Cedarville University RoboBoat

Tyler Chan, Krista Kroninger, Jimmy Myers, Tim Swanson (Team Leader)
Samuel Dallas, Josiah Franklin, Kyle Girtz, Jonathon Power, Jimmy Smyly
Team Advisor: Timothy Tuinstra

ABSTRACT

Building on the success of last year, the Cedarville Roboboat team is prepared to compete in the 5th Annual International RoboBoat Competition. The completely new hull is a pontoon fishing boat which is 60 lbs. lighter than the previous entry. Improvements to the vision software include the use of the HSV color space for color segmentation. This year a new proportional-derivative controller has been designed based on compass. This allows for more reliable control and heading following. New to the boat is the added additional feature of GPS control used to navigate the boat to the general area of the challenges. The team was able to develop an effective autonomous surface vehicle without the more advanced vision systems of Lidar or stereo vision. Nearly all of our software is written in Matlab.

1. PLATFORM

1.1 Pontoons

Last year's PVC pontoon design was too heavy and the integrity of the pontoons was being compromised after a year of usage. As a result of these drawbacks, a new team objective this year was to keep the weight of the total system under 100 lbs. This goal was surpassed with a total platform weight of approximately 80 lbs.

A pontoon style platform was chosen for maximum stability on the water. This year two Echo Fisher pontoons made by Venture Outdoors were used as pictured in Figure 1. These pontoons are made out of linear low density polyethylene (LLDP) which is much lighter than PVC. Each pontoon is 48" long by 12" wide by 10" high and weighs only 11 pounds. Originally the pontoons were meant to be used as fishing boat, with a seat holding the two pontoons together. This seat made the pontoons over three feet wide when assembled which was greater than the competitions width limit. This problem was solved by cutting out and removing a section from the middle of the seat and then plastic welding the seat back together. The result was a narrower seat that made the boat less than three feet wide. Another convenient feature of the Echo Fisher pontoons is a 4" diameter access port on top of each pontoon. This was an important factor for the mounting of the thrusters.

Figure 1. Echo Fisher Pontoons

1.2 Propulsion

The propulsion system consists of two Seabotix BTD-150 thrusters which were used last year. These thrusters were chosen because they are extremely small (6.927"x3.72"x3.673"), lightweight (1.58 lbs each), and easy to interface. With the maximum voltage applied, the thrusters can each produce 4.07 lbs of thrust per motor. The thrusters are brushed DC motors enclosed in a watertight housing. Each thruster was mounted directly underneath the center of each pontoon. Using this design zero-radius turns were possible using differential steering, an important feature to get through narrow spaces while navigating. This is the similar to the design that was used last year.

1.3 Electronics Enclosure

The electronics enclosure is a completely new design from last year's custom Plexiglas and Starcraft Marine plastic box. A watertight enclosure was chosen to house the expensive and important electronics. After much research, the Nanuk 945 protective case was chosen because it was relatively light (compared to Pelican Cases), watertight, and the interior dimensions were large enough for all of the electronics.

However, the Nanuk 945 does not have any output connectors through which wires can be routed. Watertight connectors were considered but these connectors were expensive, interference between signals on wires could be an issue, and soldering was necessary which would require cutting into shielded cables of the webcams, thrusters, and servos. Instead, simple 90-degree elbow plastic conduits (with threaded ends) were chosen. The diameters of these were just wide enough to fit USB plugs. By drilling into the box and using the provided threaded ends (with o-rings) a water resistant connection was created. It would be very difficult for water to make its way into the enclosure since it would need to go directly up the elbow which is not directly over the water.

2. ELECTRICAL SYSTEM

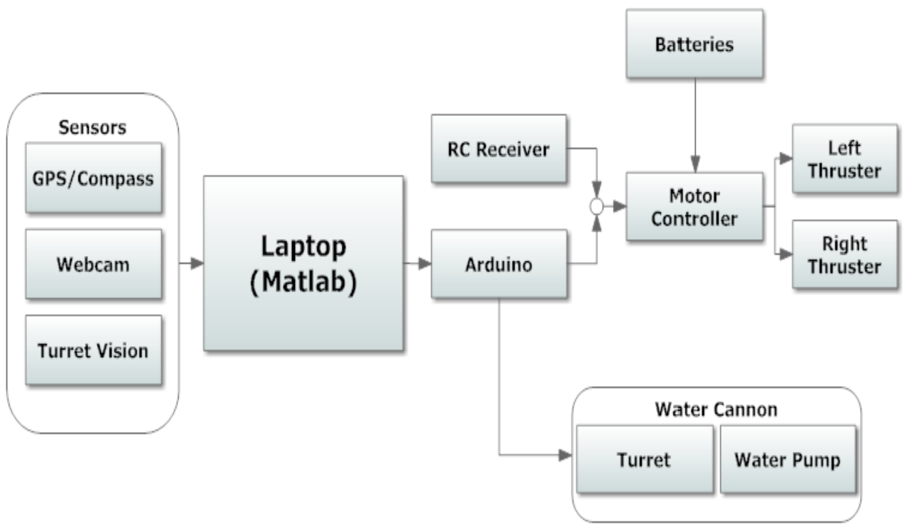

Figure 2. System Diagram

2.1 Sensors

2.1.1 Global Positioning System (GPS)

Utilizing the LS20031 GPS receiver the boat will know generally where it is in the competition course. While this GPS is not as accurate as hoped, it still can provide the approximate area and help to navigate to spots in the lake where the vision system can then take over. It is a Wide Area Augmentation System (WAAS) GPS receiver with an accuracy of $\pm 3\ m\ (9.84 ft)$. It is a Satellite Based Augmentation System (SBAS) which uses a system of ground stations and satellites to transmit differential corrections [3]. A WAAS enabled GPS receiver was chosen because it could provide greater accuracy than a normal GPS and was much less expensive than the more accurate Differential GPS alternatives. It also includes a useful LED indicator to let the user know that GPS is successfully receiving data from the satellites.

2.1.2 Compass

The Devantech CMPS03 Magnetic Compass Module has 0.1 degree of resolution and 3-4 degrees of accuracy. The headings are output via I^2C and the device is connected to the laptop by a USB port. The Devantech compass' 19200 baud rate allows reception of data from the compass in real time, which is useful for navigation. The compass is a key part of the design because it is used to follow a desired compass heading.

2.1.3 Webcams

The Microsoft Lifecam Studio Webcam was chosen because the majority of the team has used this webcam before and has interfaced it successfully with Matlab. This webcam has multiple RGB resolution settings. By changing a single line of code the resolution can be changed. For the final product we decided that a resolution of 160x120 was sufficient. The white balance, exposure, hue, and saturation can all be manually set using one line commands in Matlab which also helps the image processing given different lighting situations. Two of these webcams were used so that different views may be used for navigation and challenge completion. The navigation camera is tilted down more to get less noise from the shore and has a polarizer in order to take out some of the reflections on the water. The challenge camera is set up in a way so that it will have a better view of the challenges.

2.2 Laptop (Matlab)

The Dell Inspiron N4030 laptop is the main processor for the system. The laptop runs Matlab and performs all of the vision processing and data processing. The ease of programming in Matlab with built-in functions allowed significant progress to be made without writing basic functions and advanced blob analysis functions included in the Image Processing Toolbox.

2.3 Microcontroller

The laptop sends data out to control the thrusters. However, intermediate electronics are needed for the laptop to control the thrusters. The Arduino microcontroller converts numerical values to pulse width modulation (PWM) signals for the motor controller. The Arduino microcontroller is very popular and has open source software. Therefore, there are many forums and help sections that were referenced when programming the microcontroller. With this and the user friendly programming environment/language the Arduino has been integrated into the system to carry out the PWM conversion. This microcontroller hooks directly to a USB virtual COM port in our laptop.

2.4 Motor Controller

The Sabertooth 2x25 motor controller has two output channels that are capable of outputting a max of 25 amps. When looking at motor controllers, one was desired that could output to both of the thrusters and allow for differential steering, which means the thrusters can operate independently of each other. This motor controller met the voltage and current specifications and several members of the engineering faculty were also familiar with this motor controller. It also has built-in overcurrent and thermal protection. To be able to switch between autonomous and remote control mode a motor controller was needed that could also work with the receiver for the remote. The Sabertooth has a 5V power supply which can power the remote control receiver. A servo is used to throw a switch that is controlled by a channel on the remote control so that the input to the motor controller can switch between autonomous and remote mode.

2.5 Remote Control

This year, a Futaba 7C 2.4 GHz remote control was chosen. The previous remote control, the Vex 75 MHz, worked well during the testing stage in Ohio. Then at competition last year there were problems controlling the vehicle from the shore. After troubleshooting it was found that in a more populated and commercially busy area the frequency being used was being jammed by other RF signals in the area.

The Futaba 2.4 GHz system utilizes spread spectrum technology to increase the reliability of the communication with the vehicle. Instead of staying at one or two frequencies around 2.4 GHz, which is what a traditional remote control does, it uses frequency hopping. The transmitter and receiver are paired to achieve this feature. The receiver has two antennas for built in error checking. The receiver will choose the data from the antenna that has the most accurate reading and has the fewest errors.

2.6 Water Cannon (The Cheater's Hand Challenge)

2.6.1 Water Pump

A Rule Charge N' Flow portable pump this year was chosen this year. The entire kit including the pump, hose, nozzle, rechargeable battery, case, and other accessories all weigh about 4.3 lbs. It was desired that the pump be as light as possible to help keep the final boat weight low. It also has a maximum range of about 6 feet, which is better than the 5 ft requirement specified in the design requirements. Originally, the pump automatically turned on when powered. By sending the power to the water pump through an Omron G5V-1-DC6 relay, the pump was able to be controlled using the Matlab program.

2.6.2 Turret

It was decided that servos would be the easiest way to implement a turret mostly because of the team's familiarity with them. A kit was purchased that included two Hitec HS-322 servos and servo horn attachments intended to give a camera "pan and tilt" capabilities, but simply attached the hose instead. Servos are easily interfaced with the Arduino microcontroller. The complete water cannon can be seen in Figure 3.

Figure 3. Water Pump and Turret Together to form Water Cannon

2.7 Hockey Puck Retrieval Rover (The Poker Chip Challenge)

2.7.1 Rover

To retrieve the hockey puck, a small, scouting robot was built with “antennae” and a lever arm covered in Velcro, as seen in Figure 4. The frame of the robot is made of starboard, a lightweight, strong, easy to cut plastic composite. The robot is propelled by two continuously rotating servos. It was desired that the robot not fall off of the dock so two extended IR distance sensors were installed on the front. These point at the ground and tell the robot whether it is safe to go forward. The puck is sighted by a third IR distance sensor that is mounted under the rover. This long-range sensor is parallel to the ground and very low to the ground so that it can see the edge of the puck. A third high-torque servo slams the Velcro covered lever arm onto the puck in order to lift it off the ground. All of the sensors and servos are controlled by a PIC18F45K22 microcontroller set on a custom built PCB that was used for a past project.

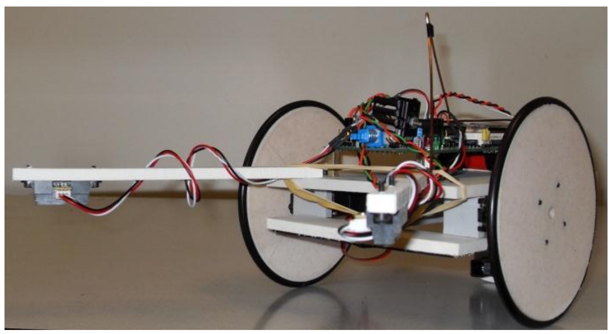

Figure 4. Hockey Puck Retrieval Rover

2.7.2 Rover Deployment Crane

Our deployment structure for the rover consists of a crane mounted on the boat and a bumper for determining when the boat hits the dock. The complete assembly can be seen in Figure 5. The bumper is simply a pull-up resistor contact switch. The crane has two DC motors, one for rotation and one as a winch. These motors are controlled by an H-bridge motor controller that receives its commands from the same microcontroller previous mentioned above. (Sec 2.3)

Figure 5. Rover Deployment Crane

3. VISION

For vision system success an algorithm was needed which could effectively separate red, green, yellow, and blue buoys from the rest of the scene. Last year, images were processed in the RGB (red, green and blue) color space. In this color space red, green and blue color thresholds were determined. In addition, a gray scale filter was employed that removed most of the water pixels from the segmented image. Unfortunately, this color space was not robust enough to allow the segmentation of yellow easily from the rest of the scene, so this year the HSV (hue, saturation and value) color space was used.

Using the test image from the lake (Figure 6), pixels were selected that belonged to the red, green, and yellow buoys and plotted them in the HSV color space (Figure 7). From plotting the pixels of the test image in the HSV color space, it was found that the different color buoys could be easily separated using just the Hue and Saturation components of the HSV color space. Figure 7 also shows that the red, green, and yellow regions are separable by simple rectangles in the HSV color space while the same regions required much more complicated shapes to properly segment in the RGB color space. Initially, the HSV color space was not used because of how slow the conversion from RGB to HSV space was in Matlab, but later, a native .mex file was found online that sped up the algorithm significantly. With the RGB color space, 10 frames per second were achieved while with HSV achieved almost 9 frames per second.

Figure 6: Test Image

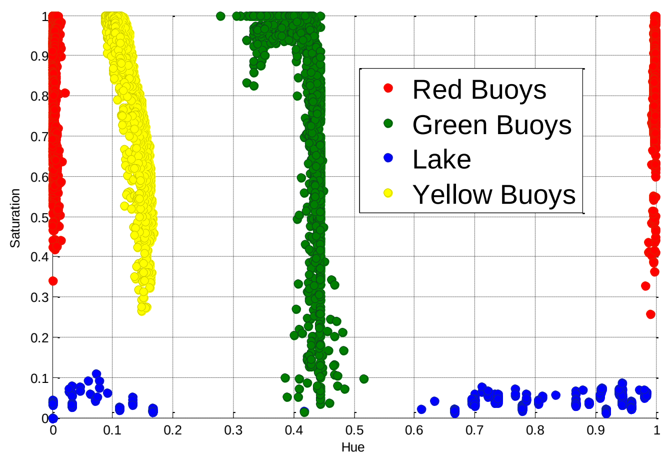

Figure 7: HSV Color Space Pixels in Test Image

4. NAVIGATION

4.1. PD Controller

Since it was found that our thrusters gave different output thrusts given the same input voltage, a controller was developed that compensated for one thruster being different than the other thruster. To make this controller, the left thruster was set to a specified value and the other thruster was set to that value plus or minus a proportional value and derivative value. Initially, a proportional controller was used that took the error of the current heading compared to the desired heading and added some multiple of that value to the output thrust. Unfortunately, a simple proportional controller caused the boat to oscillate considerably and took longer than a minute to settle out (see Figure 8).

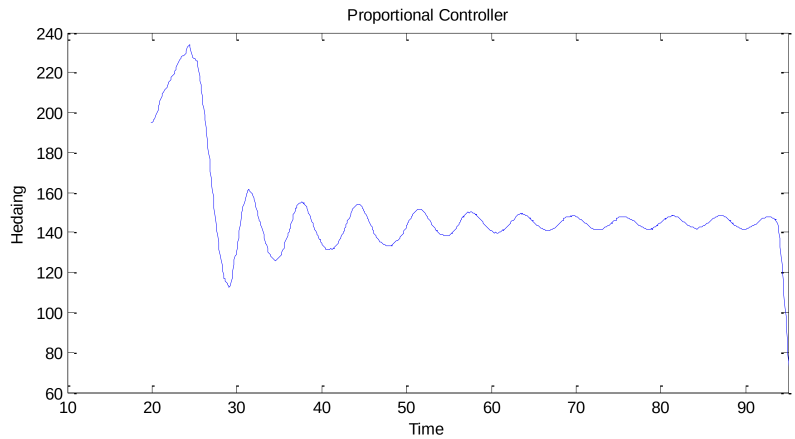

Figure 8: Heading Versus Time for the Proportional Controller

To remove the oscillations, a PD controller was implemented that added an approximation of the derivative of the error multiplied by a constant to the thruster value as well as the proportional value discussed earlier. By using the PD controller, the results shown in Figure 9 were achieved and it was proved that the erratic thrusters could be fixed using this controller. In the end, the PD controller allowed us to program an extremely reliable compass following algorithm.

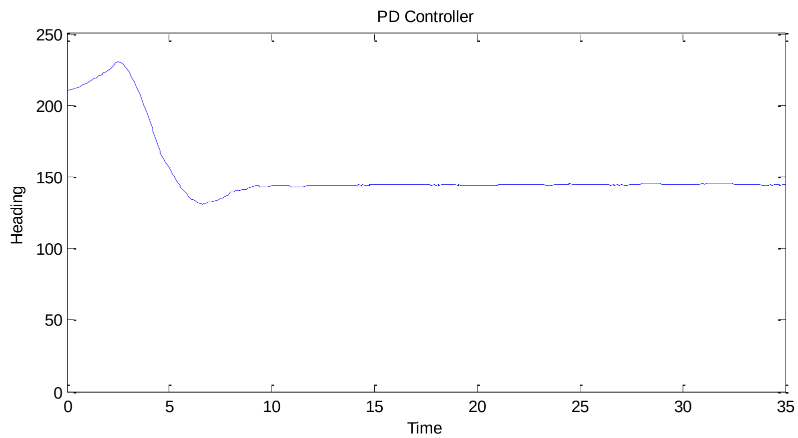

Figure 9: Heading Versus Time for the PD Controller

4.2 Speed Gates

To navigate the speed gates at the beginning of the course, compass heading information was used so that the boat navigates the speed gates in the fastest time possible. A Matlab function was developed that causes the boat to go in the direction of a given compass heading using the PD controller previously discussed. Using this function, the boat is simply pointed in the correct initial direction. When the boat detects the speed gates, it will use the same logic as the channel navigation algorithm to successfully navigate through the speed gates.

4.3 Buoy Channel

To navigate a channel of red, green and yellow buoys the webcam and compass are used. The segmented image of the buoys (discussed in previous section) on the lake is used to make logical decisions to navigate the channel. Depending on the locations of the red, green, and yellow buoys, our algorithm reacts and changes course accordingly.

4.4 Challenges

To navigate to the challenges, after completing the channel, the GPS and compass are used to navigate waypoints. The desired waypoints are geographical latitude and longitude coordinates near the challenges and other choice locations. The true compass heading and the distance from the next waypoint are calculated from the desired waypoint and current waypoint. A magnetic declination correction is applied to the true compass heading to convert it into a magnetic heading. This is now the desired heading that is the new input to the PD controller. Figure 10 below illustrates the boat navigating from waypoint 1 to waypoint 2. Once the boat is inside the red circle, the boat is considered to have reached the waypoint. The red circle represents the margin of error threshold, which is currently set to 9.84 ft (the accuracy of the WAAS receiver). Once a waypoint is reached it goes on to travel to the next desired waypoint that is stored in the desired waypoint array.

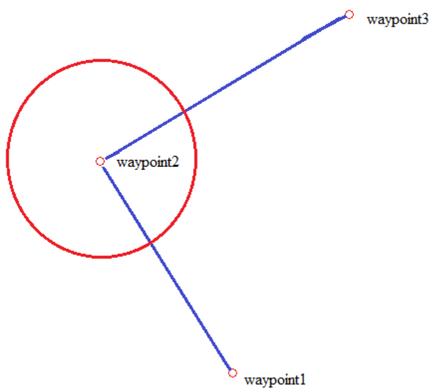

Figure 10: Waypoint Navigation

Every time there is a new waypoint, the desired compass heading and distance to the next waypoint from the current location is recalculated. Wind will not be an issue because the algorithm is based on the current position of the boat, not solely on magnetic headings which would show no error if the boat drifted.

5. CHALLENGE LOGIC

5.1 Water Cannon Vision

The algorithm for aiming at the target begins by switching to the water cannon camera and turning on the water pump. It then proceeds to identify the white board and look for the target on the board using image processing techniques. The servos that make up the turret system are sent PWM values based on the location of the target in order to aim. Once red is detected over the board, which should be the flag indicating that the challenge has been completed, the vision switches back to the main camera, the water pump is turned off, and the robot moves on to the next task. Figure 11 shows an example of a challenge as seen through the GUI.

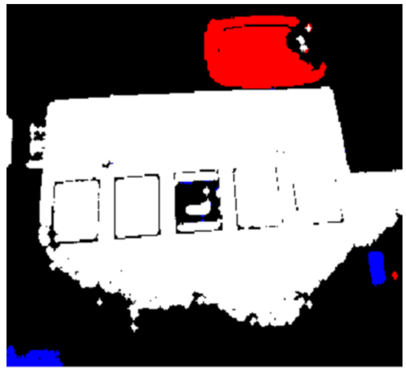

Figure 11: Screenshot of Challenge Detection

5.2 Hockey Puck Retrieval Rover and Deployment Logic

The boat does not directly communicate with the rover. When boat has made contact with the floating dock, the boat bumper sends the signal to deploy the rover. The crane rotates to place the rover over the floating dock and winches the rover down. Then the boat will wait an allocated amount of time before winching the rover back in, even if it had not found the hockey puck yet.

Before this challenge, the rover will sit on the boat with its antennae over the edge during its idle state. Once the Rover deployment crane lowers the rover, the rover will wait until it sees the ground before beginning the puck search algorithm.

The robot finds the puck by a simple algorithm: alternate going forward a couple feet and rotating in a circle. If the puck is sighted the robot locks onto it, and if the edge of the dock is seen the robot backs up and rotates away from the edge. Once the rover has the puck it goes into idle state and waits for the boat to winch it in. This method required the least amount of communication and sophistication on both sides while still completing the challenge and not dropping the rover in the water.

6. CONCLUSION

After finding some success last year, this year we built on what worked and made significant changes. The thrusters were maintained and much of the electronics but the pontoons, electronics box, and remote control system were upgraded while adding GPS capabilities and systems to attempt the water cannon challenge and the amphibious landing. The code this year has been significantly modified and some parts written completely from scratch. The hope is that this year's entry from Cedarville University will be highly competitive.

7. REFERENCES

[1] *Baumer Group*. N.p., n.d. Web. 12 Dec. 2011. <http://www.baumer.com/services/knowledge-technology/background-knowledge-ultrasonic-sensors/en.html>.

[2] "The Navy Unmanned Surface Vehicle (USV) Master Plan." *www.navy.mil*. N.p., 23 July 2007. Web. 10 Dec. 2011. <http://www.navy.mil/navydata/technology/usvmppr.pdf>.

[3] "What is WAAS." *Garmin*. N.p., n.d. Web. 12 Dec. 2011. <http://www8.garmin.com/aboutGPS/ waas. html>