Nautical Autonomous System with Task Integration

Authors: Zackary Knoll, Steven Blass
Advisors: Dr. Gary Dempsey & Mr. Nick Schmidt

Department of Electrical Engineering
Bradley University

Abstract

This paper discusses the design and functionality of the Bradley University Nautical Autonomous System with Task Integration (NASTI). The autonomous vehicle is a catamaran-style boat that was designed to compete in the Association for Unmanned Vehicle Systems International Foundation and Office of Naval Research's 2013 Roboboat competition. The project is an expansion of the Roboboat 2012 entry by Bradley University. The 2012 platform was capable of autonomously navigating the buoy channel using a webcam for computer vision, a 32-bit ARM embedded microcontroller (BeagleBoard XM) for data processing and decision making, two main motors for forward and reverse mobility, and four side thrusters for rotational and lateral mobility. The boat has two pontoons for flotation and was powered by 6V and 12V lead-acid batteries, mounted in the pontoons. A flat deck sits atop the pontoons and supports the processors, sensors, and the power electronics components.

In 2013, the competition shifted emphasis away from channel navigation by awarding more points for challenge completion and awarding point multipliers for channel navigation. Also, teams now receive the GPS coordinates of each challenge station. To compete in the additional challenges, the sensing capabilities were expanded to include a planar Light Detection and Ranging (LiDAR) sensor, a GPS receiver, and an electronic compass. The BeagleBoard was upgraded to an Intel i3-2120T 2.6GHz Dual-Core processor, and the existing software architecture was restructured to support multi-sensor input. The batteries were upgraded to 12V lithium iron phosphate ($LiFePO_4$) cells which have twice the energy density of lead-acid cells.

Introduction

In 2012, the student members of Team NASTI (Nautical Autonomous System with Task Integration) proposed an autonomous system based around a catamaran platform, as the basis for a senior capstone project. The robotic platform competed in the AUVSI Foundation and ONR's 5th International Roboboat Competition and took 9th place. In 2013, the system has been improved with the intent to compete in the 6th International Roboboat Competition. Last year's project focused on navigating the buoy channel. The project this year will focus on the addition of new sensors that will be necessary to perform the extra challenges. Additionally, a new software architecture was developed to allow for easy hardware expansion and algorithm development. The system is shown in Figure 1 and an overall block diagram can be seen in Figure 2.

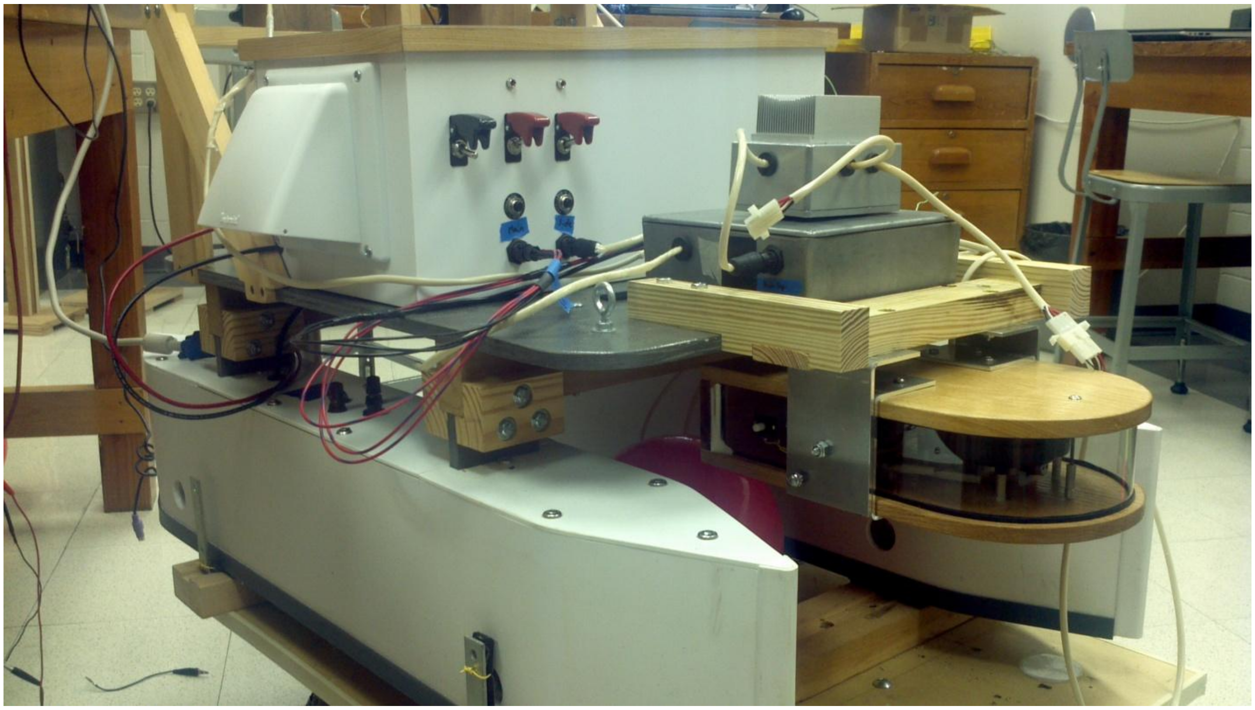

FIGURE 1: NAUTICAL AUTONOMOUS SYSTEM WITH TASK INTEGRATION

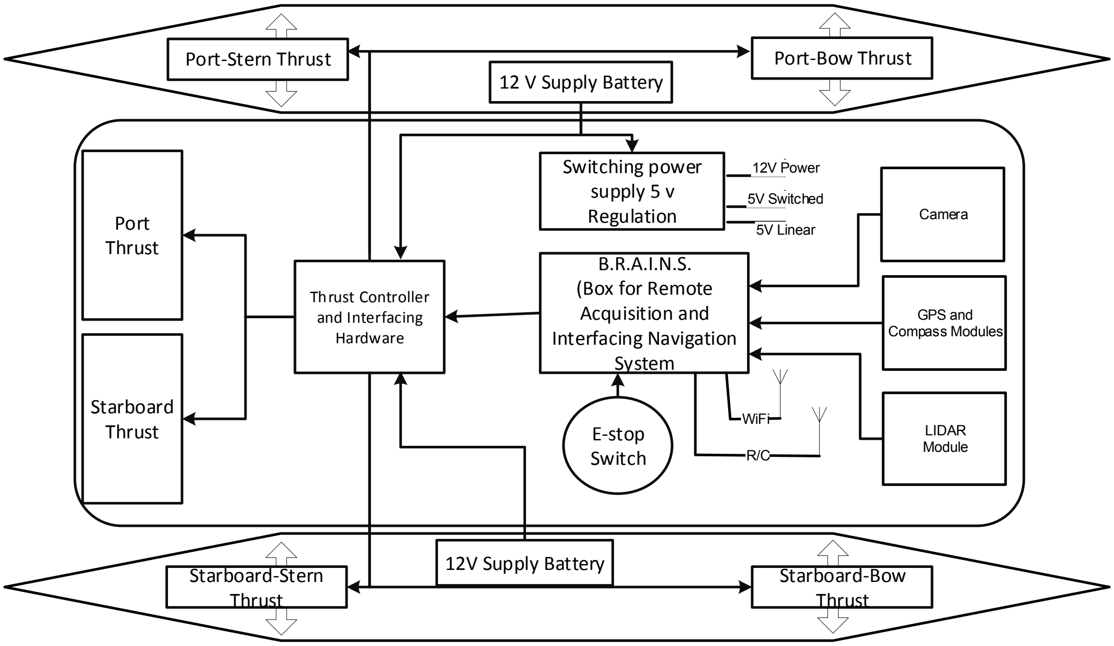

FIGURE 2: SYSTEM BLOCK DIAGRAM

Platform Design

The 2013 platform is a modified version of the 2012 platform. It has been expanded to accommodate additional hardware and sensors. The platform consists of three major components: pontoons, drive motors, and data sensor/equipment mounts.

Pontoons

Last year's team chose a catamaran style platform with two pontoons for floatation, seen in Figure 1. This design was chosen because it provides additional stability in a turbulent environment. This will help prevent pitching and rolling which would adversely affect certain data sensors. Each pontoon houses a battery module as well as two side thrusters used for additional mobility. These side thrusters allow for fine tuned movements during navigation, such as rotation in place, and strafing – maneuvers traditional designs lack.

Drive Motors

Forward and reverse thrust is provided by the two main drive motors located at the back of the vehicle. Using differential drive, these motors can maneuver within the channel while still moving forward. They are positioned so that they are angled inwards at equal and opposite angles to assist in turning while allowing the vehicle to travel straight forward while both are running. This reduces the overall forward thrust; however the team felt that this was necessary to improve the turning response compared to the forward setup utilized by the team during the 2012 competition.

Electrical Systems

This year, the primary processer was improved. The hardware was upgraded and now has significantly more processing power in order to accommodate additional inputs and processing algorithms. Also, a suite of new sensors was added to allow for participation in the additional challenges, including a LiDAR unit, a GPS, an electronic compass, and a thermal sensor.

Batteries

The boat is powered by Lithium Iron Phosphate ($LiFePO_4$) batteries. $LiFePO_4$ batteries have similar charging and discharging methods to generic lead-acid batteries, and were used as drop-in replacements for last year's power supply. $LiFePO_4$ batteries have about double the energy density of lead-acid batteries. These batteries are comprised of four 3.2 V $LiFePO_4$ cells, each with 15 Ah of charge. There are two banks of batteries for a total of 30 Ah of charge. This provides enough charge to run the boat for around 2 to 3 hours of continuous operation. Also, there is over double the total charge while maintaining the same weight as the previous year.

Motor Hardware

The purpose of the motor hardware is to translate the motor pulse-width modulation (PWM) commands that have been received from the primary processor into signals that can drive the motors.

Side-Thruster Hardware – As mentioned previously, there are four side thrusters that assist in rotational and lateral movement. Cascaded transistor (BJT) circuitry was used to interface with the motors in order to supply the proper amount of current to the motors (around 1.5A).

Differential Drive Motors –In 2013, the motor voltage for the main thrusters was increased from 6V to 12V. This was done in order to improve mobility and speed compared to the 2012 platform. With this addition, the current through the switching transistor of the interfacing circuitry became too high to use a BJT. The BJT circuitry was removed and replaced with low resistance power MOSFET circuitry, which dissipates significantly less power.

Navigation and Data Sensors

An ATmega1284 processor interfaces the GPS, compass, thermopile array, and primary processor. It is an 8-bit microcontroller with 2 USART interfaces and an I^2C interface. One USART communicates with the GPS receiver and the second USART communicates with the primary processor. The ATmega1284 acts as a master on the I^2C bus; currently there are two slaves: the compass and the thermopile array sensor. The GPS receiver has a factory determined update rate of 1Hz and a precision of ±3 meters. The compass returns a single byte indicating the heading relative to magnetic north. The thermopile array returns a 16x4 matrix of temperature values relative to the ambient.

Waypoint Navigation – The rules of the competition state that each challenge will have the GPS coordinates provided. The GPS will be used to acquire the absolute location while the compass will be used to determine the absolute heading.

Once the boat sets its desired location, it obtains its current position from the GPS receiver. A vector can be produced between these two coordinates, and the desired heading is computed with basic trigonometry. Then the compass is used to determine the heading with respect to magnetic north. The current heading, desired heading, and vector magnitude are used to control the motors.

The next step is to create a feedback loop that controls the motors based off of these values. The main thrusters drive forward at a speed proportional to the magnitude of the vector between the current location and the desired location. The side thrusters are controlled based off of the error between the current heading and the desired heading.

LiDAR Module

A light detection and ranging (LiDAR) sensor was added to the boat this year in order to assist in localized navigation. The sensor is a 2-dimensional LiDAR that provides range data for 360 degrees on a horizontal plane around itself. Utilizing the trigonometric and statistical properties of related points, basic feature extraction can be performed. The LiDAR is used for wall detection and dock detection for the catch the ball challenge.

Wall Detection – The primary objective of adding the LiDAR sensor is to be able to perform feature extraction to supplement the camera. Features such as the shoreline tend to be irregular in pattern and color, so using the camera to detect these features becomes much more difficult. A well defined shoreline will look like a wall to the LiDAR.

If the boat is aligned perpendicular to a wall, then all LiDAR points that lie on the wall have the same perpendicular distance to the wall which can be calculated by equation (1).

$$A = r_i * \cos(\theta_i), -90 < \theta_i < 90 \tag{1}$$

Even with a wall present, the mean value of A of a range of points will have a variance associated with it due to electrical noise and imperfect alignment. Observing a low standard deviation indicates a wall is located perpendicular to the boat. Walls at angles other than 0 degrees can be searched for by renormalizing the coordinate system to a different angle. The dock detection algorithm works off of the same basic principle as the wall detection algorithm.

Secondary Processor

The secondary processor is responsible for determining the PWM duty cycle commands to send to the thrust controller. It receives three different sets of input: autonomous motor commands from the primary processor, manual motor commands from the RC Controller, and a halt command from the emergency stop button. The emergency stop button is required for safety reasons, and the RC controller allows retrieval of the boat after a run which does not return to the dock as well as providing another emergency stop.

The secondary processor continuously reads input from all three sources. If the emergency stop button has been activated, the secondary processor activates a relay which creates an open circuit between the batteries and the electronics. If the button is deactivated, the secondary processor will pass through either the autonomous or the manual commands based on the state of the RC override switch.

Primary Processor

The primary processor is responsible for executing advanced data processing algorithms and making decisions based on these results. The hardware consists of a second generation Intel Core-i3 at 2.6GHz, 4GB of RAM, a 128GB solid-state drive (SSD), and a mini ITX form factor motherboard. The operating system is CrunchBang Linux 10 "Statler". The hardware can be seen in Figure 3 on the next page.

In autonomous mode, the primary processor receives input from a camera via USB, the navigation processor via serial communication, and the LiDAR processor via serial communication. The primary processor communicates its commands to the secondary processor via serial communication as well.

The primary processor receives raw images from the camera via USB interface. Images are processed using OpenCV. The image processing subsection has several algorithms to select from at runtime depending on the boat's state. These include searching the image for circles, and searching the image for colored buoys in either the HSV or RG-Chromaticity color space.

Buoy Detection-HSV – This method of detecting buoys converts the image into the HSV (Hue, Saturation, Value) color space. Thresholds are applied to HSV image to detect a desired color. The resulting blobs are detected and interpreted as buoys.

The first step is to determine a horizon level. This is ideally where the edge of the pond meets the ground and sky in the image. For simplicity, the sky is estimated to be the top third of the image. The region of interest (ROI) is bottom two thirds of the image, and further processing only occurs in the ROI.

The image is then converted into the HSV color space. This improves detection under varied lighting conditions. The representation of a color in the RGB color space can change dramatically under different lighting conditions; however, the HSV color space representation changes only slightly. The saturation is the most likely to change as the buoy is a reflective surface and direct sunlight will almost completely flood a portion of the buoy in white color. A pair of threshold values is set for each of the three color channels. Each color uses a different pair of threshold values. The threshold operation is applied to the image, creating a binary image. Pixels within the threshold pair are colored white, other pixels are black. This image typically results in several blobs of the desired color.

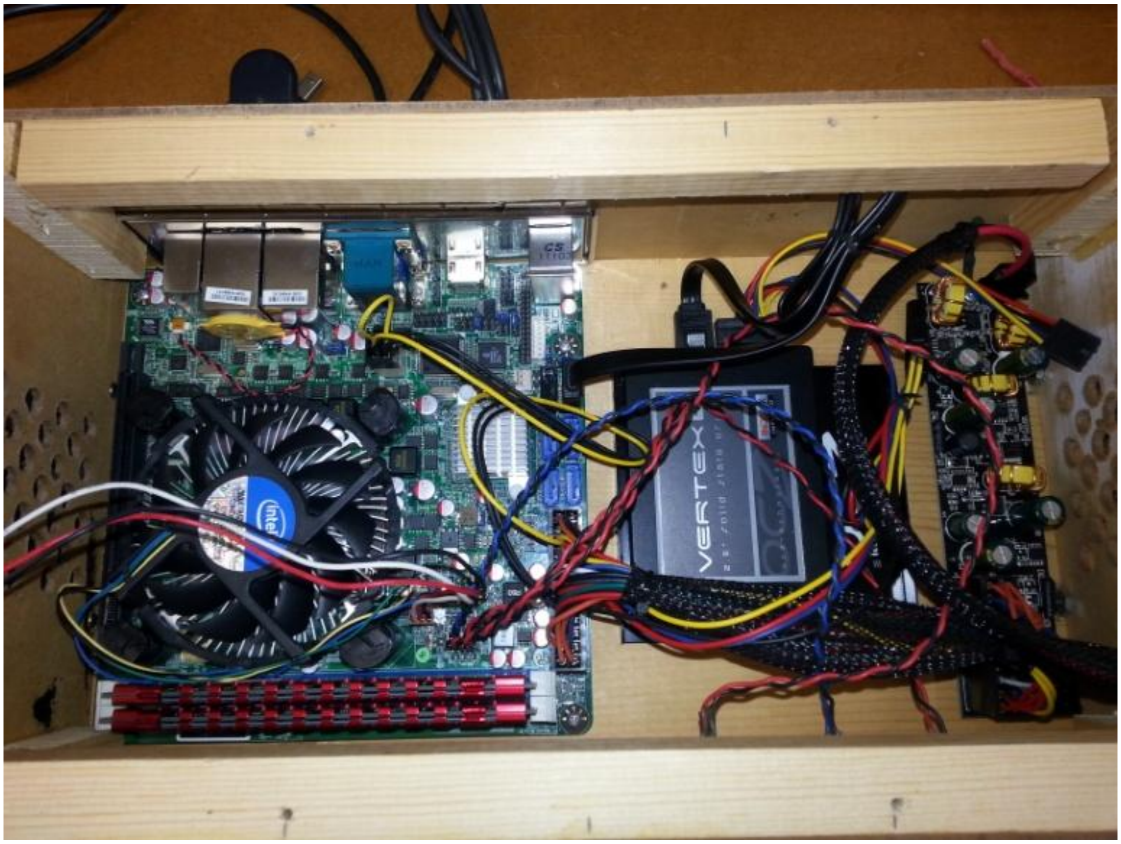

FIGURE 3: PRIMARY PROCESSOR HARDWARE

To determine the location and size of the blobs, the cvBlobs library – part of openCV – is used. The functions assume a blob is an area of connected pixels of the same color. Given a binary image, a function determines the bounds of all blobs. A vector of blob objects is returned. Blobs which match the expectation of a buoy's size are designated as buoys. A sample image is shown in Figure 4 on the next page.

Buoy Detection-RG Chromaticity - The RG-Chromaticity based buoy detection is designed to complement HSV based buoy detection. The HSV based algorithm has difficulty detecting red buoys in open sunlight. The RG-Chromaticity color space is more resilient to lighting conditions because it is two-dimensional and contains no intensity information. Each pixel contains the proportion of red, green, and blue, rather than the intensity of each. This algorithm was seen to be able to determine the locations of red buoys better than the HSV algorithm.

Hoop Detection Algorithm – The shoot the hoops challenge requires the autonomous platform to shoot a foam dart through a hoop located on the shoreline. The hoops are almost perfectly circular in nature. Using this property, a version of the Hough Transform known as the Hough Circle Transform will be used for detection.

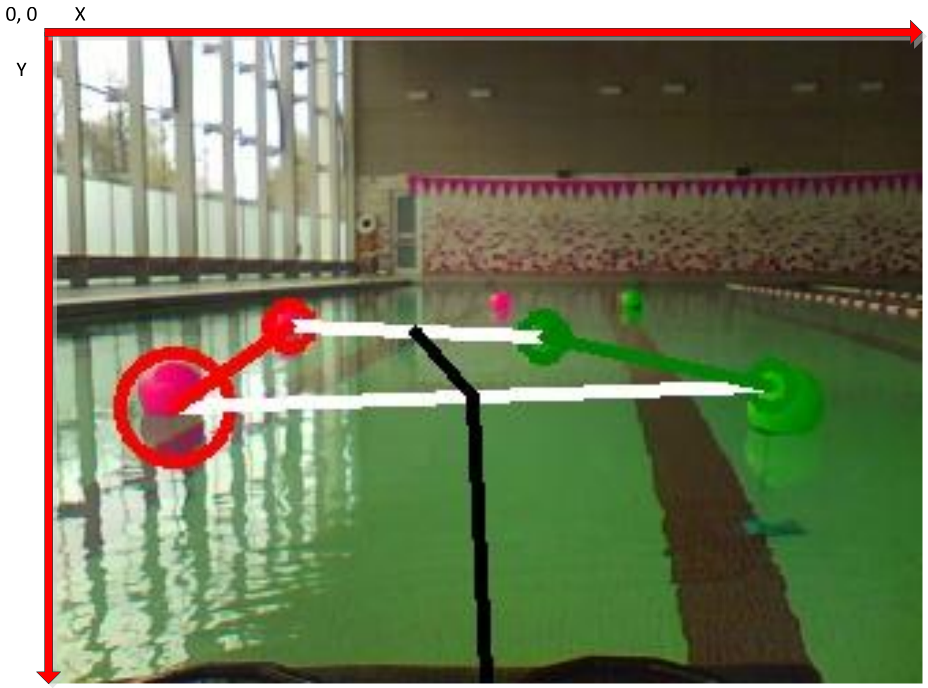

FIGURE 4: HSV BUOY DETECTION

To begin, a Gaussian blur is applied to the image in order to reduce noise. Then the image is converted from a color image to a grayscale image. The Hough Circle Transform is then applied to the grayscale image. This is done by using the openCV function "HoughCircles". This function returns a vector of circle objects that contain a center location and a radius. Further processing is performed to remove known errors such as abnormally large radii. An example can be seen in Figure 5 on the next page.

Software Architecture and Challenge Station Logic

One of the primary focuses of the project this year was expandability in both hardware and software. In order to facilitate this, the software architecture was rewritten so that new sensors could be added in alongside the previous ones without reconstructing the entire program logic. This was accomplished by programming a configurable finite state machine. Once the state machine was set up the logic for the challenge stages became embedded into the state machine configuration which has the ability to pick and choose what algorithms it wants to run.

Finite State Machine

The primary processor functions as an event-driven finite state machine. Events are triggered by any of the input sensor processing algorithms. Upon receiving an event, the current state checks it against its configured events. If the received event matches, the actions corresponding to that event are executed. The action will usually be a transition to another state, but it could also be a function call in the current state. Events which do not match the current state's configuration are ignored and no action is taken. State actions may change the data processing algorithms or turn them off entirely. This saves power and clock cycles when certain sensors are not needed.

FIGURE 5: HOOP DETECTION ALGORITHM EXAMPLE

The main loop of the primary processor is executed in 4 stages. The first stage is data acquisition. In this stage, the primary is polling the sensors for data and collecting all available information from active sensors. The second stage is data processing in which the active algorithm for each data set is executed. For example, the raw camera image may be processed to find buoys here. In the third stage, the primary processor makes decisions. This uses the processed data to plan a path and then determine commands to send to the secondary processor. The second part of the decision-making stage is checking for new events and executing the appropriate actions. The fourth stage relays the commands to the secondary processor.

Channel Navigation

The channel navigation algorithm uses the buoy detection algorithms results as input. Once buoy color and location is known, the red and green buoys are paired into 'gates'. The algorithm is designed to move sequentially through the center of each gate, forming the desired path. This is illustrated in Figure 4 on page 7.

The path section between the bottom-center of the frame and the nearest gate is used as input to a closed-loop proportional plus integral (PI) controller. This path section will hereon be called the control input. This algorithm assumes the camera is centered on the boat and the bottom of the image is level with the front of the boat. Thus, to pass through the gate, the control input should be perpendicular to the bottom of the frame and reduced to zero length. The length of the control input determines nominal speed, and the slope determines the intensity of the turn and the turning direction. The slope of the control input deviates more than the length of the control input between frames.

This algorithm terminates when reaching the blue buoy at the end of the channel, or when an obstacle is detected in the channel.

Obstacle Avoidance – The center of the buoy channel contains several yellow buoys designated as obstacles. The obstacle avoidance algorithm is activated when a yellow buoy is detected between the boat and the nearest gate.

A line from the bottom of the image to the yellow buoy is determined, hereon called the yellow line. The center of the bottom of the frame is designated the origin of the image. Angles relative to the origin are used to create an avoidance gradient. The peaks of the gradient are 0, 90, and 180 degrees. It is desirable to keep the yellow line in one of the two valleys located at 45 and 135 degrees. The avoidance gradient peak at 90 degrees prevents collision and the peaks at 0 and 180 degrees prevent excess deviation from the course. The slope of the yellow line becomes the input to a proportional controller.

The algorithm terminates when a yellow buoy is no longer visible.

Shoot the Hoops Challenge

Once this challenge begins, the state machine signals the subsystems to begin using the GPS and compass to navigate to the desired waypoint. Upon arrival, the GPS interfacing software signals the finite state machine that it has arrived. After this, the LiDAR is used to determine our orientation to the shoreline and rotate us so that we are facing perpendicular to the shore. The boat will then use the side thrusters to strafe while using the LiDAR to maintain the distance and orientation to the shoreline. During this time the image processor is running the Hough Circle Detection algorithm which will attempt to locate the hoops. When the algorithm detects the hoops, it sends an event to the state machine which will fire the on-board dart gun. The state machine diagram for the Shoot the Hoops challenge can be seen in Figure 6.

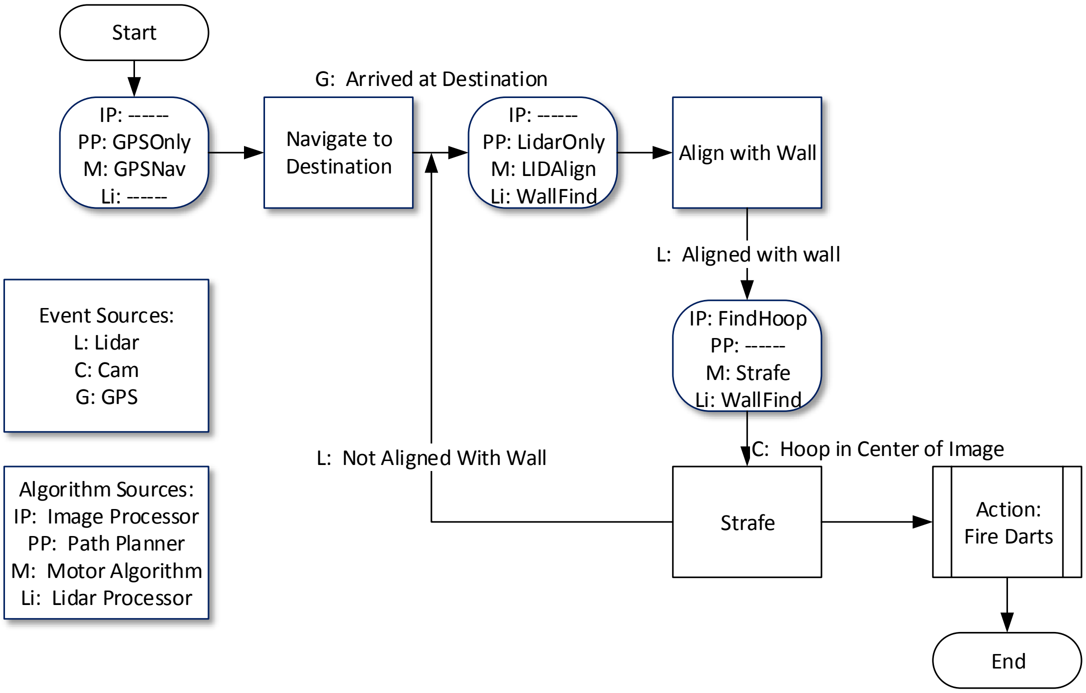

FIGURE 6: STATE MACHINE CONFIGURATION FOR THE SHOOT THE HOOPS CHALLENGE

Rock, Paper, Scissors, Lizard, Spock

Again, this challenge will begin by using the GPS and compass to navigate to the provided waypoint. Upon arrival, the LiDAR will be used to orient the boat perpendicular to the shore line. A thermopile array sensor is then used to produce a field of view of 16x4 pixels in size based on temperature with respect to the ambient. Centering the major heat source will mean that the boat is aligned with the desired sign. Since we will be able to determine the approximate location of the sign in an image from the webcam, the image will be cropped to a more manageable size. The cropped image will then be correlated with all 5 template images. The correlation that returns the highest result will be the one used. The software algorithm will determine the winner, connect to the network, and transmit the coordinates and winning sign.

Conclusion

This project saw the successful implementation of the navigation hardware and sensors. This allows for navigation to specific GPS coordinates that will be provided for the additional challenges. Additionally, the LiDAR sensor was also implemented which will allow for better localized movement and decision making once the boat has arrived at a challenge station. In order to facilitate expansion in future years, the hardware and software architecture of the primary processor was upgraded. The new hardware substantially increased the processing power, which will allow for the capability to add more sensors and processing algorithms. The software architecture now focuses on easy addition of processing algorithms while using an event-driven finite state machine to handle the high-level navigation processes.