Amador Valley Robotics Club: *Marlin* AUV Design 2017

Evan McNeil (P); Mechanical: Nicholas Maslennikov (VP), Ian Chu, Richard Ho, JJ Hu, Calvin Qin, Varun Srinivasan, Jonathan Ye, Shawn Zhang; Software: Varun Iyer (VP), Jeffery Chen, Timothy Kanarsky, Jeffery Li, Jeremy Li, Surya Ramesh, Justin Shih Charles Sun, Emil Tu, Andrew Wang, Daniel Yang, Jonathan Yang, David Zhang; Electrical: Shreyas Krishnaswamy (VP), WonJoon Lee, Wilmer Lin, Meher Nagpal, Lucas Relic, Sarah Tandean, Alan Thomas

***Abstract*—Amador Valley Robotics Club (AVBotz) is a student-run club at Amador Valley High School in Pleasanton, Califonia. The team consists of 25 students in three subdivisions—Mechanical, Electrical, and Software. Since 2000, the club has participated in Robosub, perpetuating a legacy of problem solving and independent learning.**

Last year, the AVBotz team retired the fourteen-year-old submarine Barracuda and created Marlin, a brand new autonomous underwater vehicle (AUV) from scratch. This year, the team focused on refining Marlin through rigorous testing and upgrades to its electrical system and software. This paper details Marlin's existing specifications and recent changes.

Marlin's rack houses two 16000 mAh batteries, which powers an Intel i7 CPU and eight Videoray M5 thrusters. Along with the main computer, Marlin uses an Atmega 2560 board to run a PID controller and handle low-level control of the submarine. The actuators on the submarine are the torpedo launcher, the dropper, and the grabber. These are controlled by five solenoids connected to the Atmega 2560. For detecting pingers, Marlin has four hydrophones mounted on opposite sides of the frame. The signals from these hydrophones pass through a high-pass filter before reaching Marlin's digital signal processor (DSP), which relays TDOA data to the main computer.

Marlin's software architecture is composed of various micro-processes that communicate through Unix pipes. This enables easier logging and debugging, as all messages exchanged between microprocesses are recorded. This also ensures that a single runtime error will not crash the entire program.

This year, the most notable improvements have been to Marlin's software. Optical flow, a micro-process that measures the velocity of the submarine using the down facing camera, and accurate vision processing of the bins task were appended to the software.

Fig. 1: Marlin overview specifications

Weight (in air)	36 kg
Hull	9.5 in (24 cm) diameter acrylic tube
Dimensions	Length: .95m Width: .86m Height: .56m
Propulsion	8x VideoRay MS Brushless Thrusters
Power	2x 22.2V 16,000 mAh LiPo Batteries (in series)
Underwater Connections	SubConn Power, Circular, Micro Circular, Ethernet, and Coax series
Cameras	2x 1.3 MP Point Grey Blackfly machine vision cameras w/ Theia Technologies SY125M lenses
Navigation Sensors	Pressure Sensor (Ashcroft Model K1) AHRS (PNI TRAX AHRS Module) Hydrophones (4x Teledyne Reson TC4013)
Main Computer	Intel i7-4790T on Jetway NG9J-Q87 Mini ITX 8GB DDR3 RAM 120GB mSATA SSD
Embedded Computer (Control)	ATmega 2560
Data Acquisition and Signal Processing	National Instruments USB-7855R OEM (RSeries Multifunction RIO with Kintex-7 70T FPGA)

I. DESIGN STRATEGY

Marlin serves as a platform for the next generation of AVBotz autonomous underwater vehicles (AUVs). Its design emphasizes reliability, longevity, and modularity for the benefit of future AVBotz members. The team considers the continued legacy of Marlin when making design decisions, leaving room to make upgrades and ensuring the submarine is well-documented.

Marlin's mechanical structure is robust, consisting of a framework of anodized aluminum panels, which can be singularly modified and replaced without having to remachine adjacent pieces. Additionally, the extra room on the horizontal plane of the frame allows for future modifications. Marlin's electrical infrastructure boasts extra connectors on the rear endcap, large electronic shelves, and a highly versatile computer motherboard, again ensuring plenty of space for upgrades. Marlin's software is also built to accommodate changes. It consists of a multitude of micro-processes that communicate through Unix pipes and can be easily changed. This means that any future devices added to Marlin can be easily integrated into the software through an interface program. All of these design choices ensure that Marlin can be improved in the future.

II. VEHICLE DESIGN: MECHANICAL

The mechanical subdivision is responsible for designing and constructing Marlin's structural components. The objective for this year was to identify shortcomings and develop solutions to improve the efficiency and capabilities of the submarine's current design. This task was made easier with the transposable structure configured in the previous year. The mechanical subdivision used many tools to complete this task, including SolidWorks CAD and Simulation, 3D printers, CNC machines, lathes, and drill presses.

A. Frame

Barracuda's frame was composed of a series of aluminum bars and twin High-Density Polyethelene (HDPE) side panels. The limited space hindered the placement of external components such as the grabber and torpedo launcher. The aluminum bars were also prone to corrosion, and the size and lack of any openings in the side panels made it difficult to see into the inner frame. The new frame is CNC'd from 6061-T6 aluminum, allowing for high strength with low weight. It is anodized with a type-II coating, giving it a smooth black/blue finish that protects it from oxidation and water corrosion. The hull mounts, surge thrusters, dropper, torpedoes, and many other components are mounted on the horizontal plane of the frame (See Figure 1). The two vertical side panels (see Figure 2) support the weight of the submarine and allow for the mounting of the vertical thrusters, grabbers, and hydrophones. The front and back panels of the frame help prevent warping of the horizontal plane under the weight of the hull and give it rigidity. The cutouts allow for weight reduction while maintaining structural integrity, and define areas for future additions.

Marlin was penalized points last year because of its weight. Modifying the frame would require a redesign of Marlin and re-analysis of the submarine's structural integrity, so instead, weight was reduced by modifying its other components. For example, the front-facing linear sonar was removed, as it did not contribute to Marlin's performance. The method of balancing the sub in the water was also changed. Instead of weighing down the lighter sections, the heavier sections were lightened with foam and the overly buoyant foam padding around the linear sonar was removed.

B. Electronics Rack

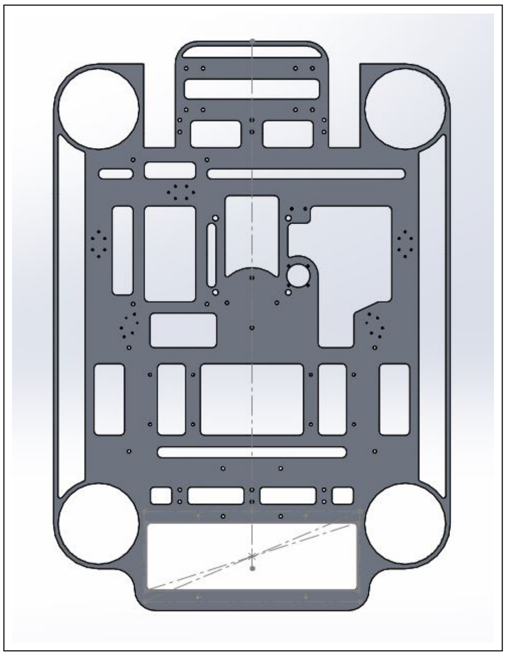

Fig. 2: Horizontal (pneumatic) plane

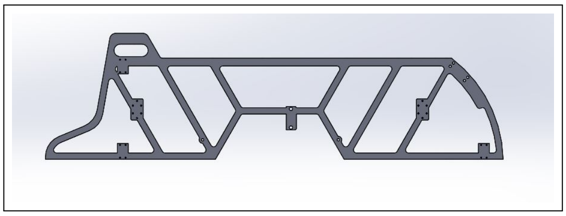

Fig. 3: Side panel

The electronics rack is machined from aluminum sheets, to which the submarine's internal electrical components are mounted securely. This allows Marlin to function in any orientation without endangering the rack's components. The previous rack design loosely attached to the removable endcap. This caused the rack to slide forward whenever the sub would tilt downward, disconnecting from the other electrical components. Now the electronics rack screws into the inside of the back endcap which also allows for a more secure structure (see Figure 3). The rack now also houses all the electrical components, enabling easier access. This also allows heat generated by the electrical components to transfer through the aluminum plates, into the endcap, and dissipate out into the water.

C. Hull

Marlin's hull consists of a 9.5-inch diameter acrylic tube that is 25.875 inches in length and sealed by two anodized 6061-T6 aluminum endcaps (see Figure 4). The front endcap is attached to the front of the hull with 3M DP420 epoxy while the front camera dome is sealed using an o-ring. However, the back endcap is attached to the end of the hull with an aluminum collar to ensure seal integrity. By design, the back endcap makes contact with aluminum rather than casted acrylic. In addition to a 10-degree chamfer,

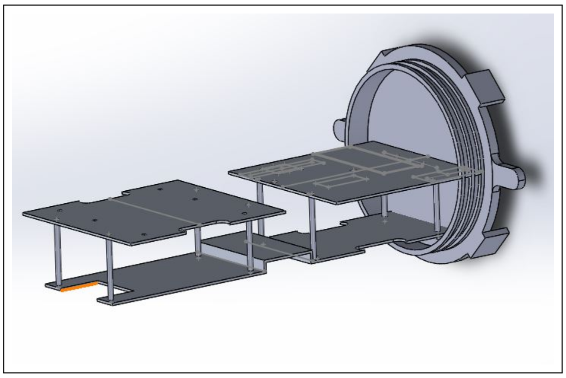

Fig. 4: Rack and back endcap

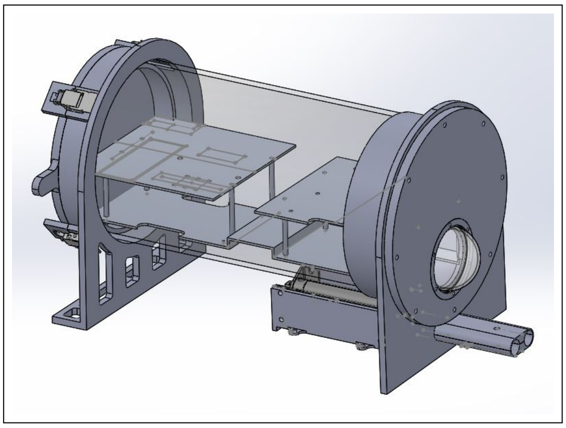

Fig. 5: Hull with aluminum collars

Marlin's hull uses o-rings that are 3% smaller than normal specification. This creates a balance between a watertight seal and ease of insertion or removal, unlike in Barracuda, in which the size of the o-ring caused great difficulty in moving the rack. Four spring-draw latches on the rear of the hull help keep the back endcap seal watertight by ensuring a solid contact between the endcap and the aluminum collar, which separates the endcap and the hull. Attached to the endcap are 13 waterproof SubConn connectors that allow for interface between Marlin's internal and external systems, 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.

D. Thrusters

Marlin's propulsion system consists of eight VideoRay M5 thrusters, which have high thrust and power efficiency. Four vertical thrusters provide depth, pitch, and roll, while four horizontal thrusters are arranged at 45-degree angles and control strafe, yaw, and surge movements, for a total of six degrees of freedom. This configuration increases front camera field-of-view and movement speed compared to the older AUV Barracuda model. In the old submarine model, a thruster mounted in the front prevented the forward camera from having an unobstructed image. Marlin's new design leaves room for any future integration of a sonar and Doppler Velocity Log (DVL).

E. Pneumatics

Marlin is intended to accomplish all mechanical tasks through pneumatic actuation. Throughout Marlin there are five Numatics 7/16 x 4 inch actuators that drive its pneumatic systems. Marlin's solenoids are housed inside an aluminum box with water-tight cord-grips (see Figure 6), while its air canister and pressure regulator are connected to the soldenoids via a tube.

F. Dropper

The basic design of Marlin's dropper has been carried over from the team's earlier submarine design, the Barracuda. It is an external attachment consisting of two primary components: a container and a piston. The container consists of three open tubes: one horizontal and two vertical, with one sticking out from the top and one from the bottom. The piston has a hole in which a 1-inch diameter stainless steel ball bearing is held. It is pushed by a pneumatic actuator and slides in and out of the container's horizontal tube. Two ball bearings are loaded into the top tube of the container, the first one falling into the piston hole. The piston is pushed forward, taking the ball bearing over the bottom tube, causing it to fall out. The piston goes back to its initial position, and the second ball bearing falls into the piston, allowing the process to repeat. Barracuda's dropper was created with a series of acrylic plates glued together. This resulted in the tubes being rectangular, and so the ball bearing's fall to the bins was not as accurate as it could be. By 3D printing it with polylactide (PLA) the tubes could be made round, allowing the ball bearings to fall out in a straighter path. Also, the one-piece plastic dropper has a better aesthetic than the obviously glued acrylic plates.

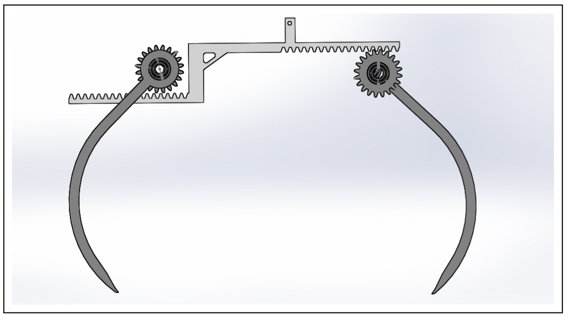

Fig. 6: Grabber halfway closed

G. Grabber

The grabber design revolves around a pneumatic actuator that moves a central piece. The movement of the central part causes the arms to open and close. The previous year, the mechanical subdivision learned that the initial grabber design was very unstable and the incapability of the meshing of the gears limited the grabber from performing at all during competition. Initially, brainstorming new designs led to the creation of a scissor grabber concept. However, multiple problems arose from the new concept: the gripping region was continually changing, thus compounding more problems for the software subdivision; a new mounting system was required to fit the new arms; and structural revisions to other sections of the submarine reduced the space needed for the design to operate. Thus, the mechanical subdivision sought an easier method to address previous problems in the grabber by modifying the existing design. By adding extensions on the central meshing piece and the arm segments, the structure has less horizontal and vertical give, resulting in greater overall stability (see Figures 6 and 7).

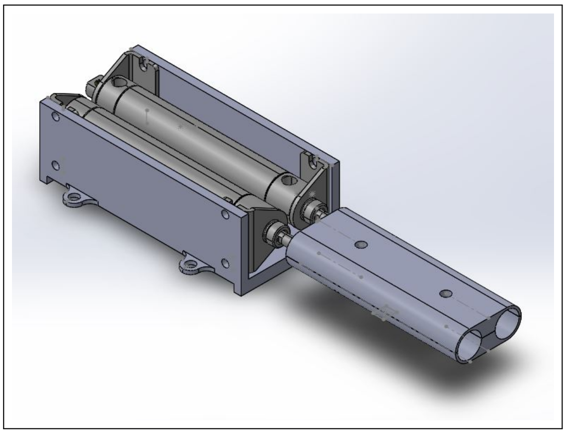

Fig. 7: Torpedo launcher

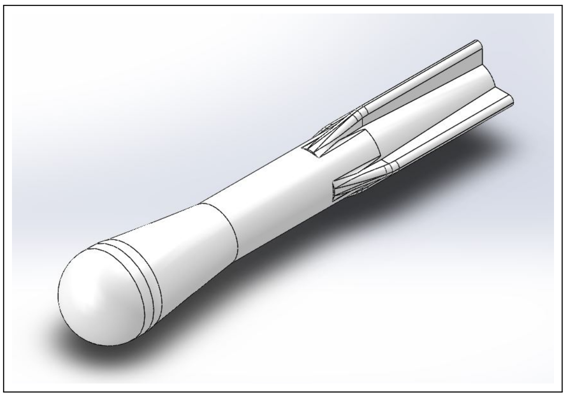

Fig. 8: Torpedo

H. Torpedo Launcher

Marlin is the first AVBotz submarine to have a torpedo launcher. The torpedoes operate using pneumatic actuators that extend pistons which in turn propel the torpedoes (see Figure 8). The torpedos are housed in a 3D-printed PLA tube complete with o-rings that secure them in place. There is no reloading mechanism within the submarine so for new salvos to be fired, the tubes must be reloaded manually. The torpedoes themselves are designed to be balanced and neutrally buoyant to ensure they remain straight as they move through the water (see Figure 9).

I. Kill Switch

Marlin's kill switch is a Carling Technologies sealed switch which kills power to the thrusters while underwater. The wiring is located within 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. This allows for a manual option to cease the thrusters in case of an anomaly while Marlin is being tested.

III. VEHICLE DESIGN: ELECTRICAL

The electrical subdivision is responsible for all of Marlin's internal and external electrical components. This year the subdivision focused on improving and perfecting the existing system with the primary goal being reliability. This strategy is a sharp deviation from last year's which was essentially adding as many different capabilities to the sub as possible in the hopes of giving the software subdivision ample tools to guide the sub through all tasks. Unfortunately, during last year's competition, the sub's performance was hindered by many of the unrefined or entirely unused elements' performance. For instance, a linear sonar intended to simplify the torpedo task went entirely unused due to its inaccuracy. This only added weight to the sub, and thus was eliminated this year. The electrical subdivision's new priority of precision over expansion resulted in a smaller but more reliable electrical system (refer to figure 11 for the broad overview of this system). One example of this is the pneumatics system. Recognizing the inconsistency and fragility of last year's system, which resulted in a flooded pneumatics connector/box, this year the subdivision designed a completely new pneumatics system. While maintaining the advantages of last year's design, and improving it by reducing the size and strengthening the waterproof seal, the electrical subdivision ensured that the system did not experience the same issues as last year. Overall, by making similar changes including a stronger ethernet cable to reduce daily wear and tear when working with the tether and also the removal of the linear sonar, the electrical subdivision succeeded in establishing a more reliable and consistent infrastructure. With the remaining time, the subdivision bug-tested and organized the existing rack to reduce the number of spontaneous malfunctions that occur during a run. By trading potential capabilities for consistent reliability, the electrical subdivision enhanced the performance of Marlin's essential functions.

A. Batteries

Two Venom 22.2V 16,000mAh 6S Drone professional lithium polymer batteries power Marlin. Connected in series, they nominally generate 44.4V with an energy capacity of 710.4Wh. With a full charge,

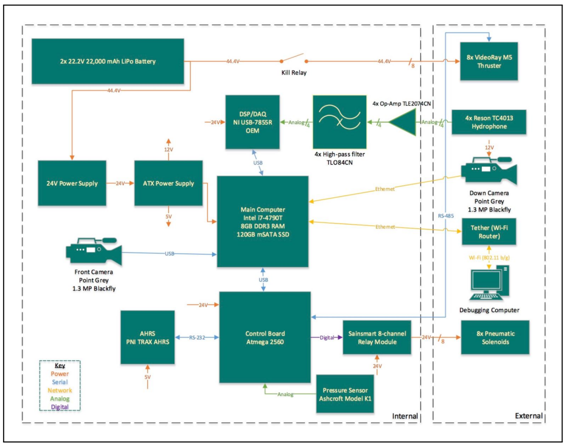

Fig. 9: Overview of Marlin's electrical system

Marlin can operate for about two to three hours. This represents a 1 hour decline in total operational charge from last year since the new batteries this year contain an energy capacity of 16,000mAh instead of 22,000 mAh. This downgrade in battery life is a result of Venom discontinuing the batteries used last year. The new batteries were selected for their compatible voltage and cell count, allowing them to effectively replace the old ones. Additionally, their smaller size frees more space for wires to pass through the rack, allowing for better organization.

B. *Power Supply*

The thrusters are powered by 44.4V, which comes directly from the two lithium polymer batteries. All the other components of Marlin run off of either 24V, 12V, 5V, or 3.3V. In order to meet these requirements, Marlin's two tier system steps down the raw battery voltage. The first tier utilizes a Cincon DC/DC converter to step down the battery voltage from 44.4V to 24V. Afterwards, the second tier takes the 24V output from the first tier and uses an ATX power supply to separately power the computer and also reduce the newly stepped-down voltage to 12V, 5V, and 3.3V that are then sent through power rails that bring current to Marlin's other electrical components. The power supply system was designed with these four different power rails in mind, not only because they can power current components, but because these voltages are also common in over-the-counter sensors and electronics. This means that Marlin is capable of powering many non-custom electronics without changing the power supply system.

C. *Power Distribution*

To guard Marlin from electrical malfunction or unexpected behavior, power is distributed through a series of fuses and switches. Thruster power is sent through the thruster relay, an Omron G9EAB, which is also connected to an external kill switch. Once the switch is activated, power to the Omron is cut, and the thrusters stop. A 100 amp fuse connected directly to the batteries adds further protection from spontaneous power surges by the thrusters. To contain other malfunctions on the inner rack, individual fuses are connected to the 24V, 12V, and 5V power rails, preventing electrical surges from propagating through the entire submarine. Refer to figure 11 to view the Omron relay and the 100A fuse.

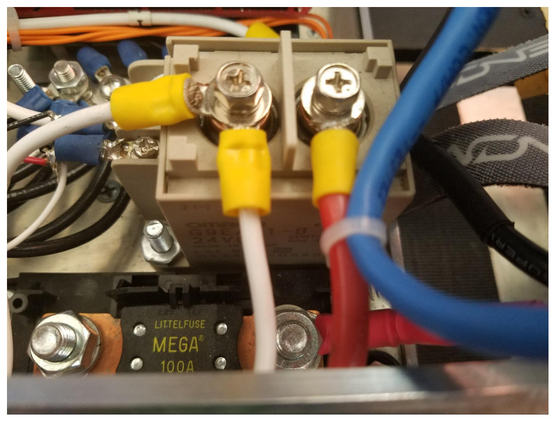

Fig. 10: Omron G9EAB kill relay

D. *Bulkhead Connectors*

Marlin uses MacArtney SubConn underwater connectors to interface with all external devices (i.e. the thrusters, the tether, the downwards-facing camera, the pneumatics system, and the hydrophones). The SubConn connectors were chosen due to their reliability and versatility throughout testing and designing. By virtue of having extra unused pins on each connector, a single connector can establish multiple different junctions between different external and internal parts. For instance, one of the connectors serves as a junction for both an FPGA debugging cable and the now defunct linear sonar. With this flexibility, the innovation process this year was easier since there was no need for major design strategy changes. In addition, the various types of connectors allow transmission of both power and I/O signals.

E. *Cable Splicing*

Internally, cable splices are protected with heat shrinks. However, external splices must not only insulate the conductor but also maintain a waterproof barrier. To solve this problem, the 3M Scotchcast inline resin kit and Polycade potting boxes filled with MG Chemicals 832C compound are used to create neat and compact cable splices. Outside the hull, on the horizontal plane, are two potting boxes, one that contains the cables that merge thruster connections and another with hydrophone connections. Additionally, the thrusters' potting box was used to splice power and RS-485 lines. Lastly, to save space and simplify cabling, both thruster power and thruster control connections were stacked on the horizontal plane using Teledyne Impulse connectors.

F. Main Computer

The Jetway NF9J-Q87 Mini-ITX motherboard was chosen for the basis of Marlin's main computer. With a 2.7 GHz quad core Intel i7-4790T Haswell processor, 8GB of RAM, and a 120GB mSATA SSD, the main computer manages high-level functions such as image processing effectively. The NF9J-Q87 was chosen for its large number of USB ports and its onboard RS-232 ports, allowing easy access to the plethora of sensors across the submarine. Large numbers of common interface slots allow for convenient replacement and addition of electrical parts. 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. A heat pipe CPU cooler handles the heat generated by the main computer.

G. Control Board

Low level tasks, such as navigation and motor control, are delegated to Marlin's control board, an ATmega 2560. The ATmega is Marlin's 8-bit AVR RISC-based microcontroller that runs at 16MHz, has 256KB of flash memory, and interfaces with Marlin's sensors through custom serial converters and kill state sense circuits. It was primarily chosen for its simple programming environment and numerous I/O lines, ultimately conserving time and preserving freedom to add or replace connections to the board. This freedom has opened up the possibility of adding an external configuration switch, a specialized hall sensor, or extra solenoids without expanding the ATmega board. This expandability is enhanced through the Protoshield (the red PCB attached above the ATmega board in Figure 12), which increases the complexity of the circuits that can be built into the control board.

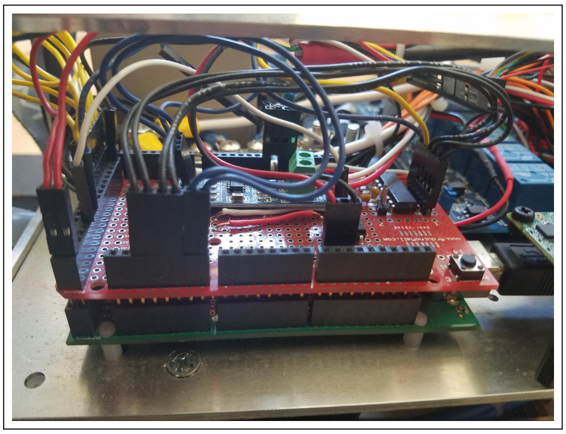

Fig. 11: ATmega 2560 with Protoshield attachment

H. Motor Control

The ATmega 2560 controls Marlin's thrusters over a bussed RS-485 interface, which enables a simplistic software setup. The thrusters are further controlled by the Omron kill relay. Connected to the external kill switch, the relay blocks power to the motors once disabled by the switch. To prevent heat buildup, the motor's electronic speed controller is housed inside the thrusters themselves, allowing heat to dissipate to the surrounding water.

I. WiFi Tether

Marlin uses the same Netgear router and rechargeable battery pack in a waterproof pelican case as previous models. Recognizing a significant problem in the previous design however, the ethernet cable was upgraded to an IP67 rated 300 foot shielded Cat6 ethernet cable. The shielded cable provides protection against electromagnetic interference of the signals that pass through the tether. A long and durable ethernet cable allowed for efficient changes in software settings without stopping water tests and forcing Marlin back to land. The ethernet cable is spliced to a Subconn ethernet cable, allowing it to be connected with the internal computer via the external subconn ethernet connector (see Figure 13).

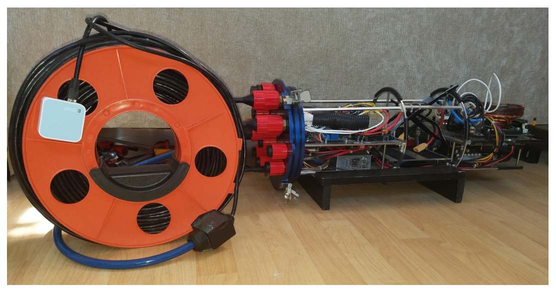

Fig. 12: Rack with tether attached

J. Navigation Sensors

To maintain accurate directional and depth readings, Marlin is equipped with both a depth sensor and an altitude and heading reference system (AHRS). Marlin uses PNI's TRAX AHRS, composed primarily of three 3-axis sensors: a magnetometer, an accelerometer, and a gyroscope. The TRAX itself processes raw data with a Kalman filtering algorithm and sends the data to the ATmega 2560 over RS-232. The TRAX also filters out signal interferences caused by erratic motion or changes in the local magnetic field. It is important to note that the AHRS replaces the IMU used in previous submarines because of its better sensors and onboard processing power. Marlin also contains an Ashcroft Model K1 Pressure Transducer that transforms pressure readings into voltage signals and then linearly converts these signals into depth readings on the ATmega 2560.

K. Cameras

Two 1.3 MP Point Grey Blackfly machine vision cameras capture accurate underwater pictures of the course. The field of view of these cameras is expanded to 125° with little distortion, increasing the amount of visual information available to Marlin. To conveniently take pictures of both incoming objects (such as buoys) and objects below the sub (such as bins), the front-facing camera is housed at the front of the sub in the rack while the downwards-facing one is housed in an external acrylic enclosure that connects to the rack through a waterproof SubConn cable. Captured images are then sent to the main computer via ethernet (from the down-facing camera) and USB (from the front-facing camera).

L. Actuator Control

An 8-channel SainSmart relay module controls the solenoids based on commands from the ATmega (see Figure 14). Controlled through digital output pins on the control board, these relays pass 24V of power to the solenoids, turning them on and off. Chosen for its affordability and reliable performance, the SainSmart module also has three unused relay channels, opening the possibility for future expansion.

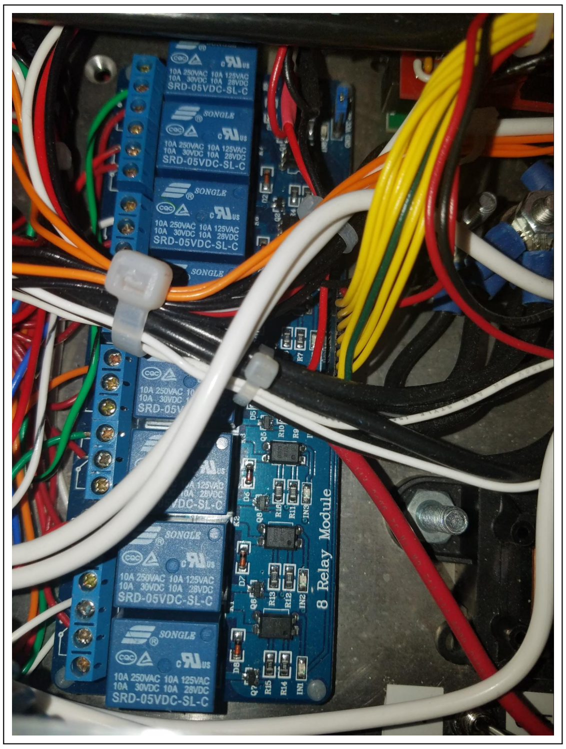

Fig. 13: 8-channel SainSmart relay module

M. Signal Processing

Marlin's four Teledyne Reson TC4013 Hydrophones capture audio from the pingers and transduce the audio inputs into electrical signals. The electrical outputs are passed to Marlin's four-channel custom amplifier, which is built on the Texas Instruments TLE2074CN operational amplifier (op-amp) and has a gain of 40dB. The newly amplified signals are then filtered by a high-pass filter that is designed around the TLO84CN, producing a cutoff of 10kHz. This initial signal conditioning takes place inside a shielded box located on the rack to minimize interference. The op-amps on the custom amplifier/filter board are powered by two 9V batteries that are housed on either side of the shielded box. After conditioning, the four analog signals are then processed by Marlin's digital signal processor (DSP), the National Instruments USB-7855R OEM, whose 1Ms/s sampling rate significantly mitigates quantization error. Furthermore, the simultaneous sampling ability of the DSP makes it capable of processing incoming data from all four hydrophones efficiently. This, coupled with the high sampling rate, made it an ideal choice for Marlin's DSP. The DSP shares the National Instrument board with a Kintex 7 FPGA that further processes the signal through real-time digital narrow bandpass filters and cross-correlation algorithms. The DSP board then sends the finalized Time Difference of Arrival (TDOA) data to the main computer.

IV. VEHICLE DESIGN: SOFTWARE

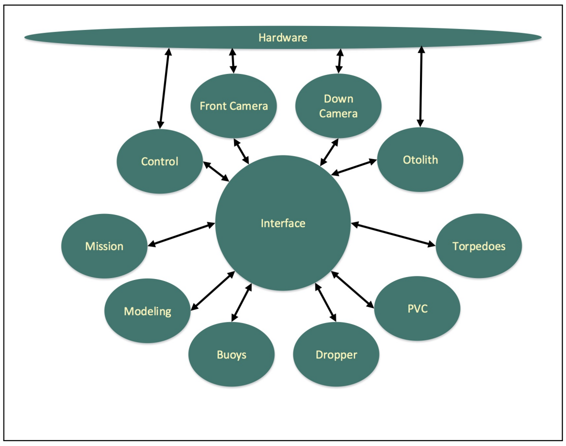

Fig. 14: Software Organization

The software subdivision is responsible for writing software to detect and complete the competition tasks efficiently. Last year the software subdivision focused on creating Marlin's base software and less emphasis was placed on improving vision processes and making Marlin's movement more calculated. However, this year the subdivision's focus was on improving those weaknesses.

A. C-Exec

Marlin runs on a microprocess architecture. Each subdivision of labor (different vision algorithms, decision making, environment modelling, etc.) is done by a separate process in the Linux kernel. Previously, each process was called through a BASH script and inter-process communication (IPC) was implemented using named pipes. Each process is now executed from within our 'handler' program, *interface*. Microprocesses now communicate using Linux two-way sockets. The greater flexibility of C over BASH allows us to have more intelligent error handling and better logging. Previously, processes had to be constantly running, waiting for input from the handler program. Now, the handler can use them and kill them as necessary. Additionally, if any process were to fail, the error can be intelligently handled, dependent processes killed, and Marlin as a whole restarted. Previously, a process failure meant that all of Marlin would fail. Every program was linked through named pipes, and if a program on either end of a named pipe fails, the program on the other end of the pipe also fails. This would cause a domino effect that would bring down all of the submarine's systems. The new process handling program avoids this significant liability to Marlin.

B. Control

Control is Marlin's low level controller which handles movement, sensor data, and actuators. It operates on the ATmega2560 microcontroller and connects to the AHRS, the thruster controllers, and interface through serial UART pins. Using the sensor data, the submarine is able to determine its state, which includes depth, derived from the analog pressure sensor; attitude, acquired from the AHRS; and horizontal position, calculated by integrating the thrust vectors. Control receives a desired state from interface and sends the current state. Control uses six proportional-integral-derivative (PID) filters [6] to control the state of the sub. The desired thruster power is produced by multiplying the thruster matrix by the PID vector, which maps the outcome of each thruster on each component of the state. Marlin can be controlled directly through a terminal due to its simple serial protocol. In addition, all configuration variables (e.g., thruster matrix) can be adjusted over serial communication to set optimal values on the fly. A majority of control is created to be platform independent, so it can be tested on any laptop.

C. Modeling

To move around and accomplish its tasks, Marlin must first be able to determine where its tasks are and as well as its own position in its pool. As a result, Modeling uses a probability distribution to represent the certainty of its hypotheses for these positions. The modeling algorithm assigns a Gaussian distribution for the certainty that the objects in the pool are in each position. Then, it uses recursive Bayesian estimation to update its prediction on the placement of objects as new data is received from the sensors. For each observation (or Gaussian distribution of the position of an object from the sensors), two hypotheses are made rejecting and accepting the new observation. The Gaussian distributions from the hypotheses are then multiplied to the original probability map, forming a new Gaussian distribution. Since there are too many observations to process at once, modeling prunes some unreasonable hypotheses with a criterion of the integral of the probability density function. This solution was found to be the better solution compared to particle filters, which would be to either too slow or too inaccurate, and Kalman filter variants (e.g., Unscented Kalman Filter and Extended Kalman Filter), which would not be able to account for bad observations without an inaccurate, hard-coded error rejection filter. The implementation of a model using several different PDFs was considered, but this would be inaccurate because the integral would have to be calculated to update the hypotheses. Instead Modeling makes all hypotheses Gaussian, resulting in a easy, constant-time method for combining distributions, at the cost of slight approximation error. This year, the method of determining the variances of the Gaussian distributions for each observation was improved. Previously, the variances were hard-coded and did not account for the position of the submarine for the uncertainty of a hypothesis. A more accurate probability distribution based on the distance to an object created more accurate hypotheses, making the overall modeling more efficient.

D. Mission Control

Mission Control creates a list of goals that correlate to the competition tasks. Each goal contains a set of basic instructions, a point value, and a time to completion (TTC). Based on the point value, TTC, and distance to every goal, Mission Control dynamically selects a new goal after the completion of a goal or the start of a run. By allowing varying task sequences, Mission Control is able to adapt to unexpected circumstances throughout the run.

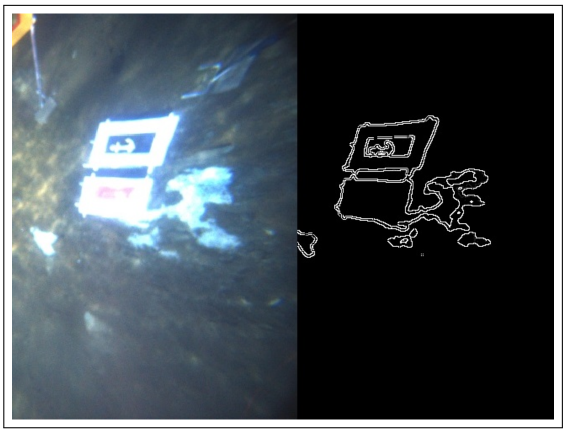

Fig. 15: Vision processing for the bins

E. Vision

Each competition task is detected with a separate vision process. The vision processes use a variation of machine learning, color mapping, thresholding, and edge, contour, and blob detection using the open source OpenCV library. Interface sends each vision process images and the processes respond with a hypothesis for the location of their task. Many vision tasks were rewritten this year to improve both accuracy and efficiency. An emphasis was placed on incorporating machine learning algorithms to allow for the detection of objects in a variety of harsh conditions, as shown in figure 16.

F. Otolith

To solve the underwater pinger task, the submarine uses four hydrophones, three of which are on the same plane. Marlin contains a Field-Programmable Gate Array (FPGA) to receive incoming signals from the hydrophones and determine the time of arrival of the pinger signals. Last year, the FPGA was programmed with LabVIEW, which is a graphical programming language. Unfortunately, the software had several problems, including long compile times and an unintuitive user interface. As a result, the hydrophone code was rewritten with MATLAB and Simulink, with Mathwork's HDL Coder and Xilinx's Vivado Design Suite used to download the bitstream onto the FPGA.

The program starts by sampling time-based data at a frequency of 500 kHz through the hydrophones. Since the pinger's frequency has an infimum of 25 kHz and a supremum of 40 kHz, a simple bandpass filter removes unnecessary background noise from the signal. Then, a fast Fourier Transform converts the time data into frequency data. Finally, the program returns the times when the signal contains a frequency between 25 and 40 kHz.

After the times of arrival are computed by the FPGA, the calculated time difference of arrivals (TDOAs) determine the location of the pinger through a multilateration algorithm. The three independent TDOAs narrow down loci of possible pinger locations to a maximum of two points.

G. Optical Flow

The high cost of a DVL, nearly $20,000, prevented adding one to the AUV design to accurately measure Marlin's velocity in the water. It was decided Marlin's existing accelerometer data would be augmented with the more cost effective method of measuring velocity: optical flow. Marlin's down facing camera is used to detect the position of key points on the floor of the pool, then the change in position of these key points is recorded with the Lucas-Kanade optical flow function. Thus the magnitude and direction of change between frames is measured and, given the change in time between these frames, velocity is determined.

One challenge of using optical flow was determining distance from the floor of the pool. After some consideration, it was decided to use the pressure sensor as the ground truth and assume that the depth of the pool is constant. Although this may seem like a fallacious assumption, the floor of the pool slopes at a constant rate, which makes the integral of measured velocity the same between two points in the pool, independent of path. During the first couple of runs in the pool optical flow data will be gathered, after which locations of objects in the pool can be set using units of measurement based on optical flow.

Since Marlin stores the images taken by both the front and down cameras periodically and labels them with the time and date, the submarine can access reference images from earlier in the run if it backtracks. This means that optical flow is not only an effective method of determining the velocity of the submarine, but also serves as a reference point to correct any drift that is accumulated from integrating velocity data.

V. EXPERIMENTAL RESULTS

This year, the AVBotz software subdivision was able to create a set of utility tools that allowed for the collection of more labeled underwater image data than ever before (over 5,000 images of buoys). Because of this, data-driven approaches to vision could be investigated. Several different kinds of object localization systems were tested, including the You Only Look Once (YOLO) Object Detection Framework[1], pretrained sliding windows classifiers[4], and early exiting optimizations[2].

A. YOLO

YOLO is a real-time object detection framework that directly regresses bounding boxes for objects. A 7×7 grid is drawn over an image, class probabilities are estimated for each square on the grid, and each grid attempts to regress a bounding box for any object centroids it contains. Experiments with YOLO were brief because of its long training times and unnecessary complexity for buoys, which has relatively simple bounding boxes (as opposed to the more complex irregular shapes found in real-world environments). Furthermore, YOLO depends upon the glut of training data provided by academic machine vision tasks such as the PASCAL VOC challenge, and initial results were unsatisfactory, perhaps as a result of the constraint of comparatively few bounding-box training examples. Other detection systems only required location data, which was also significantly easier to collect in large quantities than the bounding box data that YOLO required. Additionally, YOLO's machine learning infrastructure, darknet, relies heavily upon CUDA and cudNN. Since Marlin does not have a graphics card, it ended up being significantly slower than realtime. It was decided that the issues with YOLO and the investment of a graphics card installation were not worth the possible returns, and the avenue of inquiry was ended.

B. Sliding Windows

A significantly longer time was spent on developing an accurate sliding windows classifier for buoys. A sliding windows classifier relies upon a Convolutional Neural Network (CNN) to determine if a small portion of an image (i.e. 50×50 in a 600×400 image) contains a buoy or not. The image is then split up into many such small windows and fed through the CNN one by one, and any tile that is classified as 'yes' marks the location of a buoy. Berkeley's Caffe framework with the AlexNet CNN was used to classify the data samples. High ($\sim 99\%$) classification rates were acquired with little modification using AlexNet by taking advantage of the concept of transfer learning. AlexNet can be initialized pre-trained on ImageNet, an academic image classification task consisting of millions of samples and 1000 classes. Since the classes on ImageNet are extremely diverse, one can retrain CNNs trained on ImageNet for other tasks as well[5]. There were two primary difficulties with this approach. First of all, in an image, there can be hundreds of windows, so a 99% classification rate could mean several errors per frame in object detection. Second of all, speed is often an issue, as processing hundreds of windows takes a non-trivial amount of processor time, making the solution too slow to be effective. To address the first issue, more specific data was required. A single image can produce upwards of hundreds of negative training examples (windows containing no buoy). Previously, a few windows were chosen at random from each training image. After training the first version of the classifier, it was run on the training images to generate false positives. Each window falsely classified as a buoy was added to the training set along with some more randomly chosen examples to prevent overfitting. To address the second issue, the concept of early-exiting was explored. Early exiting allows CNNs to run with dramatically reduced runtimes by quickly discarding windows that have relatively easier examples without spending full computational resources on them. A fully-connected softmax classifier is trained at every layer of the net instead of just the last one, and if the classifier is over a certain threshold, it can classify the image at an early stage in the neural network and quickly continue. The solutions to both these problems are still being researched, and trials will continue.

VI. Appendix: Community Outreach

As always, AVBotz tries to attend as many local and school events as possible, such as street fairs and hackathons. In particular, this year's ACE Code day hosted several classes taught by AVBotz members. One such course was on control theory and walked through creating a PID controller. Another course was on the basics of machine learning and taught the different approaches and their implementations.

Multiple members of AVBotz have also volunteered at the local middle school's robotics club throughout the year to inspire the minds of young prospective engineers. At this club, robotics members helped teach basic programming skills along with basic engineering principles to guide students as they built their own robots.

Acknowledgments

The AVBotz team would like to thank its club advisor, Mrs. Bree Barnett Dreyfuss, for providing invaluable help with the school administration and allowing us to use Amador Valley High School's pool. In addition, the club would like to thank this year's sponsors: Datron Dynamics, 3D Access, Xilinx, and Tanius Technology (founded by an alumnus of the club). Without the contributions of these sponsors along with our sponsors from previous years, AVBotz would not have been able to build and refine Marlin.

References

[1] Redmon, Joseph and Ali Farhadi. *YOLO9000: Better, Faster, Stronger*. arXiv preprint arXiv:1612.08242, 2016.

[2] Surat Teerapittayanon, et al. *BranchyNet: Fast Inference via Early Exiting from Deep Neural Networks*. http://www.eecs.harvard.edu/ htk/publication/2016-icpr-teerapittayanon-mcdanel-kung.pdf, 2016.

[3] Jia Yangqing, et al. *Caffe: Convolutional Architecture for Fast Feature Embedding*. arXiv preprint arXiv:1408.5093, 2014.

[4] Ng, Andrew. *Machine Learning*. coursera.org/learn/machine-learning/. Coursera. 2017.

[5] Jeff Donahue, et al. *DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition*. arXiv preprint arXiv:1408.5093, 2013. Coursera. 2017.

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

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

[8] A. W. Eric and R. van der Merwe. *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, 2000

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

[10] National Instruments. *SPECIFICATIONS NI USB-7855R OEM R Series for USB Multifunction RIO with Kintex-7 70T FPGA*. [Online]. Available: http://www.ni.com/pdf/manuals/374241a.pdf