model, images, and the results of vision processing.

I. Renderer

The renderer is designed in the free and open-source 3D-modeling software Blender. When it is given a state, it generates a front and down image from that state and sends them over the designated pipes. Combined with a simple physics engine, this serves as a simulator for the entire competition course. Because the simulator can be run on each development computer, this allows different team members to work on different tasks concurrently. In addition, the simulator increases development efficiency by detecting time-consuming bugs during dry testing.

V. RESULTS

Marlin has undergone comprehensive dry testing. The mechanical models underwent finite element analysis before machining, and stability and waterproofness have been verified. Every electrical subsystem has been checked for reliability and connectivity, and Marlin's software has successfully been able to communicate with each device, including sensors and motor control.

Furthermore, Marlin's software stack has been extensively tested offline to ensure correctness, safety, and robustness. Each process was run through valgrind, which reports zero definitely lost and very few possibly lost blocks, which appear to come from either valgrind itself, libc, or included libraries. Control was tested with a custom physics simulator to verify working PID. Each vision module has been tested on hundreds (or in some cases, thousands) of images to ensure accuracy; each module tends to get over 90% accuracy (*e.g.*, buoys gets around 98%). Furthermore, the software system as a whole has been tested with the renderer to simulate an entire competition run.

Marlin has undergone about 10 hours of in-water testing, primarily focused on tuning control systems. The higher level functions will be tested in the coming months.

VI. ACKNOWLEDGEMENTS

The team would like to thank its club advisor, Mrs. Bree Barnett Dreyfuss, for helping with administrative issues and allowing us to use the school pool. In addition AVBotz would like to thank its Platinum Sponsors: Datron Dynamics (Chris Hopkins and Adrian Montero), 3D Access (Jimmy Kim), Tanius Technology (Karl Schulze), National Instruments (Ingo Foldvari), Analog Devices, Andreessen Horowitz, Google, SubConn, eBay, and Silicon Valley Community Foundation. Without the support of sponsors, this project would not be possible.

Fig. 15: Volunteering at HPMS

VII. APPENDIX

AVBotz has continued to engage with the community. The team has demonstrated its vehicle at local street fairs and meet-ups. The team hopes to teach and encourage the local youth to pursue engineering. This year, several AVBotz members have volunteered teaching robotics and programming to middle school students on a weekly basis. Additionally, the club ran a series of workshops at ACE Code Day, an event at Amador Valley High School to promote coding to middle and high school students throughout the Bay Area.

VII. REFERENCES

[1] P. A. Bromiley. Products and Convolutions of Gaussian Probability Density Functions. *Tina Vision*. Manchester, UK. Rep. 2003-003. 2014. [Online]. Available: http://www.tina-vision.net/docs/memos/2003-003.pdf

[2] S. Butterworth. (1930). On the Theory of Filter Amplifiers. *Presented in Experimental Wireless and the Wireless Engineer.* UK. [Online]. Available: http://www2.ee.ufpe.br/codec/paper%20BUTTERWORTH.pdf

[3] A. Doucet and A. M. Johansen. (2008, December). A Tutorial on Particle Filtering and Smoothing: Fifteen years later. [Online]. Available: http://www.cs.ubc.ca/~arnaud/doucet_johansen_tutorialPF.pdf

[4] A. W. Eric and R. van der Merwe. (2000, October). The Unscented Kalman Filter for Nonlinear Estimation. Presented in *IEEE Adaptive Systems for Signal Processing, Comm., and Control Symposium.* [Online]. Available: https://www.seas.harvard.edu/courses/cs281/papers/unscented.pdf

[5] Principles of PID Control and Tuning. *Eurotherm*. [Online]. Available: http://www.eurotherm.com/temperature-control/principles-of-pid-control-and-tuning

[6] M. I. Ribeiro. (2004, February). Kalman and Extended Kalman Filters: Concept, Derivation and Properties. [Online]. Available: http://users.isr.ist.utl.pt/~mir/pub/kalman.pdf

[7] *Parker O-Ring Handbook.* ORD 5700. Parker Hannifin Corporation. Cleveland, Ohio. 2007.