UM::Autonomy's Hugh Jackman

Michelle Howard, Alexander Bauman, Jakob Hoellerbauer
Nick Ashcroft, Cyrus Anderson, Kunjan Singh, Jeremy Lipshaw, Parker Howard,
Marissa Witcpaleck, Travis Bowers, Lance Cummings, Ryan Jackson

University of Michigan
Ann Arbor, MI 48109

Figure 1: Hugh Jackman in the Lurie Fountain in Ann Arbor Michigan.

ABSTRACT

Hugh Jackman is a fully autonomous surface vehicle with custom pontoon hulls designed for maximum maneuverability and stability. The boat was designed, built and tested for the purpose of competing in the Eighth Annual AUVSI Foundation Autonomous Surface Vehicle Competition, where it will exhibit its abilities by attempting the competition challenges, including navigation, docking, locating an underwater pinger, and identifying an underwater light sequence. The main focus of the team this year was a redesign of the electrical systems, as well as software infrastructure. This paper explains the changes and adaptations that have been made to our boat since last year's competition.

1. INTRODUCTION

Hugh Jackman is the UM::Autonomy submission to the Eighth Annual AUVSI Foundation Autonomous Surface Vehicle Competition. Hugh Jackman was designed to execute the challenges set by this year's competition. Hugh Jackman is the newest edition to the UM::Autonomy series of fully autonomous vehicles and many of the designs are based on lessons learned in previous iterations. It would be impractical to discuss every facet of every aspect of our system in this paper. Therefore, this paper will focus on the improvements that have been made over the last year. In the past few years, we have implemented a design, build, test cycle to make iterative design improvements. Hugh Jackman is the result of the most recent cycle, designed using the lessons learned from his predecessors, Eve (2013 Roboboat entry) and Serenity (2012 Roboboat entry). While Hugh Jackman appears similar to his predecessors, he has several notable differences, including an entirely new electrical system, hull design, new sensors, and extensive changes to the software.

Throughout the remainder of this paper we will discuss all of the modifications and improvements that have been made over the past year. These changes have made Hugh Jackman our most capable submission to the Roboboat competition to date.

2. HULLS AND DECK

2.1 Hull Design

The primary design objectives for Hugh Jackman's new hull forms were maneuverability and stability. The secondary design objectives were minimizing drag and weight. The design for these hulls has been developed over several years. The design has evolved from a small waterplane area quadruple hull into the twin hull pontoon design that we have now. This hull shape was designed to be streamlined to allow for good tracking and a low drag. Also, we designed the bow and stern of the hulls to be semi-planing to allow the vehicle to travel safely at high speeds. The hulls are fully symmetric to allow for maximum maneuverability, allowing the boat to drive in reverse as easily as it can drive forward. The redesigned hulls are very stable in pitch and

roll, which is very important, as all of our software is based on the assumption that the pitch and roll angles are small (±6˚). The deck of this year's vehicle sits 4 inches higher off of the surface of the water than last year's entry, to allow for sensors to be mounted under the deck for the underwater light detection challenge and to move our other sensors out of the "splash zone".

Figure 2: A SolidWorks [1] model of the hull form.

Last year, in order to achieve the maximum maneuverability we developed an angled thruster configuration that allows the boat to turn very rapidly to avoid obstacles (Figure 3). This year we have improved on that design, allowing the boat to navigate using a "look-at-point" and a "drive-to-point", which means the boat can be moving towards one point while "looking" at a different point.

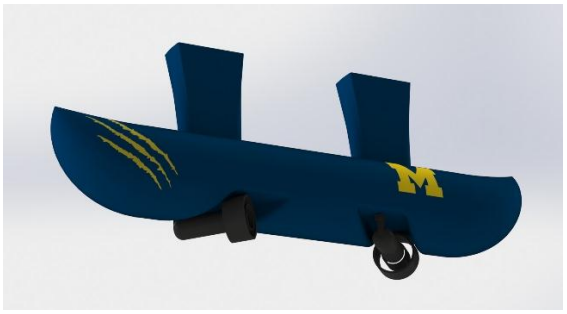

Figure 3: Angled thruster configuration that allows the boat to make rapid direction changes, as well as achieve lateral motion.

2.2 Internal Struts

Each hull is also equipped with an internal aluminum bracing system. The internal struts consist of a top plate, a bottom plate and a cross brace that runs directly though the center of the hull (Figure 4). The top plate serves as the interface between the hulls and the deck of the boat. This plate allows us to remove the hulls from the deck easily and without tools for easy transportation and servicing. The bottom plate creates a watertight interface between the hulls and each of our four Seabotix thrusters. The bottom plate is designed to allow quick thruster changes, taking approximately one minute to swap out a thruster. The cross brace connects the top plate and the bottom plate and adds structural support and rigidity to the design.

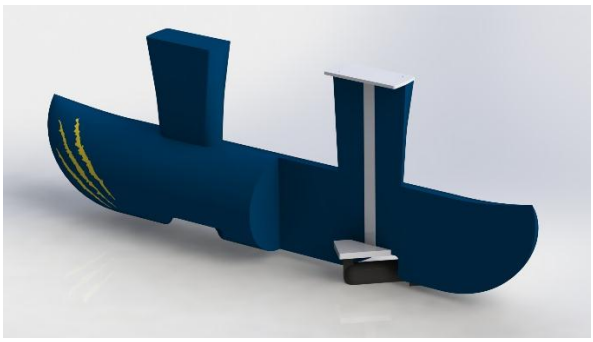

Figure 4: A SolidWorks [1] model showing a cut-away view of the hulls with the internal struts.

The components of the internal bracing system where fabricated using a waterjet (Figure 5). The pieces were then welded together to create a watertight joint.

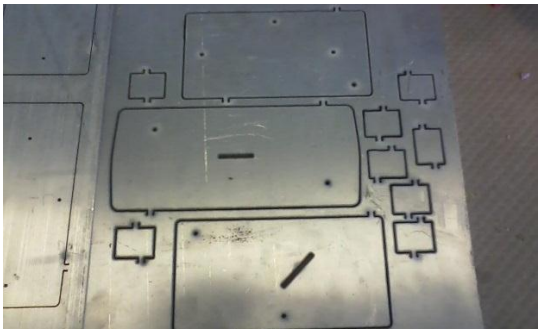

Figure 5: Aluminum pieces after being cut on the waterjet.

2.3 Hull Fabrication

Hugh Jackman's hulls were made out of High-density insulation foam using a 3D router. To begin the fabrication of our hulls, the first step was to use the SolidWorks [1] model CAD file to create a toolpath using CATIA [2] (Figure 6). The toolpath was used to guide the 3D router to cut out the shape of our hull form.

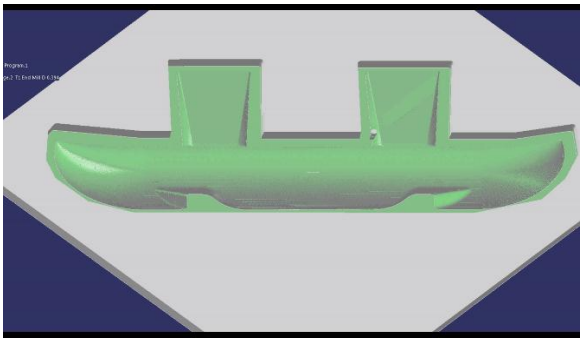

Figure 6: A screenshot of the CATIA [2] toolpath being generated.

Two pieces of high-density insulation foam were glued together into 4-inch thick slabs. The 3D router was then used to cut each slab into one-half of each hull, as shown in Figure 7. This process was repeated four times and then each half was glued together with the aluminum struts inside.

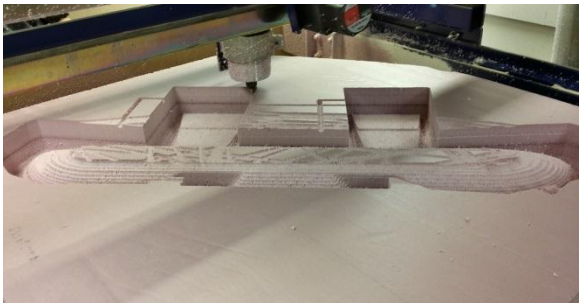

Figure 7: One half of one hull being cut on the 3D router.

Extensive sanding was done on the foam to create a smooth surface, after which three layers of 50-50 fiberglass and epoxy were added, ensuring a strong watertight shell. The hulls were sanded one final time before painting and applying custom decals. Then, a final protective clear coat was added (Figure 8).

Figure 8: Hulls after decals and final epoxy coat added.

2.4 Deck

Hugh Jackman's deck is made of an ultralight aluminum/fiberglass honeycomb composite. This material gives the deck a superior strength to weight ratio. His deck footprint is 28" by 48". Hugh's deck also has four grommet holes to allow for cables to pass through the deck, two of the grommets cannot be seen, as they are underneath the camera mounts to allow cables to run though the hollow mounts.

3. ELECTRICAL SYSTEM

Hugh Jackman's electrical system is housed within a Pelican Storm iM2700 (22" x 17"x 8"). This case was chosen because it is rugged, and machinable.

Computationally, Hugh has an Intel Core i7-4771 Quad Core processor mounted on an Asus ASRock motherboard and has 16 GBs of high-speed RAM. This formidable set-up allows for multiple processor intensive programs such as image processing to run at a high rate at the same time without any problems.

Two 120 millimeter exhaust fans are used to remove heat from Hugh's electrical enclosure. Each fan is capable of drawing 110 CFM through the electrical box's intake vents.

Power is distributed throughout the electrical system by a multi-voltage DC power bus. This bus can supply electric power at 3.3, 5, or 12 VDC. The power bus supplies power to the majority of the sensors, the cooling fans, and utility boards inside the box.

Hugh Jackman is equipped with a collection of sophisticated sensors that allow him to function autonomously. Hugh has a GPS sensor, a fiber optic gyroscope, and a digital compass that tell him his relative position and orientation in space.

Two Point Grey Flea 2, 1.3MP cameras and a Hokuyo UTM-30LX Lidar allow him to visually and spatially perceive his surroundings. By using these sensors Hugh is able to understand and react to his surrounding environment.

He uses the information from the Garmin 19x HVS GPS unit to determine his location and speed. His Ocean Server OS5000 USB digital compass allows Hugh to determine his initial heading. The KVH DSP-3000 fiber optic gyro detects instantaneous change in Hugh's orientation on the horizontal plane. Each of these

sensors communicates with Hugh's computer via a serial RS 232 signal. Hugh's GPS, digital compass, and fiber optic gyro work in conjunction to provide him with the information necessary to determine his spatial orientation and location.

Hugh's cameras allow him to visually detect obstructions and features in his surrounding environment. Combined, the two cameras have a slightly overlapping field of vision that spans approximately 180 degrees. The Lidar sweeps a laser signal in a 270 degree arc on plane with the bottom of the sensor to get range information while a Dynamixel AX-12 servo rotates the Lidar over 0.2 radians. The different laser scans are combined into a three dimensional point cloud of obstructions. Additionally Hugh has a Point Grey Firefly 1.3PM USB camera mounted underneath his deck. This camera is used to locate the underwater light display and observe the resulting light sequence that is emitted.

3.1 Sensor Mounts

Each Hugh's sensors is mounted using custom designed sensor mounts. The following sections will describe the design of each of our sensor mounts.

3.1.1 Camera Mounts

Like his predecessors, Hugh Jackman uses two cameras to observe the world. In order to get the widest field of view, Hugh's cameras are mounted as far apart on the front of the deck as possible and are turned out at a 30˚ angle. His cameras are also elevated as far above the surface of the water as the competition height limits allow. Each of the camera mounts is made of a thin lightweight plastic, with a wide rigid base that makes the cameras very stable. This means that the delicate camera-Lidar calibrations will remain valid for longer. The cameras are each enclosed in a custom splash proof acrylic case. These acrylic cases were fabricated using a laser cutter. The front of each of the camera cases is covered in a polarized film to reduce glare in the cameras (Figure 9).

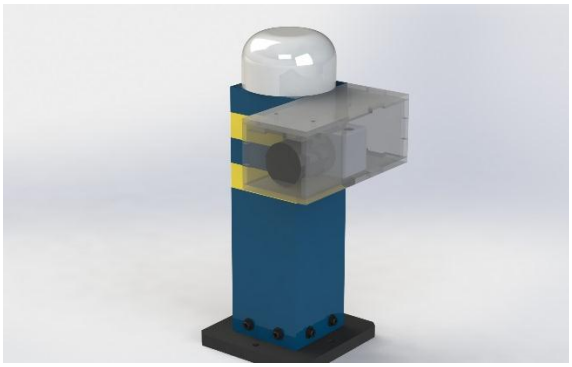

Figure 9: A SolidWorks [1] model of the camera mounts.

3.1.2 Lidar

As explained in previous sections, Hugh Jackman's Lidar collects data from a single 2D plane. In order to create a 3D point cloud, the Lidar is mounted on a pivot so the Lidar can be rocked up and down. This custom rocking Lidar mount was fabricated using a 3D printer (Figure 10). For most of the navigation type challenges, we are only interested in points approximately one foot above the surface of the water so our Lidar is mounted upside down on the bottom of the deck to give us the best view of the buoys. The range through which the Lidar rotates can be changed to provide more useful information for other challenges.

Figure 10: A photo of the 3D printed rocking Lidar mount.

3.1.3 GPS/Compass

Hugh Jackman is equipped with a GPS and a compass. The GPS is mounted on the top of the starboard side camera mount, and the compass is mounted in a custom 3D printed case on the top of the port side camera mount. This 3D printed case is splash proof, lightweight and allows for

much easier access to the compass than our past designs (Figure 11).

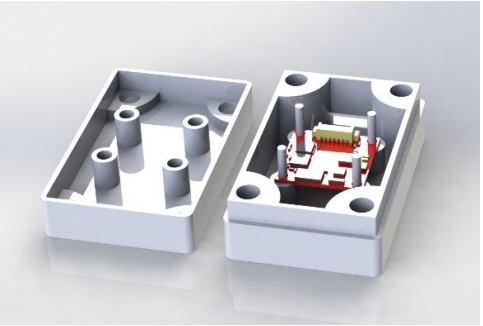

Figure 11: A SolidWorks [1] model of the compass mount.

3.1.4 Bottom Camera

The downward-facing camera was mounted in a custom-designed 3D-printed enclosure. This enclosure was then bolted to the deck of Hugh Jackman along with two aluminum spacer blocks, which provide a rigid structure in addition to space for the USB cable.

3.1.5 Hydrophone

We are using an Aquarian Audio Products H2a high impedance hydrophone to detect the location of the acoustic beacon. This hydrophone is normally omnidirectional but we created a special housing that allows us to use it as a directional hydrophone. It is mounted to the bottom of the left hull.

4. VISION AND PERCEPTION

4.1 Buoy Detection

Buoys are detected by matching Lidar detections and camera images.

From the camera images, we form blobs of colored pixels which may be buoys.

We use color filters, derived from a dataset of labeled images of buoys, to determine which pixels in an image are the same color as buoys of a certain color. Then, adjacent pixels of the color are grouped together into blobs.

From the Lidar detections, we can find the real world locations of blobs that were detected in the image.

Points from Lidar detections that are sufficiently close together are grouped into segments using Union-Find. Segments with too many, or too few points, are discarded. The radius of the segment is calculated by finding the maximum distance between any pair of points in the segment and dividing it by two. Then, the center of each segment is transformed from a spatial (x, y, z) point into pixel coordinates in the image.

Finally, for each of the segments, we find the image blob that is closest in the image to the segment's pixel coordinates. If they are sufficiently close, and not too far from Hugh Jackman, we consider the pair to be a buoy.

4.2 Sign Detection

Signs are detected first solely with the Lidar. They are then matched to detected shapes (if any) in the image.

Similar to the buoy detection, sufficiently close together Lidar points are grouped together into segments using Union-Find. Then, using the x and y dimensions of those points, a line of best fit is found using Deming (orthogonal) regression. If the linear fit is good enough, the average z value (height) of the points in the segment is sufficiently high, and the largest distance between any two points (size of the sign) is sufficiently low, then the segment is considered a sign. The slope of the linear fit is also used to calculate the angle of the sign relative to Hugh Jackman, for use in the route planning.

The location of the sign is transformed into pixel coordinates. Then, for each shape detection, the closest unmatched sign is found. If they are sufficiently close, we consider the sign to have that shape.

For each sign, the shapes are detected using the OpenCV library and overlaid on the sign to remove any unnecessary shape detections. To detect shapes, we use OpenCV image processing filters such as blur and noise reduction (bilateral filter) to improve the edge detection from the canny filter. Once the canny filter is applied, we use the OpenCV contour detection algorithm to detect contours from the edges. The contour, which is basically a set of points where the shape is located, is used with the approxPolyDP function to get the corners to form a polygon and use those to form edges and we use the edges to determine if the contour fits the parameters of a triangle, cross, or circle.

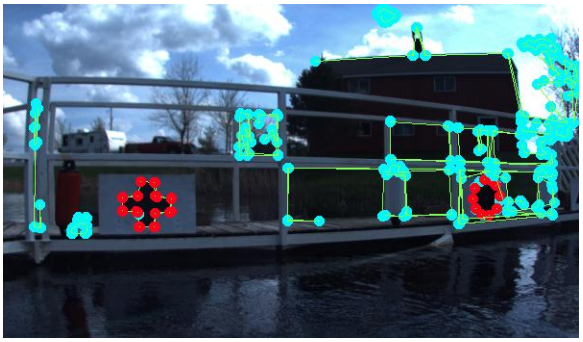

Figure 12: A screenshot of Hugh's sign detection function. The detected shapes are marked in red.

4.3 Gate Detection

To detect the gates, we consider all of the buoys that Hugh Jackman has seen so far in his run. First, all of the buoys whose radii are too small are discarded. Then, every subset consisting of one red buoy, two white buoys, and one green buoy is considered.

Our goal is to find the subset(s) we are most confident that are red white, white green buoy gates. We represent this confidence as a real number between 0 and 1, with 1 being very confident, and 0 being not at all confident.

For each subset, we first find the maximum distance between any pair of buoys in it. If the maximum distance is too large, the subset is discarded. Otherwise, a line of best fit is found on the x and y coordinates of the buoys using Deming (orthogonal) regression. The confidence for this subset is set to the "R squared" value for the linear fit. Then, we check if the buoys are in the correct color order along the line. If they aren't, they are discarded. Finally, to check how even the spacing is between the buoys, we find the distances between each pair of adjacent buoys. We multiply the confidence by the minimum pair distance divided by the maximum pair distance.

If a subset makes it to the end of that without being discarded, it is considered a gate with the corresponding confidence.

4.4 Pinger Detection

The pinger is detected by matching buoy detections with high directional hydrophone returns.

We have Hugh Jackman make a full 360 degree turn near the five pinger-buoys. At each point in time, if we are getting the highest volume so far at 0.5 Hz (the frequency of the pinger), we find the closest buoy to the line extending out of the boat in the direction of the directional hydrophone.

Upon completing the turn, we report the color and location of the buoy which corresponded with the maximum volume heard by the directional hydrophone.

5. SLAM

Hugh employs an Extended-Kalman-Filter SLAM feature based mapping system [3]. Since there are few things to localize on in the competition pond, we use a feature-based mapper and fall back on dead-reckoning when feature detections are not available for localization purposes. Due to the inclusion of the highly accurate Fiber Optic Gyro (FOG), we use the FOG returns exclusively for heading information. On initialization, we collect a set of compass and fog observations and compute a "globalization" constant for the FOG measurements. This allows us to use both GPS and FOG data in a global frame for the purpose of dead reckoning. For building data correspondences between buoy detections, we use the recursive Joint-Compatibility Branch and Bound algorithm [4]. Some challenge station features are unique; in that case, there is a known correspondence and data association is trivial. We also have some capabilities for map correction. These include detection and elimination of duplicate features and detection of map corruption. By maintaining a good map of the competition environment, we are afforded many advantages that are not possible with a simple short-term mapping technique, namely, adaptive obstacle course navigation. This will be discussed at length in the following section.

6. ROUTE PLANNER

The "route planning" part of our code takes input from sensors, detectors, and mappers, and decides where to go. Since we must make it through the speed gates before getting points for any other tasks, our route planner is split into a function for speed gates, and a function for all other tasks. Hugh will not continue to the other tasks until his attemptSpeedGates function is successful.

6.1 Speed Gates

The speed gates are handled just as buoys in previous years' buoy channels; the most optimal red and green detected buoys are paired and the boat navigates through the pairs.

6.2 Challenge Tasks

To start attempting tasks, we first explore until we're confident enough that we've detected some task. We have a detector for each task, and each of those detectors determines the confidence threshold for its task. Once we start to get detections, we order the detected tasks by how much we'd like to attempt them at the current time. We approach the highest ranked task, while constantly updating the task ordering. We take into account factors such as distance, detection quality, and preset weights. Since we are always updating our task ordering, if we see a new task, we can approach it if it's more highly ranked than whatever we're approaching. This, of course, makes Hugh better at selecting tasks to attempt, but can also lead to infinite loops. Therefore, we use a cycle detector to break Hugh out of task approaching loops and settle on a single task. More generally, the control flow of all of the route planning code for tasks is designed so that Hugh should only get stuck if he has no better thing to be doing.

Hugh remembers the tasks he's attempted, the outcome for each attempt, and the order in which he's made his attempts. This information is used when ranking tasks to approach.

6.2.1 Docking

Lidar data is used to identify possible signs on the dock. This data is then used to navigate directly in front of each sign and camera data is used to determine the symbol on each sign. Once the proper sign is identified Hugh will drive towards to correct location until at least one of the 2 limit switches mounted on the front of the deck are actuated.

6.2.2 Underwater Lights

Using the general GPS coordinate of the challenge as a starting point, Hugh Jackman starts executing a search pattern to detect the blinking white light. Once the light is detected by the camera mounted beneath the deck, Hugh moves into position above the light array, sending the activation message once ready. The sequence of lights is captured by the camera and the best matching color pattern is reported.

6.2.3 Acoustic Beacons

Hugh finds this area by its moored markers. He moves towards the center of the buoys and slowly turns 360 degrees so that the hydrophone is directed at all potential sound sources. The buoy that corresponds to the largest sound intensity as picked up by the hydrophone is chosen. Its GPS location is estimated and returned via wifi message.

6.2.4 Obstacle Field

For this task Hugh simply listens to detections of buoys in the Red-White-White-Green (RWWG) pattern and from this heads toward the gate specified in the JSON message. While avoiding obstacles, Hugh maximizes his available range of motion by keeping near to the calculated center of the channel, and at the same time examining sets of buoys matching the RWWG pattern. The channel is exited just as it is entered.

7. UTILITIES

7.1 Bot-Procman

Due to the design, our process manager can be used both for test/debug sessions and competition runs. The process manager is invoked with a configuration file as an argument. The configuration file names all the processes that can be run. Because the configuration is not hard-coded, we can use a different configuration when testing as opposed to a trial run. In addition, the process manager sports an interactive graphical user interface for managing the processes. This allows the developer to start and stop a process or view the process's output without searching for the terminal it was started from. The process manager also attempts to ensure that processes are running and working as intended. For this, the process manager will restart a process if it crashes for whatever reason. Also, the process manager listens to the LCM channels of the managed processes. Using the frequency of publishes the process manager can determine if a process has become stuck even if it has not terminated. These features aim to improve robustness by preventing a total system failure due to a minor bug.

7.3 Vis

There are too many LCM messages at one time to quickly understand as a human. To quickly gain knowledge of what the boat is doing at a certain time, we have developed a visualization system. In this environment, we can see what is within SLAM. This includes a 3D environment, where we can pan and zoom onto locations of the boat, buoys, and other challenge station elements. This aids the debugging process, since we can quickly see what led the boat to performing unwanted behavior.

7.4 Movement Control System

In the past years, the boat's thrusters were mounted oriented straight-back. This meant the boat had to use a differential drive system to maneuver. This restricts maneuverability. Since maneuverability is more important than speed in this competition, we decided to go with a different thruster orientation that would allow for greater maneuverability. To this end, Hugh Jackman's front two thrusters are angled out at 45° while his back two thrusters are angled in at 45°. With this thruster orientation, the boat can move in any direction, not just forward as with differential drive.

To take advantage of this maneuverability, the control system for the thrusters had to change. In the previous system, the boat tried to move to a desired point by adjusting its heading until the boat faced the point. In the new system, the boat has separate "move to" and "look at" points. This is because the boat can now move in a direction different from the one in which it is facing. This also allows the boat to rotate in place.

The control system resolves the vector from the boat's current position to the "move to" point into the four thruster values necessary to move the boat in that direction. A PID controller controls the speed at which the boat moves in this direction. This means the boat will be able to exactly achieve the desired position.

The control system uses a second PID controller to maintain the boat's heading so that it is facing the desired "look at" point. The output of this controller is converted into a rotation factor that is resolved into the four thruster values needed to rotate the boat.

The thruster strengths needed to move the boat and turn it are added together and then the sums are applied to the actual thruster. This allows the boat to turn to face the correct direction and move towards the "move to" point at the same time. Such an ability is especially useful when the boat has to circle an object of interest to get a different orientation while still staying focused on that point of interest.

Due to the separated nature of the control system, the boat can either rotate to face a specific direction, move towards a point while maintaining the current heading, or change both heading and movement direction at the same time. Also, the old differential drive method is still achievable by setting the "move to" and "look at" points to be the same. With this new system, the boat has a versatile range of movement.

7.6 Wireless Communication

We have made a process to connect with the network and send the appropriate messages to the judges. This program takes in the appropriate information about the wireless network, the socket address, and the desired message to send. It sends system commands to the operating system to connect it to the appropriate wifi network, then sends the JSON formatted message through the appropriate socket.

8. REFERENCES

[1] Solidworks [Computer software], 2012, Waltham, MA, Dassault Systemes.

[2] Catia [Computer software], 2010, Dassault Systemes.

[3] A. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 29(6):1052 –1067, June 2007.

[4] Joint-Compatibility Branch and Bound algorithm