AUVSI RoboBoat Competition 2011

Cedarville University

Jimmy Myers, Tyler Chan, Mitch Muhlenkamp, Timothy Swanson
Advisor: Dr. Timothy R. Tuinstra

Abstract:

As technology advances, more and more systems are becoming autonomous. Our task has been to make an autonomous surface vehicle (ASV). This vehicle needs to travel in water through a marked channel while avoiding randomly placed obstacles. Since the system is to be completely autonomous, it has to have vision, be able to analyze its surroundings, and then make a decision from the available data on where the boat should navigate. This paper will show our boat design and navigation algorithm of how the vehicle will see and make decisions

Vehicle Structure:

The first step in our design process was choosing the type of boat we wanted to use. The main qualities we were looking for were stability and ease in manufacturing. Because of this, we chose to use a pontoon-style boat. Pontoon boats are very stable, although they lack speed. To build this boat, we used PVC pipes and chose to use 8” diameter Schedule 40 PVC for the pontoons. As with most things that seem easy, we ran into some complications putting our boat together. Eight inch diameter PVC is much harder to cement together than the forgiving two inch PVC. However, we were able to make adjustments to cover up for manufacturing errors to build a solid frame. After we made the base we made the platform and added it to the top of our boat. We made it much higher than the pontoons in order to keep our electronics as far away from the water as possible. A picture of the basic frame of our boat is shown in Figure 1 below.

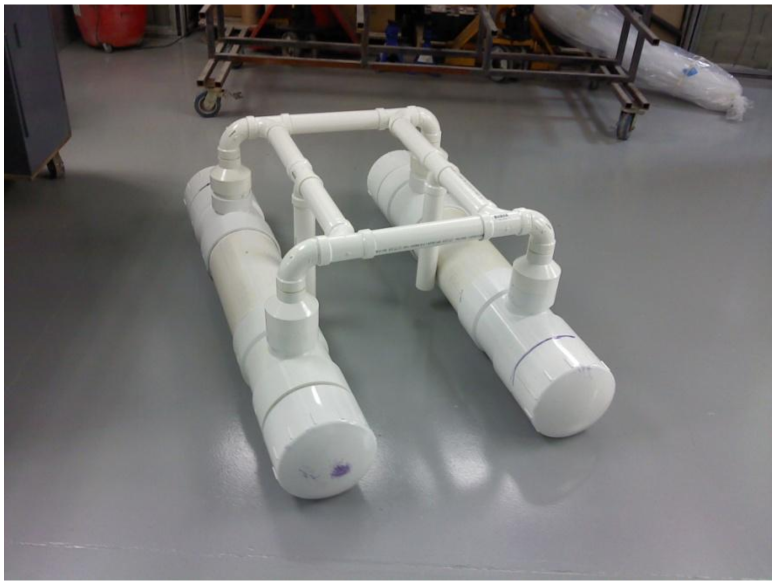

Figure 1. Vehicle Structure

Enclosure for Electronics:

After making the structure of the boat we had to decide how to house the components that were going to make our vehicle autonomous. Since we use electronic components to make this work, they had to be housed in a way that no water would spray up and cause catastrophic failure. A box was designed out of Plexiglas and Starcraft Marine plastic. Since our platform was fairly steady and we wanted a simplistic design, we decided to make the bottom of the box the platform of our boat. By sealing the bottom with caulk we have a watertight seal.

While the bottom is watertight, we did not decide to do the same with the top. Since electronic components heat up during usage, an air vent was designed at the top of the box. Instead of having the top sit directly on the sides of the enclosure, we offset it by raising the four support columns on the four corners of the box. This allowed a large enough gap for air to circulate but made it hard for water to get in. The top of the box was then designed with the air gap in mind. Since we had that gap, we did not want water running off the sides and leaking onto the electronics. Therefore the top was made larger so that there was an extra inch on each side. This allows for the water to drip off and not go into our enclosure.

Also, in case of extreme angles when putting the boat in the water we did not want out electronics to slide around in the enclosure. In order to keep everything in the place we laid them, we added heavy duty Velcro strips to the bottom of each component. We took into account all of the electronics we had and then made a layout to make sure we could fit all the components before laying the Velcro down. These strips are very strong and have no problem holding the heavier and more bulky components.

Also, we wanted to be able to make the top of our box secure, so that it could not fall off and expose our electronics to water. As a result we added threaded rods on the posts of the box. After attaching the top we add locking nuts so that the top will stay secure, but we could still take it off to fix any problems that might occur. A picture of our boat with platform and electronics containment box is shown in Figure 2 below.

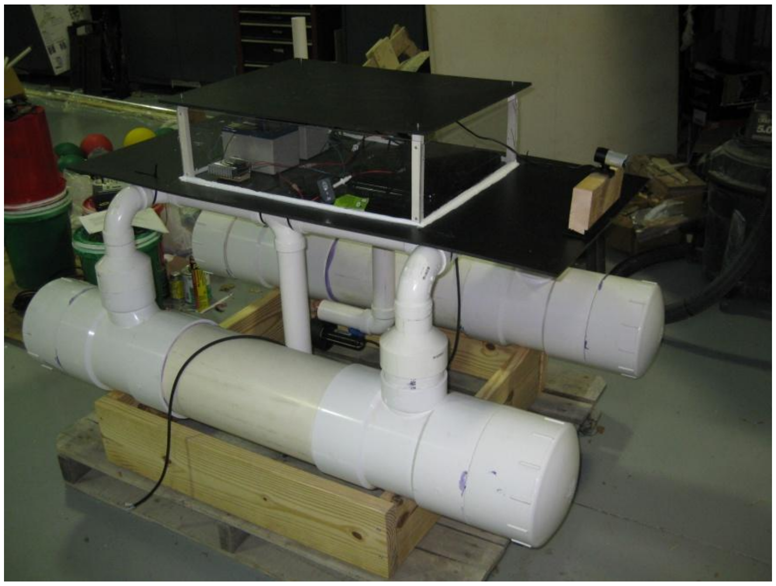

Figure 2. Vehicle with Electronics Enclosure

Propulsion System:

The other design issue with our boat was how to make it move. We ended up buying two Seabotix BTD150 Thrusters. These can supply up to approximately 6.4 ft/lbs of thrust each. Once we had our thrusters, we needed to decide how they should be used and where they should be placed. We eventually decided to place the motors side by side so that we could use differential steering to navigate our boat. We originally were going to place the thrusters in the back, but ultimately decided to put them in the middle of our boat in order to make turning much easier. Figure 3 is a picture showing the location of the Seabotix thrusters.

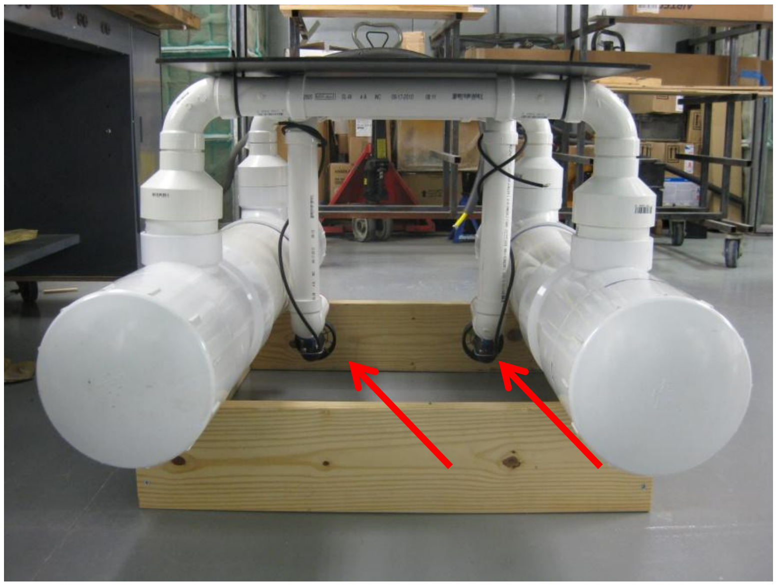

Figure 3. Thruster Mounting

Video Sensing:

In order to make our system autonomous our boat needed to analyze its surroundings so it could make a decision on where it should navigate. We thought that the most beneficial way to gather this data was through live video. As a result our algorithm is based on video processing of live video through the Matlab image acquisition toolbox. When looking at webcams and their ability to operate effectively outside we had to be sure that our webcam would be able to autoadjust white balance, aperture, and focus. After looking into cameras we decided to use the Microsoft LifeCam Studio camera. This camera allows the user to choose from multiple resolutions depending on the application.

Navigation Algorithm:

To start our algorithm we first read live video into Matlab. After reading this video in we wanted to pick out data that would help us navigate. This was done by detecting the color of the buoys (blue, green, red, white, and yellow). To do this we used the RGB color model and had as many systems as colors we wanted to detect. In each system we split the video into its three primary RGB colors (Red, Green, and Blue) and then subtracted the pure color from the video. By calculating the Euclidean distance of each channel we knew how close we were to the origin of the RGB cube. This output a grayscale image with the darker objects the color we were trying to detect. By thresholding this image, we created a binary mask where we could run morphological operations to clean out unwanted noise.

The algorithm then calculated the centroid of each blob it saw in the image. It started by looking for the lowest (closest to the bottom) red centroid in the image. If it saw this, then we told the boat move so that the red centroid stayed in the area of the image that we wanted. If it did not see red, but instead saw green, it carried out a similar process. There were obstacles in the course that were marked by yellow buoys. When our boat saw these it tried to navigate the boat along the route with the maximum space. We also added exceptions when the boat saw blue buoys, white buoys, or no buoys. Our ability to distinguish between buoys and background is illustrated in Figure 4 below.

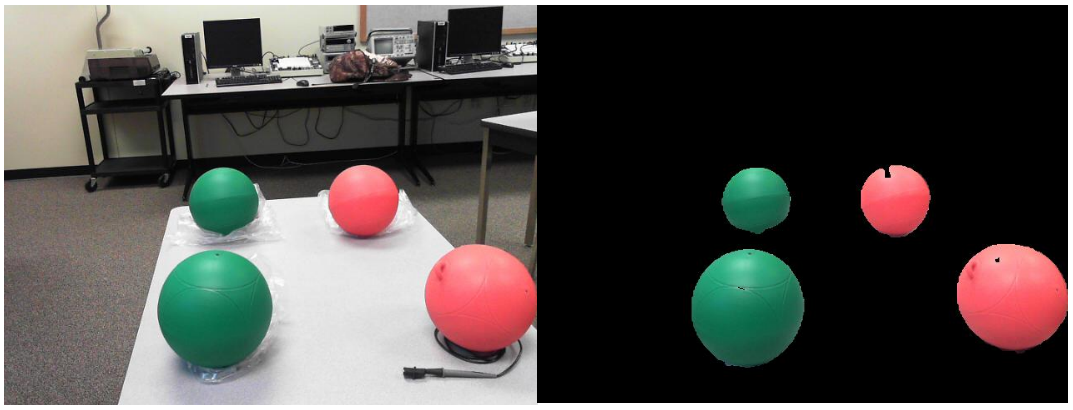

Figure 4 Illustration of distinguishing of red and green buoys from background.

Interfacing from Computer to Thrusters:

Depending on what we see through the camera, we send different values to our microcontroller. We used an Arduino microcontroller and programmed two output pins on it to send values for our left and right thrusters. The Arduino sends the signals using pulse-width modulation to our motor controller. We used a Sabertooth 2x25 because of its capability of handling both of our motors. The motor controller will then send out voltages between 1 and 24 volts which correspond to the speed of the thrusters. One advantage of using the Seabotix BTD150 thrusters is that they can go in forward and reverse. Therefore if we needed to make a sharp turn, the motor controller can tell one thruster to go full forward, and the other full reverse. While max reverse is not the same speed as the max forward, this still helped up complete difficult maneuvers.

E-Stop and RC/Auto Switch:

A safety requirement and a useful feature for our autonomous vehicle was making an physical and remote E-stop. While testing our ASV we found out that many times the algorithm would not work the first time as expected. Sometimes the boat would just sit there and not go anywhere, or worse case, it would take off full speed in a direction it was not supposed to go.

When this happens, it is very difficult to catch up to the vehicle. As a result a remote E-stop and physical E-stop (if we could get close enough) was a requirement. For our physical E-stop we used a Honeywell push pull switch which was wired in series with our batteries. When the button was pressed it would lock down all propulsion systems. This is a physical feature that makes it safe because it has to be reset in order for the power to be restored to the vehicle. In order to reset the button it needs to be twisted until it pops back up.

A remote E-stop was not as simple as a physical button. Since we used a 6-channel radio controller for our RC mode while driving our boat (we only used two channels when driving it), we decided to use one of those channels as our kill switch and at the same time be able to switch to RC mode. By using an empty channel and attaching a servo motor to the receiving pin we were able to use this motor to physically move a switch on the boat. When the button on the remote was pressed, a wheel would spin on the boat changing the input of the motor controller from the Arudino microcontroller to our radio remote. As long as the button for the servo was pressed, the boat would no longer be in autonomous mode, but would be in a state where it would not move unless we told it to with the remote. This made it easy to bring our boat back to shore while not physically being out on the water.

Concluding remarks:

Overall, we were very satisfied with the work and progress that we made this year. We hope that our boat will perform well and be successful in navigating through the channel. We plan to build off of this year and be able to attempt all challenges in next year’s competition.