

University of Rhode Island Autonomous Surface Vehicle

RamBoat 2014

URI ASV TEAM
University of Rhode Island
June 15, 2014



Abstract

The University of Rhode Island's (URI) Autonomous Surface Vehicle (ASV) Team overhauled the competition vehicle for this year's AUVSI RoboBoat Competition in Virginia Beach, VA. Through the hardwork of the team and gracious support from the URI Ocean Engineering Department and sponsors, the boat has successfully been brought back to a competition-ready vehicle.

1 Introduction

The University of Rhode Island is attending the AUVSI RoboBoat competition for the 6th year – this year with a completely new boat. After last year's competition, it became evident that drastic changes needed to be made to the existing system. At this point it was decided that a "hull-up" reconstruction of the vehicle was necessary. This overhaul covered all aspects of the vehicle except for the vision and propulsion systems.

2 Vehicle Overview

2.1 Mechanical Systems

2.1.1 Hulls

With support from Navatek Ltd. new hulls were designed for the competition. Navatek invited students into their new South Kingston office to work on the design of the hulls. The hull design is an offshoot of the tandem lifting body surface vessel. The ASV hulls consist of a catamaran with a bow lifting body (BLB) and aft lifting body (ALB). Although the vessel will not be traveling at

speeds great enough to generate lift, they serve as fixed control surfaces to damp the pitch motions. Damping these motions provides a stable platform that for all other systems to operate on.

Each demi-hull is comprised of a port and starboard hull section machined of 10-pound Precision Board foam. Precision Board is a high density closed cell polyurethane foam produced by Coastal Enterprises. The hull were machined on a 5-axis Thermwood CNC machine for accuracy. The demi-hulls are provided with a nominal 1/8" thick carbon fiber plate (c-plate) centerline vertical keel (CVK) to add rigidity to the components when joined. The Precision Board was joined to the c-plate using an urethane adhesive, PB Bond 240.

The BLB and ALB each are composed of a top and bottom half machined from 20-pound Precision Board separated by a nominal 1/16" thick c-plate horizontal strength member in a similar fashion to the demi-hulls. The lifting bodies were drilled and counterbored to recess a pair of stainless steel socket head cap screws which mount to adhesive grip internally threaded anchors in the bottom of the hulls.

The exterior finish starts with multiple coats of PB Resin, a two-part modified epoxy resin to seal the Precision Board and harden the exterior. Defects in the surface are corrected using automotive body filler and sanding up to 200 grit. A filler primer and 400 grit sanding smooths the surfaces in preparation for paint. Two coats of Pre-Kote establish the base layer and another two coats of marine grade paint as a topcoat.

The tops of the hulls are capped using acrylic plate to prevent water from accumulating in the displaced area. It is secured using a latex based caulk.

2.1.2 Frame

A ladder frame was constructed using 80/20 extruded aluminum to minimize weight while maintaining the desired strength properties. The frame consists of two rails and four crossmembers. Two sizes of stock were used: 1010 refers to 1" x 1" extruded stock and 1020 refers to 1" x 2" extruded stock.

An aluminum plate was mounted atop the bow and stern of each hull using the same hardware used to secure the lifting bodies. Angle brackets were used to hold the plate to the longitudinal frame rail that runs the length of each hull.

The center pair of cross members serve as a base for the electronics box. The stern crossmember is the mounting face for the two stern motors. The bow crossmember supports the bow thruster and imaging hardware.

2.1.3 Propulsion

Propelling the ASV are three 12V trolling motors. As mentioned previously, the motors are the same model as years past. The motors were chosen because of their weight to thrust ratio. These motors weight 3lbs and provide 18lbs of thrust – the closest motor for comparison was a 15lb motor providing 30lbs of thrust. It was decided that the additional thrust was not beneficial for the vehicle in order to keep total weight lower. However, different from last year is the number of motors. RamBoat 2014 utilizes a bow thruster to provide increased maneuverability at low speeds. To protect the propellers from foreign lake debris and as a safety precaution, shrouds were manufactured and installed.

Shrouds The shroud for the motors was designed in SolidWorks. Because the motor is designed to run in both forwards and reverse, the inner and outer diameters of the shroud are constant. The shroud slips around the rear of the motor. It is held in place with a set screw in an alignment hole in the skeg of the

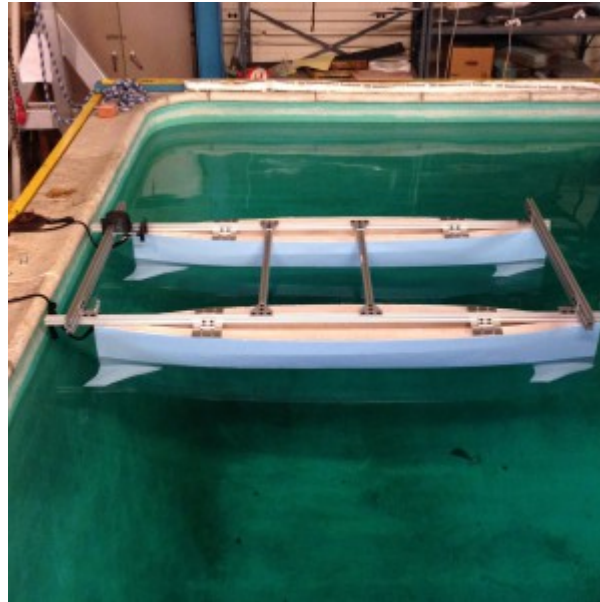


Figure 1: Initial hydrostatic tests of RamBoat 2014

motor. The drawing was printed with ABS thermoplastic in a Stratasys Dimension Elite 3D printer. A layer of plastic paint was applied to the shrouds to prevent water from entering the pores of the shroud. The shrouds have proven themselves in both lab testing and pond testing, where the side of the pool and pond rocks have not yet damaged them.

Motor Mounts During initial testing of the ASV, the stock motor mounts from the motors were used. These mounts simply applied pressure to a plate that mounts on the transom of the boat. It became evident during successive testing that these mounts were not satisfactory due to their ability to slip off of the 80/20 frame. In response to this, pillow blocks were machined that allow for motor position to be set precisely every time and then removed for transport.

Motor Control The ASV in past years utilized a dual-motor controller that provides mixing of the motors onboard. A pro to this was the simple interface to control the motors. The downside was that the motor controller

board did not provide feedback; therefore the motors were unable to be monitored. RamBoat 2014 has a new motor control system due to the use of a third motor for the bow thruster. As a result three Parallax HB-25 motor controllers were integrated into the system. The motor controllers are high current H-Bridge chips that accept a servo pulse to control speed and direction of the motor (1000us:full reverse, 1500us:stop, 2000us:full forward). The HB-25's also utilize small PC fans for self-cooling and are fused to provide safety against short circuits.

To interface the HB-25 motor controllers, the Pololu Mini Maestro 18 Channel Servo Controller . The Servo Controller receives commands from the main computer using a python library developed by the team. The library is able to send pulse width data to the controller on up to 18 different channels. The commands are generated by the motor daemon and the navigation processing, which is spoken of in more detail in the Navigation section.

2.1.4 Electronics Box

RamBoat 2014 utilizes a PolyCase 14"x16"x8.5" polycarbonate case for the electronics enclosure. This provided a durable, yet lightweight solution to mounting the electrical systems on the vehicle. Waterproof connectors for all power and data were used to ensure a water-tight case.

2.2 Computing System

2.2.1 Intense PC

The main computer on RamBoat 2014 is the Intense PC, a fanless PC that incorporates a full heatsink into the case of the computer. The computer has an Intel-i7 processor with 8GB of RAM and a 128GB Solid-State Drive. Previous RamBoats incorporated two FitPC's – one to run missions and one dedicated to vision processing. The intense PC is more than capable of running both mission and vision onboard.

2.2.2 BeagleBone Black

All low-level I/O is handled via the BeagleBone Black, a single-board computer running a full Debian Linux distribution. Since the BeagleBone runs Linux, LCM was installed to easily integrate the board into the system.

2.3 Power System

2.3.1 Batteries

This year the ASV is powered off of six Inspired Energy Lithium Ion batteries (NH2054HD24). Each battery supplies 14.4V to the system with a power output of 4.8Ah per battery. The batteries also have internal data logging abilities. This allows for the reading of instantaneous voltage, current, temperature and many other features in real time. The ability to access this feature has undergone development but was not imple-

mented on this year's boat.

2.3.2 Power Distribution

In order to combine all of the Li-Ion batteries without causing cross-charging, each battery is fed into a Schottky diode and then to both the power distribution board and the motor controllers. For a safety precaution, the power supplied to the motor controllers is first passed through the kill switch with a transient suppression diode connected on the output side to prevent voltage spikes. The power distribution board consists of a VICOR V28C12C100BL isolated DC-DC converter along with three LM-317 linear voltage regulators. The VICOR drops the voltage from 14.4V to 12V and then the linear regulators drop the voltage to 6V, 5V, and 3.3V. The board is able to provide 100W of power, well within the need of the electronics connected to the distribution board.

3 Software

3.1 Lightweight Communications and Marshalling

The ASV's software is designed around the Lightweight Communications and Marshalling library, abbreviated as LCM. The library is a connectionless communications system in which data is broadcast freely to all systems via the UDP protocol. Data is organized into C-like structs that are generated for various programming languages, such as C++, Java, and Python, and is automatically encoded and decoded when transmitting. In order to specify which information to receive, messages are broadcast on a specific channel by being given a tag. In order to receive data from a channel a process simply subscribes to a channel. This allows an overall structure of small, independent and modular processes instead of a large, complex, unwieldy

program. The modularity goes even further as it does not matter to the receiver what is transmitting the data packets over LCM, and so LCM messages can be logged and played back later.

The software design for the ASV was good in organization but lacked completion and got bogged down in a swamp of bugs and issues. A shortage of testing meant that even simple errors had not been found and rectified in time. Nevertheless, it was decided to keep the old structure and continue to fix and/or update the existing systems as required. This effort focused mainly on the Navigation systems and a new simulator-like tool abbreviated as the VLS.

3.2 Navigation

The most pertinent system to our success, as learned last year, is a working navigation system. Choosing our new propulsion configuration was the first decision the team faced. Past experience with two motors showed that using only two motors in a differential configuration is problematic. Such a system is uncontrollable, and resulted in a complex system of navigation which required switching controller types and performing multiple actions per way-point. In addition it meant that previous vehicles could not counteract currents perpendicular to its heading, or perform strafing maneuvers.

As mentioned previously, the team elected for a three-motor propulsion system. Two motors are positioned at the stern of the vehicle, each vectored 45° outwards of the vehicle. The third motor is mounted on the bow of the vehicle, perpendicular to the length of the vehicle. In order to see the controllability of the vehicle, a vector can be drawn for each thruster of the forward, side-slip, and torque produced by a unit of force from each thruster, and combine them to create the fol-

lowing matrix equation:

$$\tau = \begin{bmatrix} X \\ Y \\ N \end{bmatrix}$$

$$\tau = \begin{bmatrix} 0 & \cos(\theta) & \cos(\theta) \\ 1 & -\sin(\theta) & \sin(\theta) \\ B_x & S_x \sin(\theta) - \frac{S_L}{2} \cos(\theta) & -S_x \sin(\theta) + \frac{S_L}{2} \cos(\theta) \end{bmatrix}$$

$$\times \begin{bmatrix} F_{Bow} \\ F_{Stern-Port} \\ F_{Stern-Starboard} \end{bmatrix}$$

Where θ is the angle outward of the stern thruster, B_x is the distance forwards of the center of gravity for the bow thruster, B_x is the distance behind the center of gravity for the stern thrusters, and S_L is the distance between the stern thrusters.

After plugging in the values needed to the thruster matrix, immediately an uncontrollable system can be recognized if the matrix is non-invertible. This can be seen with the old system by removing the bow thruster from the matrix. Otherwise, inverting the matrix results in a straightforward reverse mapping that can be used to determine the forces needed by each motor in order to provide an overall desired force vector upon the ASV. This process is referred to as motor-mixing, and has been proven to work well even though our estimates for the parameters in the thruster matrix are not exact. Even if it is not perfect, an error can be fixed quite easily as long as the errors are small since the error introduced by fixing the original error will be another factor smaller.

In addition to making human-powered control intuitive, this simplifies the autonomous control problem since it can effectively treat each of the two surface dimensions and the ASV's yaw as independent variables. In reality this is not true as the positional error must be calculated in the ASV's reference frame which is dependent on its heading. However, as long as the heading settles faster than the position does this is

usually not an issue and the system can be treated as such. Therefore, in the interest of simplicity a straightforward PID controller was implemented that tracks each variable separately. This has had the side-effect of making way-point navigation much simpler because it is a single-step process of setting the desired location and heading and letting the controller do the heavy lifting.

3.3 VLS

The VLS is RamBoat 2014's secret weapon this year. Because there is not a word to describe such a tool developed this year the abbreviation VLS, standing for Visualization, Live, and Simulation was chosen. At the end of the last competition, it became obvious that a better way of debugging systems was needed. The two bottlenecks to debugging were a shortage of testing time, and a lack of a good visualization system for the ASV's systems. A tool included with LCM spy provides a method for creating visualizations by creating a plugin to create a visual for a specific channel. However it does not allow one to create a visual that utilizes multiple LCM channels. In addition, the tool uses Java's Swing library which is not an ideal system for creating visuals. The decision was made to begin work on a new visualizer which uses information from all available LCM channels. Because of how LCM works, the visualizer could be used to visualize information from logs as well as live during a test, without it even knowing the difference. To improve this tool, a simulator was added, as well as useful functionality for when doing live testing.

3.3.1 Visualization

The rendering system chosen for is the LightWeight Java Game Library, famous for it's use in the popular game Minecraft. It is an open-source platform-independent library designed for gaming. It is the combination of a Java Native Interface binding of OpenGL,

for rendering hardware-accelerated graphics, utilities, and the input API JInput. In addition, the add-on library Slick was used for font rendering. This was chosen with the interest of being able to integrate code from the open-source LCM library and related Libbot libraries. The first visualization element to be implemented was satellite imagery in order to place the ASV on the map so to speak. After that, a visual of our LiDAR's data was added, as well as a camera feed from the main camera. This allowed us to go back and look at our logs from the previous competition and confirm that our vision system from last year worked excellently. The addition of visualization of way-points for the navigation system also helped to confirm the cause of the bug that prevented us from qualifying.

3.3.2 Live

The VLS became a useful tool for live testing because it acts as mission central. The user can see the state of the systems and is provided with a software-based remote control system. In addition, the user can instead elect to simply set waypoints by clicking on the map. Planned is also an integration of Libbot's process management system which uses LCM to manage running processes on the ASV from a shore computer. This would allow the user to quickly configure and fire up the ASV for a run.

3.3.3 Simulation

Being able to simulate the ASV in order to debug our systems is critical. In order to implement the navigation dynamics, a simplified marine model was developed and then converted to a discrete-time zero order hold equivalent model. The corresponding discrete time matrices can be used to simulate the ASV's dynamics recursively. The matrix operations are implemented with the open-source C++ header-only library *Eigen*. In

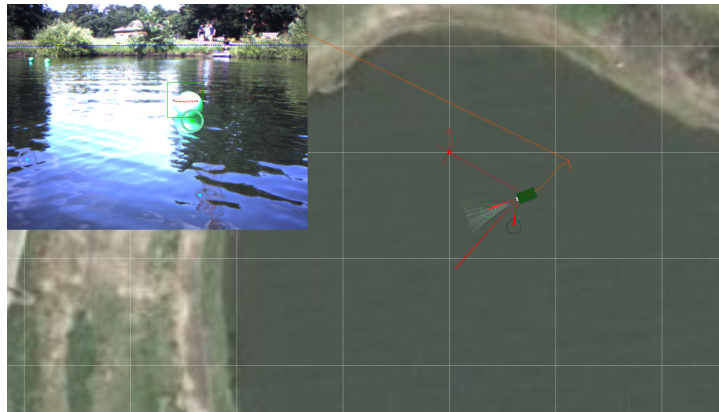


Figure 2: Screenshot of VLS running past mission logs

addition, a method of simulating vision detections with a virtual buoy course was developed in order to test tracking algorithms and methods of navigating the obstacle course task. Noise is added to the simulated detections and detections have a probability of not occurring in order to test robustness of tracking algorithms.

The last major feature of the VLS that is especially important to simulation is that it includes a live python console implemented with the open-source Jython library. Jython is very powerful in this application because it allows Python code executed with it to directly access and call Java code. With the use of Jython, the high-level systems in the VLS effortlessly interact with the underlying Java system for in-depth debugging. In addition, the live python console can be used to compartmentally test the Python utilities as well as helping in debugging of the VLS itself.

4 Mission Tasks

4.1 Simon Says

This challenge resented a clearly defined problem. The LEDs flash in a sequence with a constant start and stop time. Using the assumption that muddy water diffuses light better than clear water, a specific point of

light does not need to be detected. Therefore, Ramboat 2014 utilizes a color sensor to detect the overall color of the water.

In order to detect the sequence, the first unknown parameter would be the delay of when the light sequence starts as it is unlikely that the sequence will start instantaneously. To account for this, the data for the entire sequence will be recorded. Given a specific time delay, the color values for each segment are estimated by averaging the samples. To estimate the actual time delay, a grid search will be performed over a certain range to find the time delay which minimizes the resulting variance of the data when compared to the estimated color values.

4.2 Obstacle Avoidance

4.2.1 Buoy Detection

The first challenge to this task is detecting the Polyform buoys. We already have a working system to do this, using a combination of a LiDAR and machine-vision camera. The idea is that the LiDAR can provide accurate detection and locations for buoys, while the camera can provide identification of the color. By modeling the spatial projection that is inherent in the camera taking images, the LiDAR points are able to be projected onto the images. By matching detections from

the LiDAR with detections from the camera positions and colors can be associated. For avoiding the black and yellow obstacles this entire process is not necessary as the color for the buoys marking the entrance and exit are all that are necessary.

4.2.2 Navigating the Course

The buoy detections are accurate enough that RamBoat 2014 is able to track them in order to provide positional feedback to the controller. In addition, locating the entrance buoys and extrapolating should provide a good estimate of where the exit to the course will be. The ASV will plot a path directly across and then simply modify the path when it detects an obstacle buoy. In addition to help avoid going off the course, the ASV will extrapolate, where the boundary of the course should be as well as use detections of obstacle buoys to confirm that space ahead of it is inside of the course.

4.3 Automated Docking

This task is mostly an image processing problem to identify the dock symbols. First, edge detection is used to find the rectangular outline of the dock markers. This is the simplest detection employed for the task and will then allow for tracking of the location the markers. Then, by using a second layer of edge detection the circle is detected and also series of line segments for the cruciform and the triangle. By looking at the angles of the line segments RamBoat 2014 can determine which of the other two are the triangle and the cruciform. Once the docks are identified then the ASV tracks the rectangle of the symbol and guides the ASV into the dock.

4.4 Acoustic Beacon Positioning

4.4.1 Pinger Detection

Given the success of past year's ability to detect buoys, the approach for this task is to first identify all potential buoys for the mission. After all buoys have been located, the buoy with the pinger is differentiated by the use of two hydrophones mounted on the bow of the vehicle. Using the two hydrophones together, the time difference between pings to each hydrophone will be used for the localization of the pinger buoy. The hydrophones interface with the on-board audio port on the main processing computer (Intense PC).

4.4.2 Maximum Likelihood Estimator (MLE) for Pinger Heading

The estimator provides a single heading estimate for every two seconds of data. The model used for the pinger's signal is as follows:

$$x[n] = A \sin(2\pi f_0 n + \Delta) p(n + \Delta) + w[n]$$

$$n = [0, 2f_s]$$

A is an unknown amplitude, $w[n]$ is assumed to be white Gaussian noise, p is a pulse function with a period of four milliseconds (pinger pulsing). The corresponding maximization function for a MLE is:

$$J(f_{0,\Delta}) = \sum_{n=0}^{N-1} x[n] \sin(2\pi f_0 + \Delta) p(n + \Delta)$$

Given that there is a specific range of frequencies that will be used during the mission over a set window of time, a grid search was implemented. To simplify the grid search, a separate estimator was used to determine the frequency of the pinger ahead of time; assuming that the frequency does not change. Estimation of the frequency is easily done

with the following maximization function – a modified Fourier Transform:

$$J(f_0) = \left| \sum_{n=0}^{N-1} x[n] e^{j2\pi f_0 n} \right|$$

The time delay is estimated for each hydrophone, and an overall delay is determined using the speed of sound in water and the distance between each hydrophone – helping to avoid any type of interference due to a pinger on a separate course. If it is assumed that the hydrophone is much farther away than the hydrophones are actually separated, the following estimator for the heading by the invariance property holds true:

$$\hat{\theta} = \arcsin \left(\frac{\Delta \hat{C}}{L} \right)$$

6 Sponsors

We would like to thank the following companies for sponsoring the URI ASV Team:



5 Conclusion

Over the past 8 months, the University of Rhode Island created an entirely new autonomous surface vehicle for submission into the 2014 AUVSI RoboBoat competition. This overhaul incorporated positive aspects of previous year's vehicles, such as the use of LCM, the vision system, motors, etc. With the use of the newly developed VLS coupled with frequent pond testing, RamBoat 2014 will prove to be a strong competitor in Virginia Beach on July 8th-14th