Georgia Tech Savannah Robotics' ASV Victoria 2012 AUVSI and ONR's RoboBoat Competition

Valerie Bazie, Steven Bradshaw, Michael Bunch, Phillip Cheng, Lisa Hicks, Brian Redden, Carol Young

***Abstract*—Victoria, is an autonomous surface vehicle (ASV) dually used as a research and competition platform. Currently, the vehicle is being used as Georgia Tech Savannah Robotics' (GTSR) 3^{rd} entry into AUVSI and ONR's RoboBoat competition. The vehicle has been designed to detect and navigate through objects using an intertwined LiDAR-camera system, and perform mission specific tasks.**

Fig. 1: Victoria

I. Introduction

VICTORIA is Georgia Tech Savannah Robotics' entry into AUVSI Foundation and ONR's Fifth International RoboBoat Competition. While the overall appearance of Victoria has remained the same, her internal hardware differs tremendously from our previous entry. In this paper we will outline the mechanical, electrical and navigation specifics.

II. Mechanical Systems

A. Hulls

Significant improvements have been made to the recycled hulls from last year. Many leaks formed from the previous year's use. After extensive research, it was discovered that the main cause of the leaks was due to improper finishing of the hulls. To prevent such damages from reoccurring, the hulls were stripped, coated in marine paint, and a gel coat to create a water tight seal. The interior of the hulls were also coated with a spray-on rubberized utility coating for additional protection.

B. Interior

The major interior change was the switch from a wooden frame to an aluminum frame. The new design is shown in Figure 2. The electronics are attached to

Fig. 2: Frame

racks constructed of acrylic and aluminum angle pieces. Nylon bolts were chosen to bolt down the racks in order to prevent stripping of the aluminum threads on the frame. The hulls are held together using pieces of angle aluminum bolted to the frames. Similarly, the thruster mount, shown in Figure 3, is constructed from aluminum and is bolted to the frame. They are located in the center of the vehicle to allow the rear of the thrusters to be aligned with the back of the hull. The thrusters are located in between the hulls to prevent them from being damaged during handling.

Fig. 3: Thruster Mount

C. LiDAR Mount

Previously, the LiDAR was attached directly to the hulls causing it to move up and down as Victoria rocked. As a result of the vehicle's constantly changing pitch, the LiDAR was unable to detect the buoys. The solution was to make a stabilizing mount for the LiDAR as shown in Figure 4. Acetal sleeves were used as bearings for

Fig. 4: LiDAR Mount

the shafts that connect the servo to the LiDAR and the potentiometer's shaft. The potentiometer is driven by a shaft which is attached to a gear. The gear is driven by another gear attached to the servo shaft. The entire mount was constructed out of aluminum for its good strength to weight ratio and corrosion resistance. It was designed with slots to allow a forward, backward, upward, and downward movement of the LiDAR. To compensate for the change in pitch, as measured by the IMU, a PID controller is used to correct the angle of the LiDAR.

III. Electrical Systems

Victoria is powered using two 25.9V lithium polymer batteries for control systems, two 12V lead acid batteries for propulsion, and six D-cell alkaline batteries for the backup systems. For safety purposes, all power is routed through fuse blocks and a master cut-off switch. For emergency telemetry and control, a pair of Xbee 900 Mhz radios and Arduino embedded controllers have command over power, thrusters, and basic sensors. The complete electrical schematic of Victoria is shown in Figure 5.

A. Computers

Victoria's central computing occurs on a National Instruments CompactRIO (cRIO) embedded computer. The cRIO uses a state machine architecture to perform each of the competition tasks autonomously. The cRIO communicates with both the shore laptop and the on-board computer designated to vision processing, through both TCP and UDP network protocols. These communication protocals were chosen to ensure both high-speed and high-reliability data streams. Victoria's internal ethernet network is linked to the shore station using the Ubiquiti airMAX long-range wireless system. The shore laptop is a Lenovo ThinkPad T520 with a Core i5 2.5 GHz processor.

As previously stated, vision processing is performed on a dedicated PC running LabVIEW in a Windows 7 environment. The hardware platform is the Fit-PC3, a consumer-grade embedded computer outfitted with a ribbed aluminum case for fanless heat dissipation.

B. Sensors

Victoria's navigation system includes a Garmin 16X GPS receiver, Microstrain 3DM-GX3-45 GPS enabled Inertial Navigation System (GPS/INS), Sick LMS-291 Light Detection And Ranging (LiDAR)sensor, and two Microsoft LifeCam Studio 1080p HD webcams. The GPS, INS, and LiDAR are connected to the cRIO via serial connections. The GPS is an integrated waterproof antenna and receiver, providing WAAS-enabled GPS data at 1 Hz with an accuracy of 3 meters. The INS contains three accelerometers, three gyros, and three magnetometers providing data at a rate of up to 100 Hz as well as secondary GPS data. The LiDAR is a 75Hz, 180-degree scanning range finder used for object detection and avoidance. The data from the LiDAR is used in conjunction with camera images for littoral navigation, shoreline detection, and detecting mission station props. The cameras are connected via USB to the vision computer, which is connected to the cRIO via a Linksys router. These cameras were chosen for their high definition, low distortion, and auto focus feature. For the Hot Suit mission, an Omega OS136 IR temperature

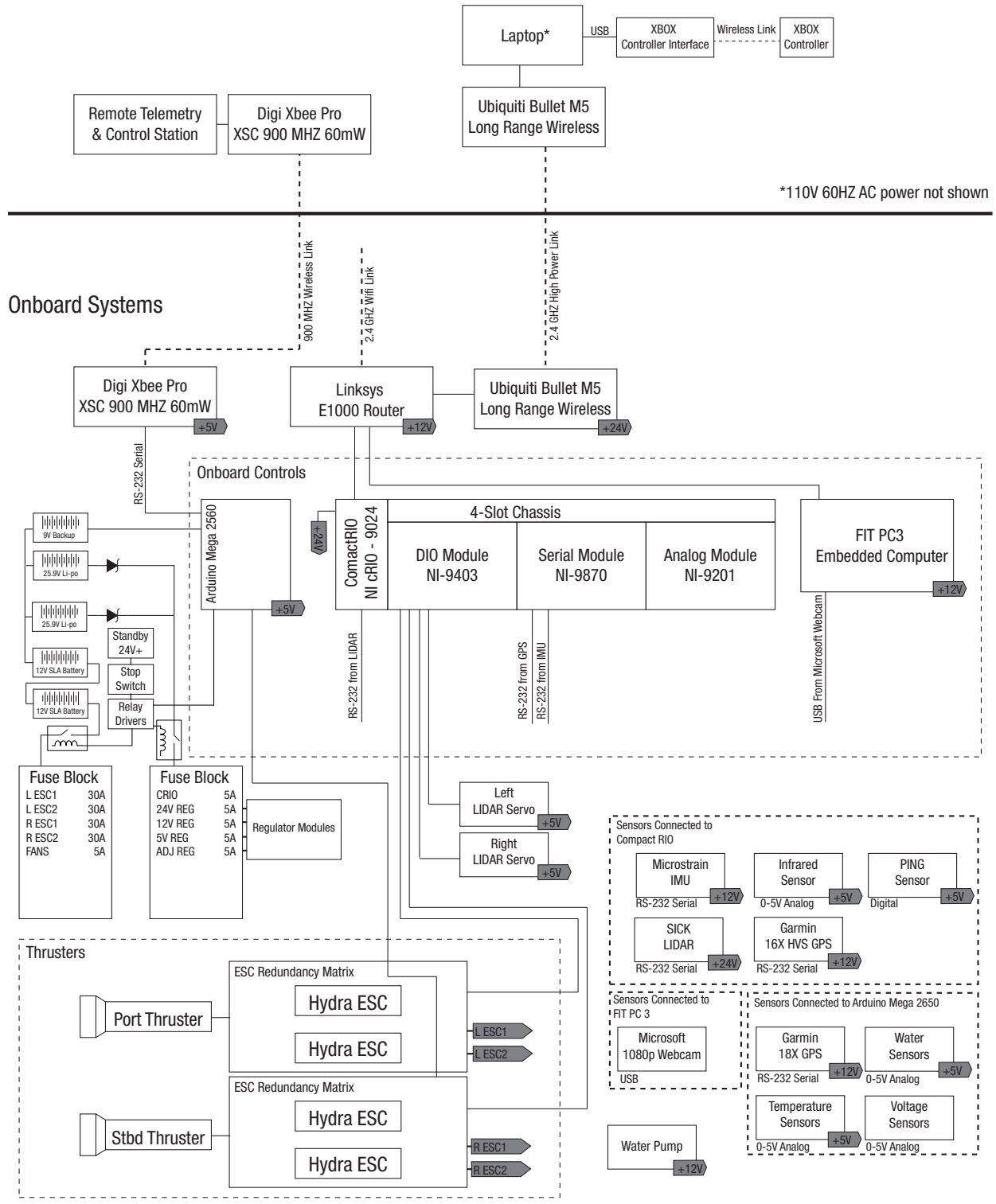

Fig. 5: Electrical Schematic

sensor is mounted to the vehicle to determine the suit with the highest temperature.

IV. Mission Control

A. Speed Gates and Channel Navigation

An Extended Kalman Filter (EKF) based Simultaneous Localization and Mapping (SLAM) algorithm was implemented in order to navigate through both the speed gates and the channel. This complex algorithm involves several operations. First, we must detect potential obstacles; this step consists of identifying objects using range data from the LiDAR and color data from the cameras. Then, we accurately determine the location of the object by filtering the different sensors data through the Extended Kalman Filter. Finally, Victoria has to successfully proceed through the speed gates and the channel using a custom curve tracking controller. The only difference between navigating through the speed gates and the channel is avoiding the yellow buoys in the middle of the channel.

Object Detection - Camera

Image processing is performed using the National Instruments Vision Assistant software. Once an initial image is obtained several separate buffered images are saved, one for each color of buoy. The following process is repeated from the original saved image for each color. First, blocks of color are detected by setting acceptance thresholds in the Hue Saturation Intensity (HSI) color space. HSI is used instead of a Red Green Blue (RGB) space because the HSI has a lower dependence on background lighting. This filter then produces a binary image consisting of passed or not passed pixels. If there exists a large amount of noise close in color to the buoy, the erode function is used. Erode removes textured surfaces, by shrinking them significantly faster than smooth surfaces. This function is useful since natural surfaces tend to be more textured than buoys. Next, a filter for the area of each blob is applied for all images, and any blob smaller than a predetermined threshold is rejected. Each of the remaining blobs are bounded by a rectangle.

The location of the center of the bottom line for each rectangle is exported along with the color code from the filter. This data is then combined with the camera properties: angle, height, field of view, and resolution, to approximate the direction of each blob. This direction is used to match each object in the camera's field of view with the objects detected by the LiDAR.

The system is calibrated by using videos recorded during test runs. Individual frames are analyzed and the different parameters are adjusted to capture the buoys. These adjustments are then tested on the entire video clip to confirm their accuracy.

Object Detection - LiDAR

To identify whether an object is detected, we must analyze the difference between the previous and current data points. For objects in the water two parameters must be considered: (1) the distance threshold, Δr; (2) the angle span, $\Delta\phi$. The distance threshold is defined as the difference of the distances between two consecutive data points, and the angle span is defined as the difference of angles between the beginning and end of an object. These parameters will vary depending on the current environment, and must be experimentally set. An illustration of LiDAR data can be seen in Figure 6. A large difference in distance represents a gap in

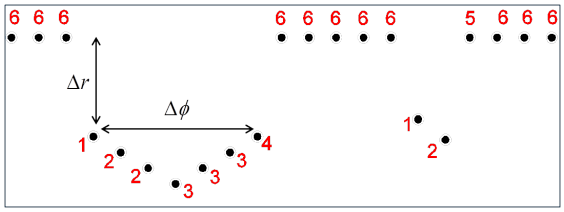

Fig. 6: Diagram of Sample LiDAR Date and Table of Each Scenario

the data, denoting the beginning or end of a potential object. Since noise is to be expected in LiDAR data, we must further inspect the data to confirm that the potential object fulfills both parameters. A feature label pattern recognition method is used to detect the objects from the LiDAR data. Each point of a LiDAR scan is given a numerical feature value, 1-6. The labeling can be seen in Figure 6, where each data point is labeled in red with its corresponding feature number.

To successfully label each data point, we must traverse through multiple nested if-else statements. Figure 7 shows the hierarchy structure of the if-else statements.

A flow chart of the data labelling process is illustrated in Figure 8. Once the data points of an entire LiDAR scan are labeled, a pattern recognition algorithm is used to identify where actual objects are located in the scan.

The template used to identify an object consists of a sequence of features represented by $\{6, 1, 2, \ldots, 2, 3, \ldots, 3, 4, 6\}$. In Figure 6, the data represented by $\{6, 1, 2, 2, 3, 3, 3, 4, 6\}$ is an object, while the data $\{6, 1, 2, 5, 6\}$ is considered noise.

SLAM

Simultaneous Localization and Mapping is a popular,

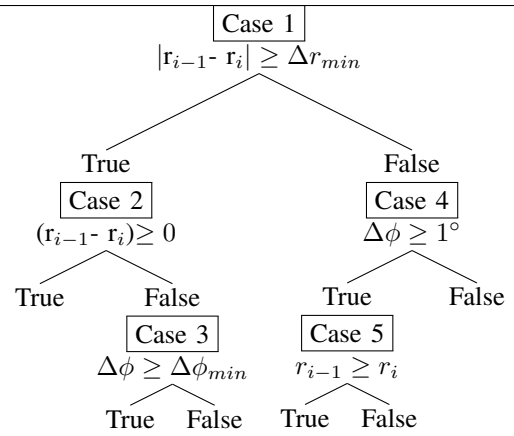

Fig. 7: Tree Diagram of Nested Case Structures

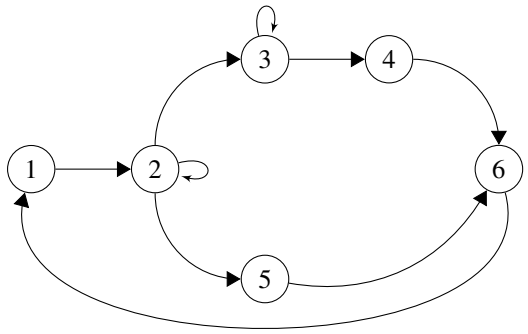

Fig. 8: Feature Labels Diagram

yet difficult to implement, technique used to map an unknown environment while tracking a robot's location within that environment. Although erroneous data from sensors are useful in creating a rough estimate of an environment, passing all data through a Kalman filter is an advantageous way to reduce noise. Since Victoria's dynamics are based on a unicycle model, which is nonlinear, we use an Extended Kalman Filter (EKF) rather than a Kalman Filter. Victoria successfully uses an EKF-SLAM algorithm to aid in all navigation. In order to use EKF-SLAM, we must create a state space model for the system. The state space model consists of a state equation and an observation equation; the input parameters are the state and input vectors. We use the position and heading of the robot, as well as the coordinates of each detected buoy to construct the state vector. The input vector is composed of the linear and angular velocity of the robot. The state vector **x** and the input vector **u** are defined in equations (1) and (2).

$$\mathbf{x} = [x_r \ y_r \ \theta_r \ x_1 \ y_1 \dots \ x_N \ y_N]^T \tag{1}$$

$$\mathbf{u} = [v \ \omega]^T \tag{2}$$

We use the state and output equations, (3) and (4), from [1] in our implementation. At each time step sensor data is used to update (1) and (2). The non linear matrices f and h are added to zero mean Gaussian random noise vectors, ϵ and δ .

$$\mathbf{x_k} = f(\mathbf{x_{k-1}}, \mathbf{u_k}) + \epsilon_\mathbf{k} \tag{3}$$

$$\mathbf{z_k} = h(\mathbf{x_k}) + \delta_\mathbf{k} \tag{4}$$

The GPS provides the latitude and longitude location of the robot, which are then converted into a Cartesian coordinates. The IMU yaw reading represents a true north magnetic bearing, which is directly used as the vehicle heading. To compute the coordinates of the buoys we use angle and distance data from the LiDAR, as well as the Cartesian coordinates and heading of the robot. By sending the GPS, IMU, and LiDAR data through an EKF, we reducing the amount of error associated with each sensor. This allows us to accurately map each detected buoy.

Controller

The channel navigation and the speed gate missions are based on the curve tracking algorithm from [2]. Curve tracking is a fairly common controller used on autonomous vehicles, which navigates the robot along a predefined curve or line. The first step in this algorithm is to select the objects located in front of the robot. Then, the objects are sorted by distance and color, which are determined by comparing the camera and LiDAR data. This condition is given in equation (5).

$$(\theta_i - \delta_{th}) \geq \alpha_i \leq (\theta_i + \delta_{th}) \tag{5}$$

where θ_i represents the angle between the robot heading and the buoy, α_i is the angle to the buoy from the camera data, and δ_{th} is the angle threshold.

For the curve tracking controller, we must assume the vehicle and objects satisfy the Frenet-Serret equations shown in equations (6) and (7).

$$\begin{aligned} \dot{\mathbf{r}}_1 &= v \ \mathbf{x}_1 \\ \dot{\mathbf{x}}_1 &= vu\mathbf{y}_1 \\ \dot{\mathbf{y}}_1 &= vu\mathbf{x}_1 \end{aligned} \tag{6}$$

$$\begin{aligned} \dot{\mathbf{r}}_2 &= \dot{s}\mathbf{x}_2 \\ \dot{\mathbf{x}}_2 &= \dot{s}\kappa\mathbf{y}_2 \\ \dot{\mathbf{y}}_2 &= -\dot{s}\kappa\mathbf{x}_2 \end{aligned} \tag{7}$$

where v is the linear speed control, and u is the angular speed control of the robot; κ is the curvature constant and s is the arc length parameter of the curve [2]. The robot and the object position vectors are $\dot{\mathbf{r}}_1$ and $\dot{\mathbf{r}}_2$, respectively. The curve tracking algorithm generates an imaginary curve through the two closest pairs of red and green buoys, and a circle around each yellow buoy. The red and green curves serve as the outer boundaries for the channel, while the circle allows Victoria to avoid the yellow buoys. Using equations (8), (9), and (10), an angular speed is calculated for each curve.

$$u_r = \frac{-\kappa_r}{1+\kappa_r\rho_r}\cos(\phi_r) - \lambda(\rho_r - \rho_o)\cos(\phi_r) - \mu\sin(\phi_r) \tag{8}$$

$$u_g = \frac{\kappa_g}{1+\kappa_g\rho_g}\cos(\phi_g) + \lambda(\rho_g - \rho_o)\cos(\phi_g) + \mu\sin(\phi_g) \tag{9}$$

$$u_y = \frac{\kappa_y}{1+\kappa_y\rho_y}\cos(\phi_y) + \lambda_y(\rho_y - \rho_o)\cos(\phi_y) + \mu_y\sin(\phi_y) \tag{10}$$

where ϕ_r, ϕ_g, and ϕ_y are the angle from the robot to the respective curves; ρ_r, ρ_g, and ρ_y represent the distance between the robot and the respective curves; ρ_o is the desired separation between the robot and the curves; λ and λ_y are the proportional gains; μ and μ_y are the differential gains. The controller then calculates the total angular speed, using equation (11), needed to stay within the imaginary curves. The linear speed, v, is set to a constant value.

$$u = u_r + u_g + u_y \tag{11}$$

An illustration of the imaginary curves and the desired path is shown in Figure 9.

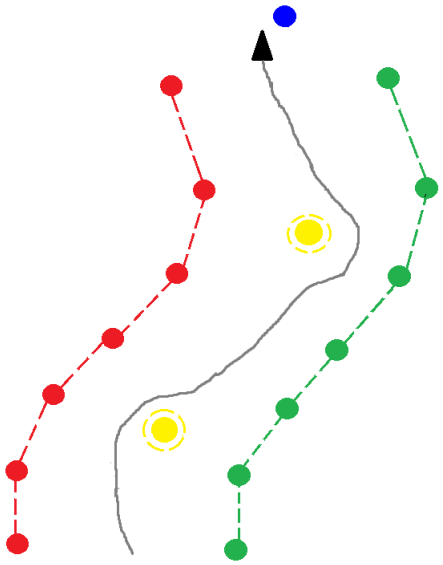

Fig. 9: Channel Navigation Path

Simulation

To begin software testing as early as possible the EKF-SLAM navigation controller was tested using the National Instruments Robotics Environment Simulator. The simulator is a new feature of the LabVIEW 2011 Robotics module, and includes simulation of a few vehicles and sensors. Since Victoria has a differential drive system, we have opted to use the NI Starter Kit 2.0, which is a three-wheeled differential drive robot. The Starter Kit powers two TETRIX wheels with DC motors and uses an omni wheel for steering. The drive system similarities made it straightforward to convert our thruster duty cycle controller outputs to angular velocity inputs for the DC motors. In addition, the simulator provides drivers for a few select sensors. The available simulated GPS is the U-blox 5 series, which is comparable to the Garmin 16X GPS, but also provides bearing data. The available LiDAR is the Hokuyo URG series, which only has a field of view of $150°$. Although the SICK LiDAR has a significantly larger field of view, we felt the Hokuyo LiDAR would suffice. Additionally, the environment simulator provides an AXIS M1011 camera, which has a field of view and resolution that are approximately equivalent to the LifeCam Studio cameras used on Victoria. The environment simulator GPS and LiDAR data were ran through the Extended Kalman Filter to obtain more accurate sensor data to be used in the state and input vector. The Axis camera was used to obtain the color of the different obstacles on the vehicle's route. The simulator environment used for channel navigation is shown in Figure 10.

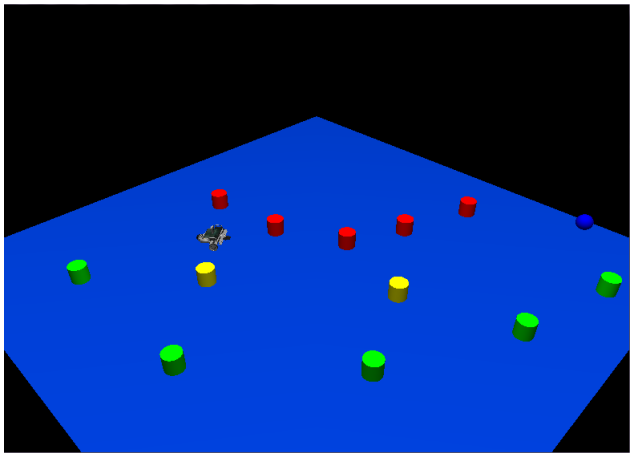

Fig. 10: Channel Navigation Simulation Environment

Additionally, sample LiDAR data used for object detection is illustrated in Figure 11. The set of three consecutive points located at an angle of approximately -50° represents a detected object, whereas the scattered dots are considered to be noise.

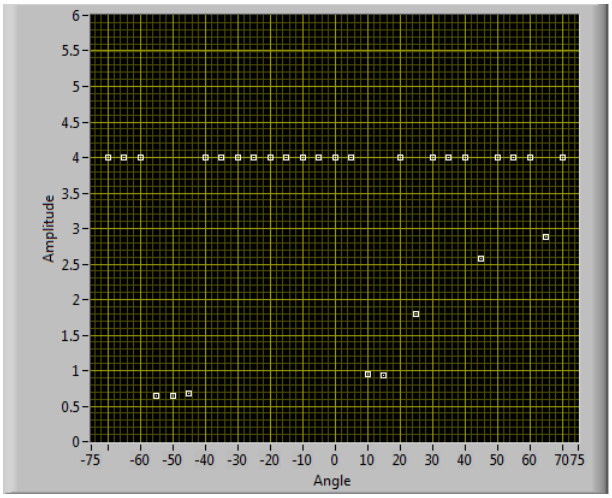

Fig. 11: Sample LiDAR Data collected in Simulation Environment

Furthermore, a sample of image processing data used in the object detection algorithm is shown in Figure 12.

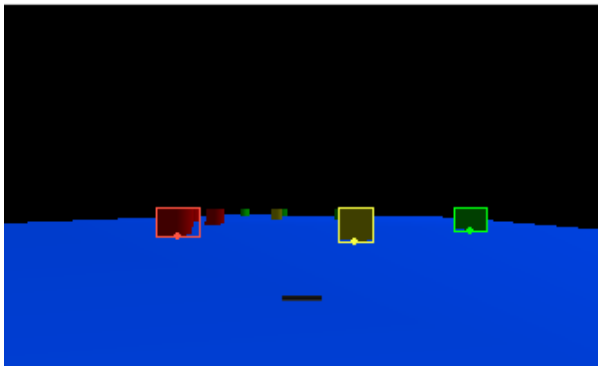

Fig. 12: Sample Camera Data From Simulation Environment

B. Poker Chip

This mission was the most difficult to prepare for as launching an autonomous vehicle from an autonomous vehicle is not a trivial task. We have designed an amphibious vehicle to deploy, shown in Figure 13, affectionately named the 'Mother Pucker' (MP) by its designers, from a custom deployment ramp. We plan to have each vehicle act independently, while Victoria monitors the sensors of the MP.

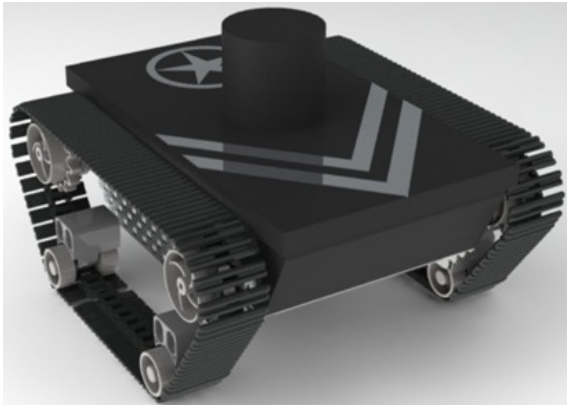

Fig. 13: Autonomous Amphibious Vehicle

The first step of the mission requires image processing to look for the specified orange duct tape lining the ramp to the dock. Once this color is within the specified pixel locations, slight forward thrust will be used to 'dock' Victoria. A linear actuator will deploy the telescoping ramp system shown in Figures 14 and 15.

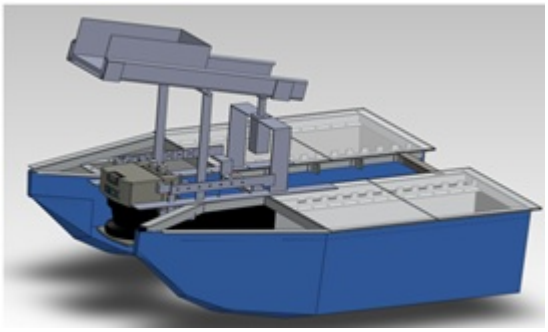

Fig. 14: Amphibious Vehicle Deployment Ramp Pre-deployment

The stored potential energy will be used to launch the secondary amphibious vehicle as shown in Figure 16. Once the MP is on the dock, it uses IR range sensors to determine when it has been deployed. Then, it begins its navigation algorithm which is a simplified occupancy grid mapping using IR sensors in much the same way a submerged vehicle would use side scan SONAR. Occupancy Grid Mapping (OGM) assigns a 2-D grid of cells, each with an associated discrete random variable that can take on two values: occupied and empty. A probability can then be assigned to each cell to determine the likelihood that a cell is occupied.

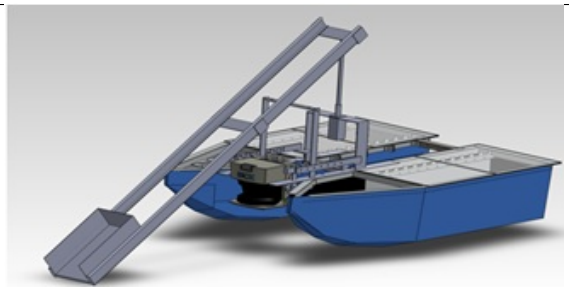

Fig. 15: Amphibious Vehicle Deployment Ramp model Post-deployment

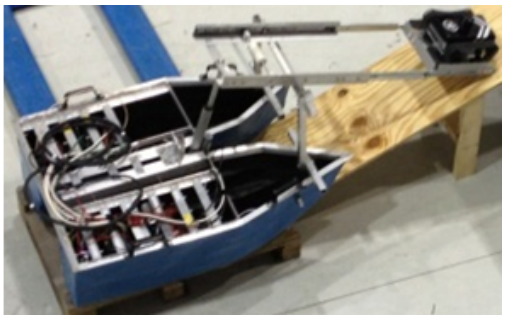

Fig. 16: Ramp Deployement Test

Occupancy grid mapping also requires a known path. Since the dimensions of the dock are given, a simple 'lawnmower' path can be generated to cover the entire area. The vehicle uses front and rear facing IR sensors to prevent it from falling off the dock, while using side scanning sensors to make a map of the environment. When the edge of the dock is detected, the grid cell probability is updated from 0.5 to 1. For the areas within sensor range, but free of any obstacles, the grid cell probabilities are updated to zero. Indexing the occupancy grid probabilities to filter out the cells that have adjacent probabilities of 1 yields the exact puck location. The vehicle can then plan a path to retrieve the puck. Once the MP has executed the planned path, it uses a dedicated IR sensor placed in the front of the vehicle to verify the mapped puck location and places the vehicle directly over the puck. Once in place, a linear actuator attached to a plate of Velcro will deploy, retrieving the puck. A mechanical limit switch is placed on the vehicle to activate this capture mechanism for redundancy. When the chip is captured, the vehicle uses the map generated to navigate back to Victoria.

There are several fail safes to ensure the deployed vehicle is safely retrieved. The first method utilizes the map generated during the run to drive the vehicle back to the ramp. A mechanical limit switch as well as an IR sensor will trigger the linear actuator to retract the ramp once the MP is in place. In the event that the vehicle can not return by its own navigation capabilities, a winch with a safety line attached to the MP will pull the vehicle back to Victoria. The final safeguard is a timed, forced retrieval. After the allotted amount of time for the amphibious vehicle to find and retrieve the chip is over, the winch will activate, reeling in released line, and then the deployment system will be retracted.

Simulation

The mission was modeled in National Instruments Robotics Environment Simulator. The challenge environment is shown in Figure 17. The map and occupancy grid created from sensor data are shown in Figure 18. The white trajectory is the robot's path, and the green represents an object detected by the IR sensors. The initial occupancy grid is on the left, with the updated grid on the right. Figure 18 shows the occupancy grid when the robot detects the chip; therefore, the cell with the chip holds a value of 1. t

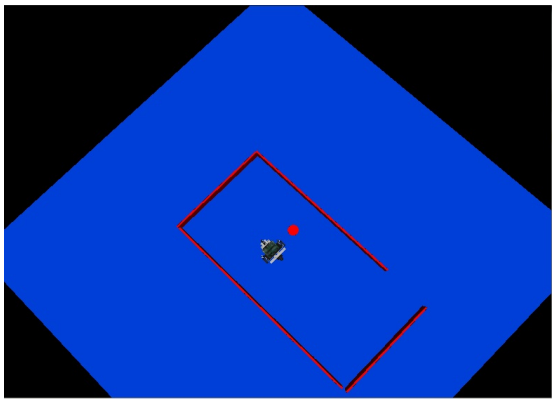

Fig. 17: Poker Chip Challenge Simulation Environment

C. The Jackpot

In addition to Victoria's primary camera, a Microsoft LifeCam Studio will be deployed inside an acrylic casing under the surface of the water. The primary camera will detect the two red buttons by color matching and a height/width ratio filter. The underwater camera will focus on finding an object with high intensity, rather than looking for the color white, so that the detection of the buoy will be more robust to color changes caused by light filtering through the water.

Because the acrylic casing for the underwater camera will create distortions in the image, a software solution

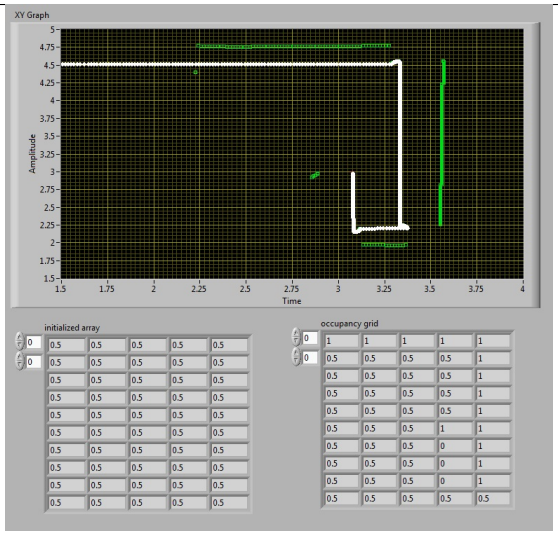

Fig. 18: Occupancy Grid Graph

utilizing a non-linear mapping between the pixel location and the angle of the object with respect to the camera will be used.

Both the underwater camera and primary camera will be placed on the same vertical axis, which allows them to work in conjunction for more precise matching. Once both a button and the buoy are detected at the same angle with respect to Victoria, Victoria will drive to that location to hit the button.

D. Cheaters Hand

This challenge relies mainly on image processing. The algorithm includes four primary steps. The first step of the mission requires image processing to identify the location of the mission station. The current camera feed is compared to a predefined template which consists of an image of the mission station. A template matching score is used to determine when Victoria has reached the station. Once the matching score has reached the threshold, Victoria searches for the fake card, i.e. the blue square. This step requires the use of a color matching function. The vehicle drives in front of the set of cards until it detects the color blue. Once the blue color is detected, a message is sent from the vision computer to the cRIO via TCP. When the cRIO receives this message, a signal is sent to the water pump to start the dispersion of water. The water pump keeps firing until the red flag is detected. Similarly, a color matching function is used to determine when the red flag has risen. The region of interest is specified above the top corner of the blue square. Once the red flag is detected, a message is sent from the vision computer to the cRIO to stop the water dispersion.

E. The Hot Suit

This challenge involves minor image processing and sensor measurement. As Victoria is navigating along the mission station, an IR sensor is used to measure the temperature of each suit, while recording their GPS locations. After Victoria has reached the last suit, the maximum temperature is determined. The suit symbol, its temperature, and its location are then transmitted back to the shore station using the provided communication protocol.

V. Conclusion

Victoria was subject to considerable redesign not only in the mechanical and electrical systems but also in software. The new mechanical and electrical design provides a more stable and reliable platform that can certainly be used for a variety of goals. Additionally, there was a complete redesign in the software architecture, adding more complex and sophisticated algorithms. Not only do the EKF-SLAM and curve tracking algorithms provide seamless navigation, but the new vision capabilities also increase our chances of completing each mission in this year's competition.

References

[1] T. Bailey, "Mobile Robot Localisation and Mapping in Extensive Outdoor Environments," Doctor of Philosophy thesis, Department of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, Sydney, Australia, 2002.

[2] F. Zhang, and M. Egerstedt," Curve Tracking Control for Autonomous Vehicles with Rigidly Mounted Range Sensors," *Journal of Intelligent and Robotic Systems,* 56(1-2):177-197, 2009.