University of Southern California SeaBee Autonomous Underwater Vehicle: Design and Implementation

Leah Gum, *Captain,*
Dylan Foster, *Leader, Electrical Team,*
Edward Kaszubski, *Leader, Software Team,*
and Michael Benzimra, *Leader, Mechanical Team*

***Abstract*—The SeaBee is an autonomous underwater vehicle (AUV) developed by a team of students at the University of Southern California for the 15th International RoboSub Competition. The 2012 SeaBee AUV was entirely designed by the University of Southern California Robotics Society (USCRS) and hosts an array of new features integrated into a robust returning system. New improvements such as an external frame with increased configurability, revised grabber, shooter, and dropper designs, a passive sonar array, and an advanced dynamic software architecture offer greater capability while also increasing the modularity of the SeaBee AUV for additional revision in the future.**

I. Introduction

The University of Southern California Robotics Society (USCRS) focuses on designing and building an autonomous vehicle for the annual AUVSI and ONR RoboSub Competition as part of its main objective to participate in and further robotics research not only at USC, but also around the world. USCRS uses a two-year full design cycle between iterations of the SeaBee AUV, allowing time for thorough testing and integration of new designs. From preliminary internal design reviews to a critical review attended by experts in the field, designs and changes to SeaBee III undergo a thorough analysis culminating in vital improvements to the AUV without wasteful or unnecessary expense. This year USCRS is mid-way through the SeaBee IV design cycle and the current SeaBee AUV features the SeaBee III platform with new mechanical, electrical, and software improvements that increase not only task-centered capability but also focus on modularity and reuse.

II. Mechanical Design

A. Overview

The mechanical design philosophy for the SeaBee AUV is intended to create a modular, compact, and lightweight AUV. The single-hull design with an internal rack system enables easy removal and centralized access to electronics while the external frame provides support for long term development by allowing individual component redesigns with little to no effect on the overall structure of the AUV. Electrical connections are established by using waterproof connectors, supplied from Fischer Connectors, from the hull end cap to external components such as the IMU, SONAR, marker dropper, shooter, cameras, and thrusters allowing simplified electrical reconfiguration.

B. Hull

The hull for SeaBee is constructed out of 3/16" thick T6061 anodized aluminum. The enclosure employs a cylindrical design measuring 7.5" in diameter and 13" in length. It shields the internal electrical systems from water damage with a watertight design. In the hull cap design, the cap is covered with waterproof Fischer Connectors allowing access to the electrical systems from components outside of the hull.

C. External Frame

The new external frame is part of the design for the next iteration of the SeaBee AUV and consists of an octagonal cage surrounding the main hull. Each Octagon measures 10.836" around an inscribed circle. Four Octagons create three sections of the frame. Panels constructed of T6061 anodized aluminum measuring 7" x 4.5" x 3/16" make up the walls of the octagonal cage resulting in a total of twenty-four panels. Each panel is associated with a certain location of the frame and a component. In addition, panels can move around the frame easily if design changes are necessary. The front and back surfaces of the octagonal cage serve as mounts for the depth thrusters and the flotation structure.

These panels are designed around a common panel template in SolidWorks and can be quickly machined through a laser-cutting process and then anodized. The panel redesigns dramatically reduce the design and manufacturing time of any mechanical changes to the SeaBee AUV or its components and are compatible with either the SeaBee III or upcoming SeaBee IV designs.

The flotation is also built into the external frame. Due to the higher density of aluminum, flotation is a necessary addition to the sub. Four 3" diameter acrylic air canisters are placed at the corners of the AUV. Mounted alongside them are smaller acrylic tubes that can be loaded with weights. The weight tubes can be manually adjusted to achieve neutral buoyancy. With the mounted flotation system, SeaBee occupies a space of 28" x 26" x 20".

D. Internal Frame

SeaBee utilizes a custom designed internal frame attached to the main hull end cap, which can slide in and out of the hull on Teflon rails. This frame is machined from aluminum and is designed around the central electronics, including the carrier board, power board, and batteries ensuring a secure mounting platform for every device that is installed in SeaBee. 3D CAD drawings are used to guarantee that the electrical boards are compatible with the Internal Frame. The frame maximizes and efficiently uses space within the hull and provides a cooling infrastructure for the main computer (see Fig. 1).

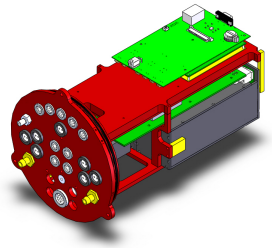

Fig. 1: Rendering of the SeaBee internal frame.

E. Liquid Cooling

SeaBee takes advantage of a unique liquid-cooling system consisting of custom aluminum cooling block, an external radiator and a circulation pump to cool a dual core processor and other highly thermally demanding elements. The cooling block is mounted to the standard XTX heat-spreader to cool the CPU while simultaneously drawing heat away from the motor-driver H-bridges. The radiator is an off the shelf 120mm radiator used in PC liquid cooling applications. Using the lower temperature of the water as a cooling source, SeaBee is able to passively cool the radiator in the surrounding environment.

F. Motors/Thrusters

SeaBee uses six SeaBotix BTD150 thrusters arranged in three main groups: horizontal for forwards and backwards movement, vertical for depth, and strafing to enable five degrees of control.

G. Marker Dropper

The marker dropper mechanism is constructed using linear actuators. This design features a compact frame and a robust releasing structure that is quick, accurate, and reliable. The dropper is connected to the internal electronics using a 4-pin connector

on the end cap. When the control signals activate the relays, the two solenoids on the marker dropper release a marker, which is held in place with two spring-loaded arms. The markers are shaped like mini-torpedoes with angled fin tails to induce a spin during the descent. This shape ensures a highly hydrodynamic marker that will fall in a more streamlined way than that of a spherical marker.

H. Shooter

The SeaBee shooter incorporates a slingshot-powered design to fire a torpedo (Toypedo) through the window. For accuracy and space constraints this model only fires one toypedo when triggered. Due to the projectile's shape and buoyancy, the Toypedo is capable of firing straight for 8ft. The Toypedo, developed by Swim Ways has a hydrodynamic design combining a streamlined fin-stabilized shape with a rubber material and measures 1" in diameter and 5" in length. The Toypedo was designed to be neutrally buoyant, with a specific gravity between .95 and 1.05 relative to water and is cheap and dispensable.

I. Grabber

The grabber mechanism is the component of the SeaBee that is used to pick up the object in the recovery phase. The SeaBee grabber is machined from aluminum. It is lined with several small acrylic teeth that push the PVC briefcase into one of the slots. This revision of the grabber incorporates a retractable design (see Fig. 2).

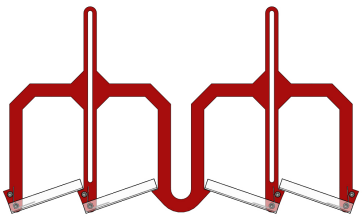

Fig. 2: Rendering of the SeaBee Pipe Grabber.

The grabber is able to slide within the constraints of the robot when it is not in the pool to maintain SeaBee's compact design. When the SeaBee AUV is positioned above the object, the slots open allowing the PVC pipe to fall into place. The slot levers are controlled by small torsion springs that close the slots after the object is acquired.

III. Electrical Design

A. Overview

The electrical team focuses on developing the SeaBee power management, processing, and sensor payload electronics to support the constantly evolving software and mechanical components. To this end, the power board, carrier board, and battery boards are custom designed by the team to provide for all requirements without wasting space or power. An emphasis is placed on designing a modular electrical system, which allows boards to be reused through multiple design iterations and provides support for future unforeseen requirements. In addition, the electrical team has responded to the need for an accurate sonar system with a new design for a passive sonar array that improves upon previous iterations.

B. Power Distrbution

The SeaBee Power Board is a versatile platform supporting a wide array of electrical systems. In addition to supporting the vehicle's power management and regulation needs, the Power Board serves as an interface for any sensors not capable of directly interfacing with the XTX Carrier Board.

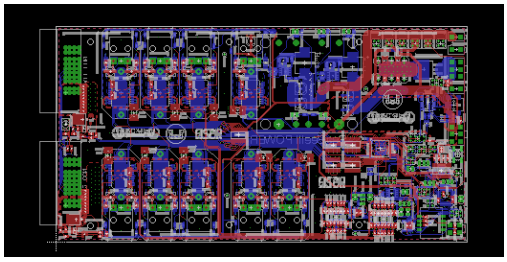

Fig. 3: Power Board schematic.

At the heart of the power board is the Parallax Propeller P8X32A microcontroller.

The P8X32A has eight 32-bit processors, making it well-suited to the robot's high-precision, low-latency requirements. Nicknamed BeeSTEM, the microcontroller is responsible for maintaining communication with these sensors. As the BeeSTEM receives data from the sensor suite, it updates the control loops and forwards the data to the XTX Carrier Board. The BeeSTEM also produces the appropriate commands to initialize the sensors and place them in desired operating modes.

The Power Board produces regulated power at common voltages, such as +3.3V, +5V, and +12V. To ease the process of swapping or adding sensors, the Power Board has auxiliary connectors. The Power Board incorporates nine motor drivers, which are controlled by BeeSTEM. Six of these provide power to the thrusters, and the other three support additional actuators, such as the Marker Dropper or Torpedo Launcher.

C. Computing

Computer design for SeaBee is governed by the vehicle's need to perform complex computer vision and machine learning algorithms in real time in spite of restrictive space requirements. An XTX computer-on-module (COM) was selected for its balance between size and performance. The SeaBee XTX Carrier Board was designed to break out important signals such as power, USB, VGA, and Ethernet, SATA, and USART.

D. Sensors

1) Overview: The SeaBee sensor suite provides as close to a 6-degree-of-freedom state-space solution as is possible within team budget. SeaBee is capable of actively monitoring the status of its own systems through the use of internal temperature sensors, internal pressure transducers, current monitoring in each motor driver channel (nine total), and coulomb counting in each battery pod. Four high-performance Internet-Protocol (IP) cameras equipped with wide-angle lenses are used to complete specific vision-related tasks and to provide an additional position estimate via visual odometry. For the aforementioned 6-DOF localization the SeaBee AUV incorporates a Xsens MTi Attitude and Hearing Reference System (AHRS), a Honeywell HMC6343 3-axis compass, and a pressure transducer used to calculate depth. The primary advantage of using an AHRS like the MTi over a traditional inertial measurement unit (IMU) comes in the form of the unit's on-board signal processor, which allows it to be calibrated for the electromagnetic fields present on SeaBee and thus achieve virtually no drift.

2) Interface: Some of the sensors on SeaBee, such as the IP cameras, are capable of directly interfacing with the XTI carrier board via USB. However, most of the sensors on the robot utilize serial protocols such as RS232, I2C, and SPI. The BeeSTEM is responsible for meeting these bus requirements.

E. Batteries

The SeaBee battery system consists of two custom +22.2V lithium polymer battery packs in parallel for a total of 20,000 mAh. Each pack contains a Battery Board based on the ATMEGA 406 microcontroller. In addition to regulating the LiPo cells to ensure even discharge, the Battery Boards actively monitor state of charge (SOC) through use of current integration, or "coulomb counting". Each Battery Board incorporates an LED display to provide visual feedback to the operator.

F. Killswitch

The killswitch was designed with reliability as the primary focus. The design is robust and minimal while remaining functional in demanding situations. A single reed switch is mounted inside the primary hull, actuated by a magnet tethered to the outer hull. A tiny logic-gate-based circuit ascertains the state of the reed switch and produces a 5V TTY "kill" signal to the microcontroller on the power board. The microcontroller is then responsible for placing the motor and actuator drivers into a low-power state. If the robot enters the "kill state", processing can be halted until the state

is exited and then proceed as normal, a helpful tool during development.

G. Passive Sonar Array

A passive sonar system is essential to completion of the recovery task, and can provide an estimate of relative location to be incorporated into the SLAM implementation. SeaBee features four Reson TC4013 hydrophones placed in a tetrahedral configuration inside a pressure case attached to the primary mechanical frame. The hydrophones are spaced to allow for non-ambiguous determination of phase difference between incoming waves in the 20-30 kHz range. Using a plane-wave approximation, a high-confidence estimation of the relative location of the pinger in three-dimensional space is obtained. A single-channel preamplifier stage is connected to the immediate output of each hydrophone inside the pressure case, and the ADCs are placed inside the main hull. Because of the small distance between the hydrophones and preamplifiers, very little noise is amplified. Since any noise picked up between the preamplifiers and the ADCs will have significantly lower amplitude than the hydrophone signal, filtering is trivial. The ADCs interface with the on-board computer using Cheetah SPI Host Adapters. Each ADC-Cheetah pair (one per hydrophone) can achieve a sampling rate of 40 MHz. As such, the data that reaches the software portion of the sonar solution is as close to the "pure" signal as possible.

H. Controls

Robust vehicle control is achieved through a combination of a carefully-tuned 6-DOF PID (proportional-integral-derivative) Controller and a SLAM implementation inspired by the FastSLAM 2.0 algorithm proposed by Thrun et al. While the SLAM is implemented as a node within the Seabee III ROS package, PID is implemented on the Power Board via BeeSTEM. BeeSTEM effectively serves as a hub for control operations. Sensor data is used immediately to update the PID control loops, then is passed to the SLAM node, which uses the new data to update its state estimate. Based on this estimate, new setpoints are sent back to the BeeSTEM. This low-level PID implementation is highly favorable, as the frequency at which the control loops update is only limited by the rate at which measurements are taken.

IV. Software Design

A. Overview

The physical capabilities of the SeaBee AUV are constantly changing in terms of both mechanical and electrical systems. This has necessitated the design and use of a robust, dynamic software architecture specifically engineered to maximize both functionality and ease of implementation through the use of a modular paradigm. Similar modules are connected via generic interfaces, allowing for high levels of specificity where necessary while still ensuring the trivial addition and removal of modules as physical constraints vary, even at runtime. Furthermore, several pipelines, including vision, localization, and navigation, have been implemented to handle the complex process of feature conversion and filtering, starting with low-level, often noisy feature sources such as physical sensors, and ending at the high-level representations required for efficient planning and decision making.

B. Architectures

1) Ubuntu: The Seabee uses Ubuntu Linux 10.10 "Maverick Meerkat" as its operating system. This selection was made based on the open-source nature of Ubuntu, its portability, and its good support for the intended software architecture.

2) ROS: The Seabee has used ROS, an open-source toolkit developed by Willow Garage, since the 2010 RoboSub competition. ROS, or the "Robot Operating System", provides a language-generic, modular paradigm for the development of software systems along with fairly generic implementations of many common algorithms known to the field of robotics, including such categories as sensing, navigation, planning, and visualization.

3) Quickdev: While ROS provides a solid foundation on which to build, working within the toolkit often involves creating unnecessary redundancies across implementations. Furthermore, the serialization-free communication described above is traditionally time consuming to set up, as modules utilizing it must follow the "nodelet" paradigm rather than the more common "node" paradigm. We solve this issue by maintaining a set of scripts and generic wrappers around the more commonly-used ROS components in an open-source package called "quickdev" available in USC's "usc-ros-pkg".

C. High-Level Implementation

1) Vision Pipeline: Seabee views the world through two PointGrey Firefly USB cameras: one facing forward and one facing downward. Both cameras are configured to stream bayered 320x240 images at 30 hz. While the hardware interface to these cameras is USB, they support the IIDC 1394-based Digital Camera Specification over USB, allowing for the use of "firewire" camera drivers. Figure 1 shows an example of some of the competition objects as seen by the cameras.

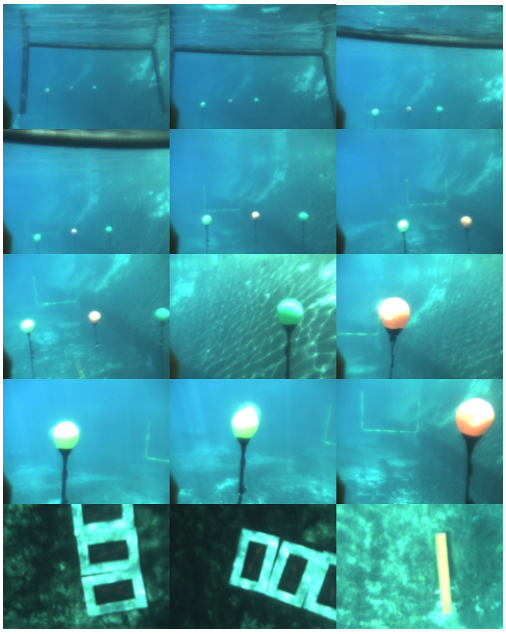

Fig. 4: Seabee's view of the competition pool.

Images streamed from the cameras are bayered and distorted by various optical effects, including those from the camera lenses and the results of different physical mediums surrounding the sensor tubes containing the cameras. To compensate for these effects, we use a community-developed ROS node called "image_proc", which utilizes several common OpenCV functions to de-bayer and un-distort (given a camera calibration) the incoming raw images.

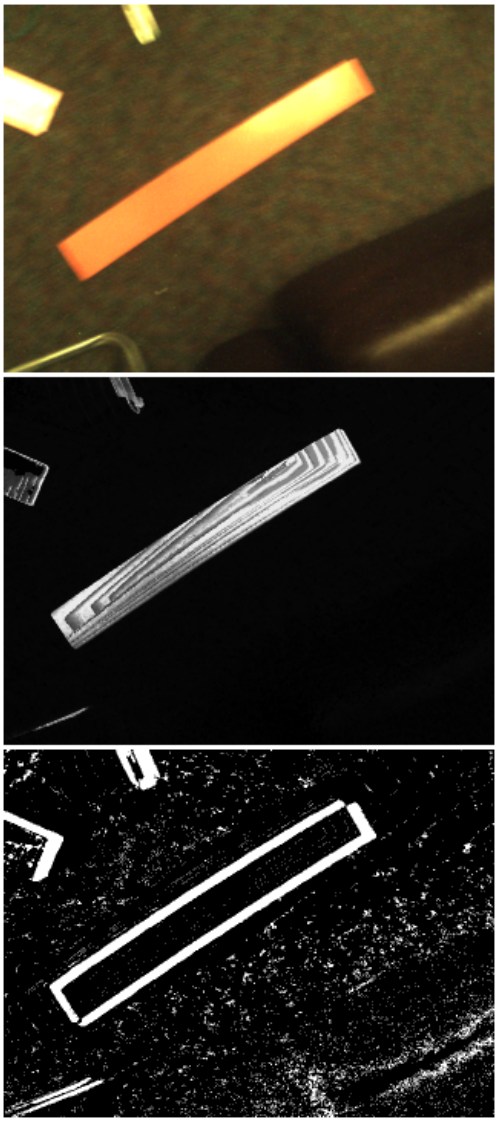

Fig. 5: An example of an "adaptation image" fed to the color classifier.

Following basic low-level image filtering, we perform a color-space conversion from BGR to HSL. In order to optimize the vision algorithms farther down the pipeline, we seek to mimic neural adaptation and avoid unnecessarily processing pixels that are not changing by a

sufficient amount. Pixels found to have changed by a minimum amount are recorded in a binary mask, which is published along with the corresponding HSL image. We call this mask-image pair an "adaptation image." Figure 2 shows an adaptation image fed to the color classifier, reducing the number of pixels fed through the classification algorithm by over 90%. The bands seen in the middle image (probability image) are the result of pixels that have not yet changed enough to necessitate re-classification.

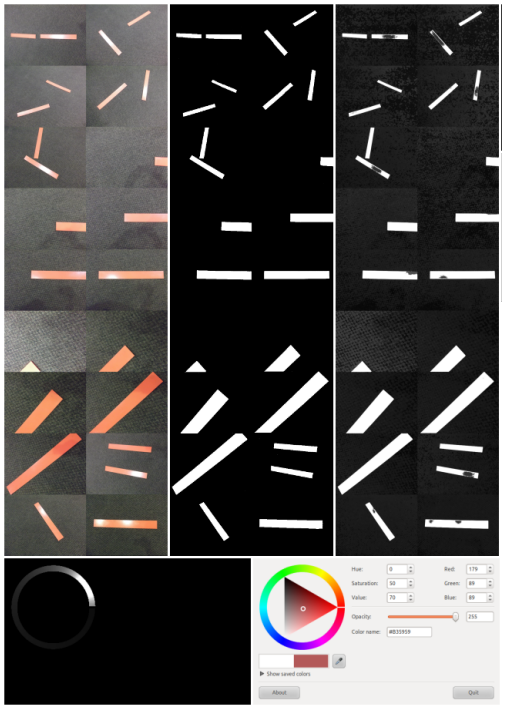

Fig. 6: An example of a probability image.

Newly-generated adaptation images are fed through a color classification node which utilizes a naive Bayes classifier trained on actual camera footage. The classifier identifies newly-changing pixels, calculates the likelihood that a given pixel should be classified under each of a set of colors, and then publishes these probabilities as an array of images, with each image representing the classification for the corresponding color. Subsequent color-sensitive feature extraction algorithms in the pipeline can utilize these probability images in their calculations. Figure 3 shows an example of a probability image for the orange color produced by the color classifier next to the input image as well as the image on which the orange color model was trained.

2) Recognition Pipeline: Recognition and subsequent localization relative to landmarks, or unique competition objects, is critical to success in the RoboSub competition when the robotic platform used is not aided by a Doppler Velocity Logger; however, these landmarks are subject to change each year. Seabee utilizes an extensible landmark recognizer that accepts a landmark filter and calls on child modules to perform specialized landmark recognition. This ensures that, with the exception of drastic changes to the competition, landmark-dependent algorithms can remain mostly unchanged, while specialized recognition algorithms can be easily developed, tested, and deployed through a standard, familiar interface.

SeaBee's specialized landmark algorithms can offload a significant amount of specialization to a central, generic system, yet still ensure the use of alternate, arbitrarily specialized recognition methods. SeaBee accomplishes this by calling on a scale- and rotation-invariant feature recognition system which utilizes OpenCV 2D feature extraction, or contour extraction, performed on color-classified images produced by the vision pipeline. A set of template contour features, in the form of normalized, rotation-aligned histograms, is trained for any landmarks or landmark components and passed along with any candidate contour features located within incoming images to a generic recognition algorithm, which calculates and returns match qualities for each template-candidate pair.

In its current state, SeaBee uses a specialized recognition algorithm for each unique landmark, excepting landmarks differing only in color. Furthermore, recognition is specialized based on the expected sensor source of landmarks relative to the sub; for example, we assume that pipelines and bins will only enter the field of view of the downward-facing camera, while buoys, hedges, and windows are expected to be found only in the field of view of the forward-facing camera. Given this assumption, we only search

for the former set of landmarks in the images streamed from the corresponding source, and so on.

We assume that certain landmarks are only located within certain parts of the competition pool, and we further assume that given an arbitrary set of goals, only some subset of these landmarks need to be recognized. Given these assumptions, we conclude that it is possible to search for only some subset of landmarks dependent on the current location and/or goal. Therefore, it is desirable to utilize some simple means of applying a landmark filter, with either narrowing or widening constraints, through recognition algorithms, in order to improve performance. We accomplish this via a custom color- and shape-based filtering API that accepts a list of filter items, each specifying either a narrowing or widening constraint to be applied to the color or type of a landmark. For example, when we are attempting to locate a buoy, we look for orange pipelines and buoys of any color on approach, then look for buoys of a single color on each buoy-touching attempt, then look for only orange pipelines and yellow hedges as we attempt to locate the first hedge, etc.

Fig. 7: The Xsens MTi inertial measurement unit

3) Sensing and Localization: Currently, the most advanced sensor on the SeaBee is an XSens MTi IMU. This inertial measurement unit provides us with "drift-free" 3D heading and acceleration data calculated by an on-board EKF fed by the device's accelerometers, gyroscopes, and magnetometers in realtime at 100 Hz.

SeaBee's hull is also fitted with an external pressure sensor, which is used to estimate the absolute depth of the AUV below the surface of the water via an experimentally-derived conversion from arbitrary pressure units to a distance in meters. With the current electronics infrastructure, this measurement is taken at about 10 Hz.

Without a Doppler velocity log (DVL), we must rely on alternate, often noisy sources of odometry. We have found it necessary to develop both a generic realtime simulation of our vehicle's dynamics, as well as a generic Bayesian measurement fusion system capable of combining all components of all observables, whether simulated or actual, into corresponding "filtered" measurements.

Given current sensing capabilities, the linear components of pose and velocity (those that would be trivially provided by a DVL) are the most difficult to obtain. In order to compensate, we run a realtime simulation of the sub's dynamics using a software library called BulletPhysics. Given the vehicle's mass, per-axis linear drag coefficients, the current thruster configuration (per-thruster capabilities and relative pose), and the motors value being set on each thruster, we are able to calculate all components of pose and velocity with moderate accuracy. We then fuse this output and any other odometry measurements together on a per-axis basis into a complete odometry estimate.

When combined, the filtering and simulation modules allow for maximum functionality and modularity within the constantly changing constraints of the platform; as sensors are added and removed, the accuracy of the corresponding measurements will vary accordingly. As an added bonus, these systems also allow for the advanced testing of other software modules via arbitrary levels of simulation of sensor values and other information, in circumstances when a desirable real-world testing environment is not practical or is entirely unavailable, or when the effects of a theoretical change to the system need to be studied.

4) Navigation Pipeline: As with most of the software, the team sought to build the navigation system out of modules with varying levels of specialization, connected by generic interfaces. At a high level, the current implementation accepts a series of navigation constraints in the form of "waypoints", generates a trajectory with an arbitrarily high "temporal resolution", and then attempts to follow the trajectory within an additional set of constraints. This functionality is distributed over several modules including a trajectory planner, a high-level trajectory follower, a low-level velocity-based controller, and a platform-specific serial interface. Any module wanting to accomplish trajectory-based control of the vehicle must provide a trajectory to be followed. Within the system, a trajectory is composed of discrete intervals, each containing a constant acceleration over that interval and the desired state of the vehicle at the beginning of that interval, in the form of a waypoint (pose and velocity).

V. Future Work

In the coming year USCRS will enter the second half of the SeaBee IV design cycle. New designs for the power board, carrier boards, and battery board are complete and will undergo a final internal review before being printed. Some notable electrical improvements will be increased processing capability, additional motor drivers for simultaneous use of all necessary competition elements (dropper, grabber, and shooter), battery status indication via coulomb counting. Designs are complete for external battery pods and a new internal frame, reducing the time spent with the hull open and thus the risk of damage to sensitive electrical components. USCRS will hold both a Preliminary Design Review (PDR) and Critical Design Review (CDR) to obtain feedback from research and industry experts. In addition, USCRS will increase its outreach into the community by repeating involvement in the LA Times Festival of Books, USC Robotics Open House, and the USC Viterbi School of Engineering Open House while also increasing participation to include USC's Engineering Week, partnership with other USC Organizations for outreach into elementary and middle school STEM education, and a formal program with high school after-school math and science programs. For more information, please visit http://uscrs.sagarpandya.com/wordpress/

Acknowledgment

Support from The University of Southern California, iLab, the USC Dornsife College of Letters, Arts, and Sciences Machine Shop, and the Viterbi School of Engineering allows USCRS to continue to be an integral part of student research at the USC Viterbi School of Engineering.

Thank you to our industry sponsors: the Boeing Company, SolidWorks, Fischer Connectors, Northrop Grumman, Digi-Key Corporation, Lockheed Martin, and McMaster-Carr.

References

[1] Rish, Irina, *An Empirical study of the naive Bayes classifier*, IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, 2001.

[2] Emery, William J., Thomson, Richard E., *Data analysis methods in physical oceanography*, Gulf Professional Publishing. p. 83, 2001.

[3] Pearson, K.G., *Neural Adaptation in the Generation of Rythmic Behavior*, Annual Review of Physiology , 1997.

[4] Julier, S.J.; Uhlmann, J.K, *A new extension of the Kalman filter to nonlinear systems*, Int. Symp. Aerospace/Defense Sensing, Simul. and Controls 3 , 1997.

[5] Thrun, Sebastian, Montemerlo, Michael, Koller, Daphne, Wegbreit, Ben, *FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping tha Provably Converges*, IJCAI 2003 Workshop on Empirical Methods in Artificial Intelligence, 2003.