Team Owltonomous
RobotX Journal Paper 2018

Eric Nieves, Armando Sinisterra, Mehmet Tasbas, Andrew Barbosa,
Jacqueline Parkinson, Travis Moscicki, Mickael Pigeot, Daniel Resio
College of Engineering and
Computer Science
Florida Atlantic University
Boca Raton, Florida 333431

***Abstract*—This document presents the design overview and challenges within the development of the 2018 16 foot Wave Adaptive Modular Vessel (WAM-V) as it is adapted to achieve all RobotX task requirements. The team is comprised of Florida Atlantic University (FAU) students. The vessels sensor suite employs a previously used RoboSub autonomous underwater vehicle (AUV) that has been down-scaled in size for the RobotX specifications, RGB-D vision system robust to lighting variations, underwater USBL (Ultra Short Base Line) acoustic localization system, GPS aided MEMS-based Inertial Measurement Unit, and a Velodyne VLP16 3D LIDAR. As a research platform, the FAU vehicle has served to further the development of adaptive control and robot vision systems. For example, the controller is robust to various environmental disturbances, including wind force, current, lighting variations and rain, while the vision system implements agglomerative hierarchical clustering (AHC) to produce unique object detection. A hierarchical structure and finite state machine allow for the development of modular routines which can be rapidly implemented and are utilized to conduct mission-level control. Path planning, mapping, obstacle avoidance, navigation, and three degree of freedom state estimation embody the focus of Team WORXs WAMV USV16 platform.**

I. Introduction

This document serves as the means to convey our teams approach to the RobotX 2018 Competition Tasks. As a constructive Pacific-Rim Partnership between Universities from five countries, the third biennial RobotX 2018 Challenge presents autonomous system development for the maritime domain. The challenge is organized by the Association for Unmanned Systems International (AUVSI) and the Office of Naval Research (ONR), and seeks to foster innovation and develop competitive engineers with a thirst for robotics in the maritime domain.

The scenarios presented for each competition task simulate problems encountered in real-world applications, while dynamic task association enforces modularity and the need for a robust state machine within the software development scope.

Section II will review the vehicle designs, Section III explains how these vehicles will be used in the team's competition strategy, and Section IV will showcase preliminary results. Section V will conclude the paper with our findings.

II. Vehicle Designs

The vehicle designs consist of the predominant WAM-V and it's various subsystems as well as a secondary AUV, named Hedy, used for the underwater ring task. The following section reviews the mechanical, electrical, and software subsystems for both vehicles.

A. USV Design

1) Mechanical: Although the skeleton of the WAM-V is the same across teams, there are a few mechanical additions to the FAU competition vehicle that are notable.

a) Owl's Nest: Fig. 1 shows the Owl's Nest, an independent subsystem designed as an integration platform for various components like the light tower and GPS unit. The components were fabricated and welded in house by students.

Fig. 1. A custom frame created for the integration of various sensors

b) Acoustics Set-Up: The need of a low cost solution for the acoustics subsystem was fulfilled by utilizing components already present in the lab space: transducers, a carbon fiber pole, and a linear actuator. The system was designed to not interfere with the AUV deployment system. Fig. 2 shows the carbon fiber casing and linear actuator. Fig. 3 displays a

close up of the custom made transducers used in the acoustics system.

When retracted, the system allows the WAMV to clear a buoy. When deployed, the transducers are located two feet below the water surface. All components for this subsystem were machined by hand by team members in the FAU machine shop except for the triangular plates, which were fabricated via water jet. The spacers which mount to the carbon fiber boom are 3D printed from ABS and employ heat set inserts for structural integrity.

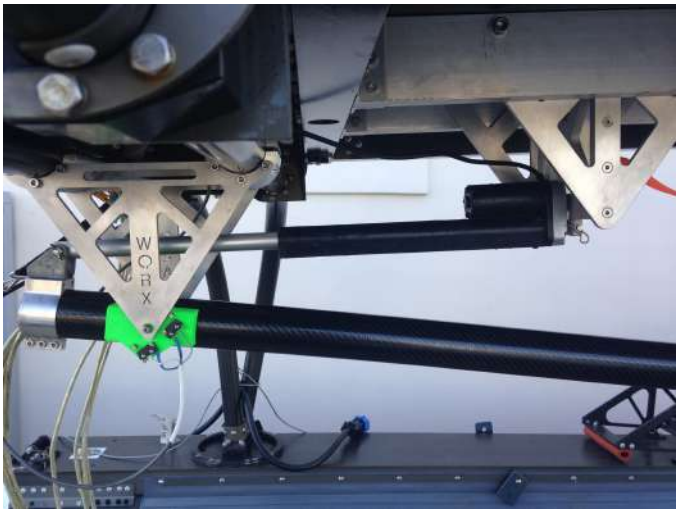

Fig. 2. Acoustic Boom and actuator configuration shown mounted beneath the WAMV payload tray

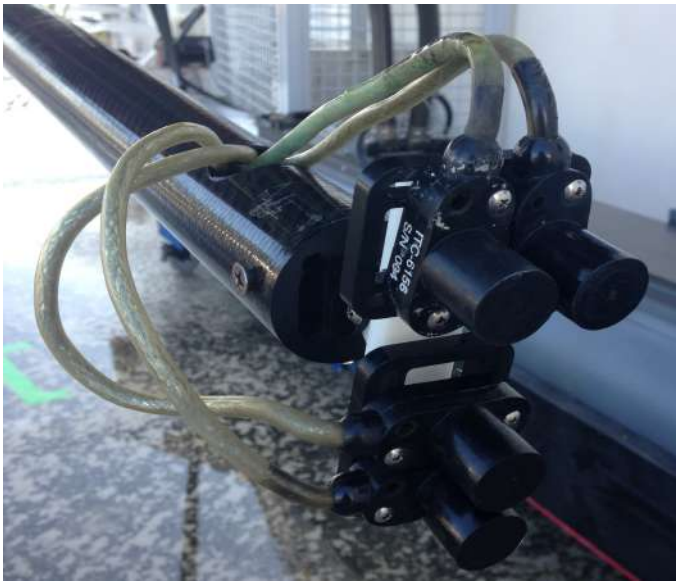

Fig. 3. Physical Hydrophone Configuration for Acoustic Data Acquisition System

c) Thrusters: The thrusters chosen were re-purposed trawling motors with anodized and custom made mounts. An example of the port side mount can be seen in Fig. 4

Fig. 4. Port side thruster mount

d) Winch System: The custom made winch system shown in Fig. 5 was recycled from the previous competition and applied to the new AUV design.

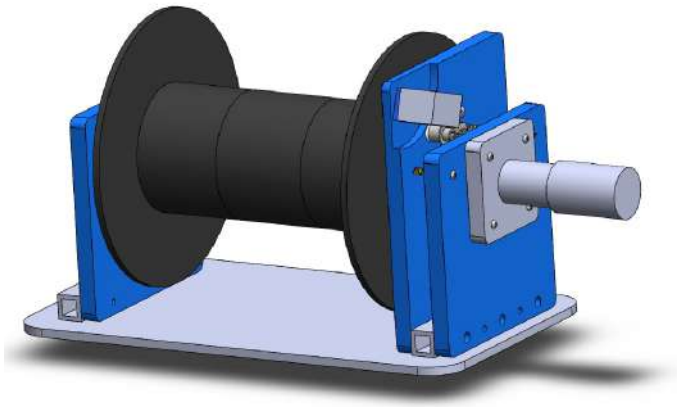

Fig. 5. A custom made winch system for deploying the AUV

2) Electrical: The inspiration for the electronics box was largely derived previous designs [1]. Mission critical components in the electronics box include the Jetson TK1 Compute boards, a STM32 microcontroller, power distribution circuits, telemetry systems, and the motor control interface. Emphasis was placed on making sure each component can act individually and concurrently without overloading the power distribution systems. A rendered view of the motherboard can be seen in Fig. 6

a) Jetson TK1: The Jetson TK1 Compute boards were chosen to act as the CPUs for the system. The design ensured that these boards were able to communicate with the other systems during operation of all condition. In order to sync this communication with accurate timing, a Real Time Clock (RTC) was chosen to be battery operated so that during communication failure, the Jetsons would still have the correct time.

Fig. 6. Electronics motherboard rendering

b) STM32: The STM32 Microcontroller acts as the Bottom Level Manager (BLM), where the low level real time devices are managed. This board has a primary function of measuring capacity in the battery supplies and indicating their state to the system. This was important to ensure the power distribution system runs properly.

c) Telemetry: The telemetry system has two channels, a primary link with 80 Mb/sec used for link the vehicle to the ground station as well as a back-up link with 115Kb/sec for mission critical data.

d) Motor Control: The motor control (PWM) interface was designed to provide hardware control over the motors from shore with a standard RC remote. A multiplexer control line originates from a PIC microcontroller with a singular job of detecting which way a switch is set on the RC remote, exceeding the level of control a user could have got from software solutions.

3) Software: The software for the WAM-V is split between low-level, middle-level, and high-level systems.

a) Low-Level: The low-level subsystems constitute the set of nodes to be executed in order to obtain basic functionality of the USV. The primary components of this suite are positional sensors and a low level control system. Secondary components of the low level system are the health monitoring system (HMS) and the communications system.

Navigation and inertial sensor drivers allow data acquisition from the XSENS IMU/GPS and OS-5000 digital marine compass. The XSENS unit provides filtered data regarding the current state of the USV in terms of its position, orientation, corresponding speeds, and accelerations. The pose information is available in both the North-East-Down (NED) and East-North-Up (ENU) frames inertial and body-fixed frames as appropriate. Both conventions are readily available and permanently defined in order to allow for a smooth transition between the Virtual Marine RobotX Challenge (VMRC) simulation environment (built in ENU) and on water operation (built in NED). The transformations shown in equation asdf are applied via ROS's TF pipeline.

A rich set of previously developed station keeping controllers remain available for the current distribution. New to this years effort is an Enclosure Based Steering [2] approach to mitigating cross track error via a coupled heading and velocity controller. The complete solution shown in Fossen's Handbook for Marine Controls is left out for brevity, the equations which require solving are show in (1)-(3). Results of these two are shown in sections results.

$$\chi_d(t) = atan2(y_{los} - y(t), x_{los} - x(t)) \tag{1}$$

$$[x_{los} - x(t)]^2 + [y_{los} - y(t)]^2 = R^2 \tag{2}$$

$$tan(\alpha_k) = \frac{y_k + 1 - y_k}{x_{k+1} - x_k} = \frac{y_{los} - y_k}{x_{los} - x_k} \tag{3}$$

The HMS provides battery voltage and current information, internal temperature readings, and state information about the acoustics and UUV deployment actuators. The communication is predominately handled by ROS, however the ground station maintains a 900 Mhz RF link to the boat with mission critical data to ensure telemetry while operating on a heavily congested Wi-Fi network.

b) Mid-Level: Mid-level systems provide information about the vehicles operating environment. The optics suite includes the Velodyne VLP16 3D LiDAR driver, a Logitec c9000 webcam driver, and a projection of the LiDAR voxels onto the webcam's field of view . Laser-based obstacle recognition takes advantage of the fact that the LiDAR only returns information from obstacles floating on the water surface, but not from the water surface itself. However, for certain tasks color information is required. To leverage the sparcity and robustness of the LiDAR, the entire point cloud is first segmented by use of a Nearest-Neighbors unsupervised algorithm, called Agglomerative Heirarchical Clustering (AHC) [3]. The benefit of AHC over supervised approaches, such as K-means or K-Nearest Neighbors, is that the number of detected objects is not and a priori requirement. While the computational cost of AHC can be larger than K-means or KNN, $O(n)^3 * O(n)^2$, this performance hit is not truly felt on the water, as the available information is far less sparse when compared with on land operation. Results from AHC are shown in Figure asdf.

This performs a pre-classification routine based on clustering points into single identities associated to the corresponding obstacles of the challenge. This information is also used to train a convolutional neural network for a definitive classification inspired by models such as [4].

Mapping functionality is provided by implementing the move_base package from the ROS Navigation Stack, which computes local and global cost maps around instances of obstacles captured by the LiDAR in order to avoid collisions while minimizing the traversed distance. Global cost maps can

be computed over static maps of the environment generated by methods such as SLAM. Local cost maps, on the other hand, build a map that travels locally around the USV, considering only the most updated data from the perception system.

c) High-Level: Path-planning is also implemented using the move_base package from ROS Navigation Stack. It is given a start and goal pose for the vehicle to achieve, and generates the corresponding global and local trajectories. Global trajectories are computed using the global cost map as a series of discrete points that defines it. Local trajectories unfold from using local maps to compute linear and angular velocity commands for vehicle motion, preventing collisions while minimizing deviations from the original global trajectory.

From a procedural perspective, the high-level mission planner, defined in the system as the_planner, is a sequence of switch cases, each of which flags and executes a particular task from the RobotX challenge if associated conditions are met. Every task is defined as a derived class that inherits from a general base class. This provides a uniform and modular interface for the user and allows all missions to share a common communications and persistent data.

d) Localization, Mapping and Motion Planning: At the base of the High-Level Mission Planner resides fundamental navigation requirements for the USV, namely localization and mapping, which provides the system with data regarding vehicle pose as well as knowledge of the surrounding obstacles. This information is a crucial aspect for the autonomy of the USV, since it supplies all the required data in order to compute optimal trajectories across safe navigation area. Potential obstacles could be any RobotX challenge task, natural objects, (e.g., shoreline, mangroves, rocks, etc.) or other man-made obstacles (such as boats or docks in the vicinity).

Over the design stage, our approach was first tested using the Virtual Maritime RobotX Challenge (VMRC) environment. This simulation provides all the physical models for the WAM-V USV (geometry, mass and inertia) and environmental conditions using the Gazebo physics engine. Gazebo plugins were also provided to simulate the GPS and IMU sensors, as well as the Velodyne VLP16 3D LiDAR, which is the actual 3D Lidar used in the USV configuration for the competition. The implementation of this virtual environment allowed us to quickly assess the localization and mapping approach explained above, so as to determine its viability for its further implementation in the actual USV. Fig. 7 illustrates this approach, in which the mapping functionality is tested based on the virtual model from Gazebo, and shown at the same instance of time.

Localization was approached by fusing data from the GPS and IMU sensors into an extended Kalman filter (EKF), implemented using the robot_localization package from ROS. This also incorporates a general non-holonomic motion model of the vehicle in order to estimate a true belief of its current state. The state of the vehicle is represented as a 15-dimensional vector:

$$\{X, Y, Z, \phi, \theta, \psi, \dot{X}, \dot{Y}, \dot{Z}, \dot{\phi}, \dot{\theta}, \dot{\psi}, \ddot{X}, \ddot{Y}, \ddot{Z}\}$$

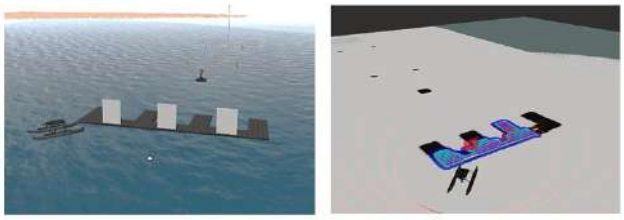

Fig. 7. Left: Gazebo representation of RobotX challenge using VMRC. Right: LiDAR-based mapping, visualized using RViz.

where the first six components correspond to the linear position and angular motion (roll, pitch, yaw), respectively. The rest appertain to their first and second derivatives (only first derivative for the angular motion). A slightly different approach was considered for the actual implementation of the localization system, which did not require using the robot_localization package. Instead, the filtered output from a XSENS device was used. At the core, the output of the XSENS unit is also based on fusing GPS and IMU data using an EKF; however, the unit entails a highly calibrated system which provides very accurate localization data.

Mapping of the environment is achieved by implementing the ROS Navigation Stack, specifically the move_base package. the package provides the functionality of constructing local and global cost maps around the obstacles, as well as to choose among various types of local and global motion planners. Cost maps consider the geometries of both obstacles and the vehicle to compute clear and efficient trajectories. Global cost maps may use prior knowledge of the world, provided from previous maps generated by any kind of technique (SLAM, map rasterization [5] and map-to-image coordinates transformations). Correspondingly, global motion planners use data from the global cost map to compute a global trajectory from the current position of the vehicle to a certain goal location [6], as shown in Fig. 8. Local cost maps (shown as the colorful portion of the large obstacle in Fig. 8), are built around and travel with the vehicle, using only the most updated sensor data available. This in turn allows the vehicle to avoid obstacles as their positions update in time.

In order to get the most out of the ranging and mapping capabilities from the Velodyne VLP16 3D LiDAR, a singular configuration of the move_base node was implemented. Point cloud data was acquired from the sensor driver in order to be used for marking instances of new obstacles and performing an artificial conversion of the point cloud data into laser scan information using the pointcloud_to_laserscan ROS package. The computation of navigation trajectories not only rely on the position of the obstacles, but also on the knowledge of free and unknown-state cells in the map. Three potential states of the map cells can be inspected from Fig. 8, with cells depicted in black, white and grey, for occupied, free and unknown states, respectively.

Global and local trajectories are defined parametrically different. While global trajectories (from global planners) are

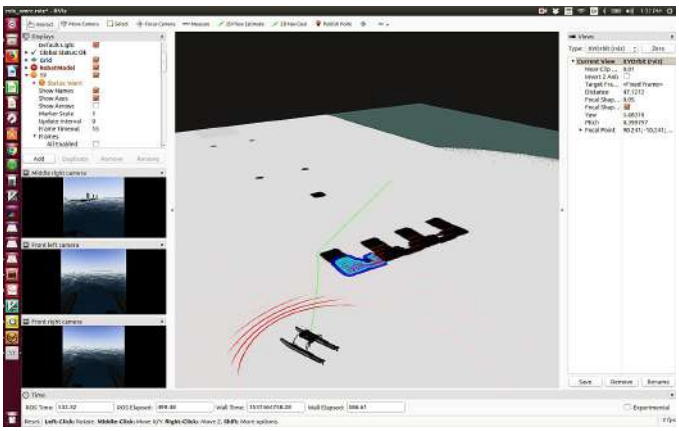

Fig. 8. Global path-planning

defined as a sequence of points comprising the desired (global) path, the local motion planner computes linear and angular velocity commands as inputs for the actuators of the vehicle, from which the actual local trajectory of the vehicle unfolds [7]. The desired trajectory generated by the local planner is generated according to a selected holonomic or non-holonomic motion model of the vehicle, as defined in the move_base package, which in general are associated with motion models of ground robots.

However, the dynamics and the medium associated with the operation of a USV greatly differ, so some adaptations are necessary. Provided that the low-level controllers implemented in the USV effectively control the vehicle state (defined in terms of position, speed and orientation), one possible adaptation is to recover the local trajectory as a sequence of points in a horizontal plane. This can be done by using the velocity commands computed by the local planner as input parameters for a non-holonomic kinematic motion model, from which it predicts the most suitable local trajectory. It also accounts for the original global trajectory and goal location, all of which are weighted inside an optimization function.

Simultaneous Localization and Mapping (SLAM), was also explored for some time. The hector_slam package from ROS was implemented because it did not require odometry data [8], since this can be a problem in marine vehicles. It relies entirely on the information provided by the LiDAR at a high update rate. This approach was tested on ground and marine operations, successfully providing information regarding the pose of the vehicle while simultaneously mapping the surroundings. The success was dependent upon the number of features in the environment, preferably of a structured nature, such as buildings, walls, or large ships. However, given the scarcity of these features in regular conditions in the marine environment, this approach was swiftly discarded. A move towards the more traditional way of solving the localization problem was chosen: mapping the environment based on the localization information provided by the fusion of GPS and IMU data.

B. AUV Design

As a club, the team participates in several AUVSI competitions. As such development of a vehicle to compete in RoboSub has been in progress for over a year. It was a simple task to revisit the platform and modify it to accomplish tasks for RobotX. The secondary vehicle system consists of two parts: Hedy (our AUV) and the Launch and Recovery System.

1) Mechanical: First and foremost, the volume of the vehicle needed to be shrunk. When the original vehicle was conceptualized, the plan was to maximize our size to give us the greatest adaptability possible. This meant that the original design was far too large to use on the WAM-V. To solve this, we decided early on to discard the original frame and create a new one that makes launch and recovery easier and minimizes the vehicles footprint. Importantly, the thruster count would remain the same which means previously performed work on control systems would still be useful. The frame was designed to take advantage of the flexibility and ease of fabrication of PVC. We found that PVC is robust enough to use as a protective frame for our electronics enclosure and rigid enough to withstand a reasonable amount of force exerted on it.

The next task was to determine the locations of the thrusters. Six Blue Robotics T200 thrusters were used to achieve full 6DOF (six degrees of freedom) movement. One important change from the first design to the second was the placement of the surge thrusters. Initially, the surge thrusters were placed near the rear of the vehicle, these two thrusters provide us with rotation. Having them placed at the rear of the vehicle meant the axis of rotation is at the midpoint between them, causing the vehicle to move in a motion reminiscent of a shopping cart. Pushing these two thrusters forward to the center of the vehicle, now allow rotation about the central axis of the vehicle, a simple change that greatly increased the maneuverability of the AUV.

With the thruster placement settled, the shape of the frame was designed. 3D printed thruster mounts were created and printed in PETG to allow for an inexpensive yet robust part that we can create on demand in a few hours. Printing these pieces at 50% infill ratio also meant that these are close to neutral buoyancy. A similar part was used to mount our electronics enclosure to the frame.

The final piece of the vehicle was the acoustics electronics box. The design process for the acoustic pinger housing was straightforward, it had to be able to house the acoustic electronic assembly, the power source, and our extra receiver cable. The starting point of the design was figuring out how the internals were going to be laid out within the housing in order to keep it as compact as possible while still having room to comfortably route the necessary wires. Fig. 9 shows the frame and electronics casing for the AUV.

In order to start figuring out how all the components would go together in the most compact way each part was blocked out to show mounting points and heights of any component over one millimeter tall. After this was completed, low profile mounting brackets were designed so the components could be neatly stacked upon one another without interference. Once

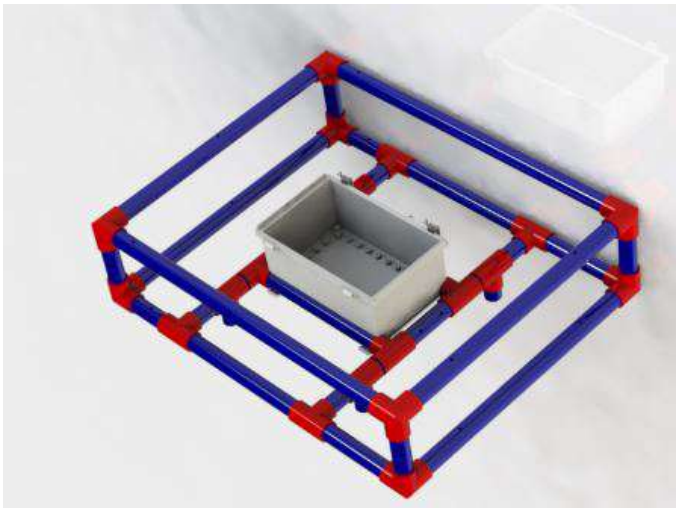

Fig. 9. The framework for Hedy

this was completed the initial housing sketch was done in SolidWorks (shown in Fig. 10 and Fig. 11) and adjusted to the minimum height and width necessary to allow cables to be routed to the main PCB. After these constraints were met then it became a matter of lengthening the housing to give room for the battery and the extra wire to snugly fit within. Once the inside profile was finished a 3d printable insert was designed to secure all the components to the box. This was done so that it could be used in future tasks with minimal alterations to the main housing.

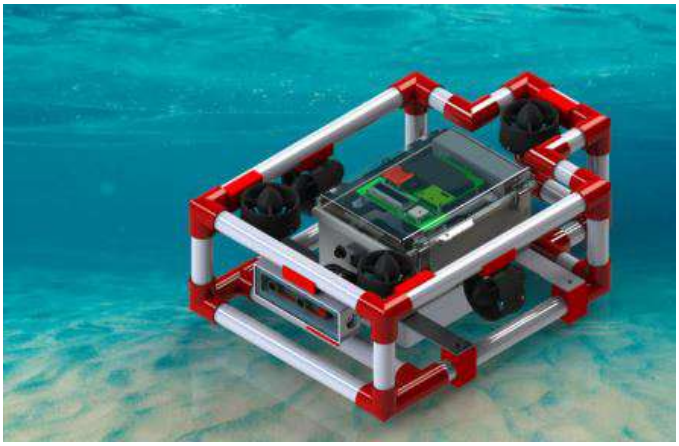

Fig. 10. An isometric view of Hedy

2) Electrical: Hedys electrical system in Fig. 12 is at the core of power distribution and communication for the vehicle. A multi-level bootup process allows the system to be powered on in a safer and more controlled manner. Many components were selected from those readily available for a cost-effective system that is still able to perform at a high level.

a) Power System: An intuitive multi-level power system provides battery power to all system electronics. When the magnetic reed switch is activated, a low-power solid state

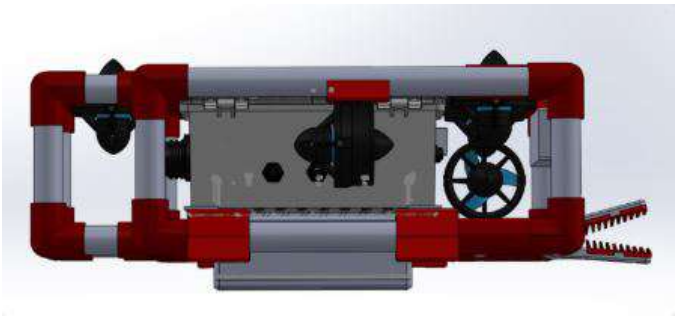

Fig. 11. A side view of Hedy

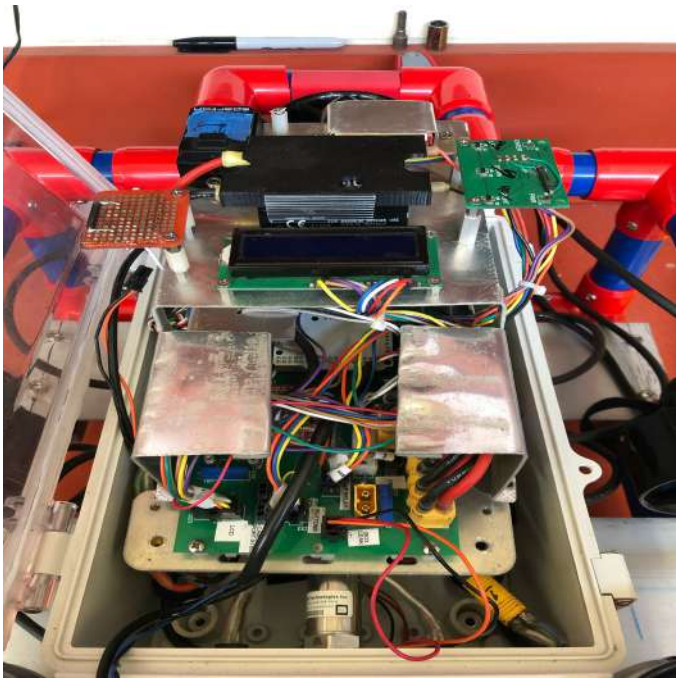

Fig. 12. A view of Hedy's electronics tray

relay connects the low-level microcontroller to power. This microcontroller then takes control of bringing the system online.

Due to the use of various capacitors in the electrical circuity, the system has high in-rush current. The capacitors help maintain stable voltages and reduce sensory noise. The large flow of current can damage several system components, so a clever solution was needed. The initial power up stage is through a low-power relay with a power resistor to limit the amount of current through the circuitry. As the capacitors charge up, the low-level microcontroller completes several tasks to initialize the system. After this short delay, it activates the high-power relay that bypasses the power resistor and allows the system to pull 100 amps from the battery. This high-power relay is also designed to fail if the system attempts to use more current than the battery can provide.

Hedy uses a custom 4S 10Ah LiPo battery pack that provides a nominal 14.8V to the system. The full battery

voltage is used to power each of the 6 Blue Robotics T200 thrusters and the i3 Intel NUC. The power system also has a 3.3V and 5V rail to power various sensors. The microcontroller continuously polls the battery voltage to protect the battery from undervoltage fault. The system is halted if the battery falls below 3V per cell, or 12V for the pack.

b) Processors: The low-level microcontroller is a 32-bit Atmel SAM3X8E ARM Cortex-M3. For ease of development, an Arduino Due board is used to interact with the SAM3X chip. This board breaks out 12 analog input pins, 2 analog output pins, and 54 digital I/O pins of which 12 are PWM output.

The Due is paired with a small form factor i3 Intel NUC. The NUC is responsible for all computationally intensive work such as vision processing. It also acts as a ROS node for connection with external PCs to debug and log various processes and sensory readings.

c) Thruster Control: Hedy uses 6 T200 brushless thrusters rated at 25 amps max. Each thruster requires its own electronic speed controller (ESC) to control the power level. HobbyKing 30A Boat ESCs were initially chosen for this purpose, but it was discovered bad heatsinking made them fail prematurely. All 6 ESCs were upgraded to HobbyKing 50A Boat ESCs that have several programmable parameters. The low-level microcontroller provides the signals required to control each ESC. Fig. 13 shows the layout of the ESCs.

Fig. 13. ESC Layout

d) Sensors: The system uses several key sensors to gather information about its environment. A Zed Mini stereoscopic camera is used to extract 3D information from the vehicles surroundings. Hedy is equipped with an external pressure sensor that is processed by the ADC on the Arduino Due to derive the vehicles depth. A robust Sparton AHRS-8 with 3-axis magnetometer, 3-axis MEMS accelerometer, and 3-axis MEMS gyroscope provides the vehicle with accurate heading, pitch, and roll angles.

e) Communication: Serial UART communication is the main protocol used throughout the system. The low-level Arduino Due microcontroller has 4 dedicated UART ports. These ports are used for communication with the debug console, AHRS, and the remote control. The due also has a native USB port that allows direct serial communication with the SAM3X chip. This port is used for a fast serial connection between the Due and Intel NUC.

3) Software:

a) Arduino Code: The Arduino Due is programmed with the open source Arduino development environment based on the Processing language and Wiring framework. However, the IDE accepts standard C++ code as the avr-gcc is the target compiler. Arduino code is divided into two main functions: setup and loop. The setup function is called when the microcontroller is initially powered on. The loop function runs forever following competition of the setup function. The loop function is similar to setting a while loop permanently to true in modern programming languages.

The setup function is used to initialize the DUEs I/O pins and communication lines. It also provides feedback using an LCD and an RGB LED. The setup function is responsible for controlling the transition to the secondary power level as described in the Power System section.

The loop function is responsible for all low-level control. All sensor management (reading and processing) happens inside of this function along with mission planning and thruster power configuration. The loop also manages the user menu which allows manual selection of several missions, tests, and settings. The menu is displayed on the LCD and navigated by 3 interrupt driven reed switches tied to next, back, and select commands.

To set a stable control frequency, the loop run time is measured and a delay is added at its completion if necessary. The loop period was selected to be 100ms which yields a control frequency of 10Hz.

b) Control Code: Hedy uses a P controller to determine its thruster power levels. The AHRS is polled to get the vehicle angles and checked against corresponding set point values to calculate errors in each of heading, pitch, and roll. The depth sensor is used to determine the vehicles error in depth. Each error in is fed into the P controller which outputs the power levels of the six thrusters.

c) Vision Code: Hedys software is split between the Arduino Due Microcontroller, and an Intel NUC. The vision processing from the stereo cameras feed is handled on the Intel NUC, while mission planning, and sensor interfacing is done on the Arduino Due.

The frames of the ZED Mini Camera are obtained using OpenCV. The ZED Mini combines both images into one frame before it is sent to the computer, so the frame must be split in half and converted to their respective frame. Once this is done, the left and right frame are sent into the ROS network. The ROS network implements the OpenCV Stereo Vision Processing node that takes the two frames and generates a disparity map useful for objection detection.

III. Competition Strategies

This section explains our task strategies, in order of priority, that we planned to complete. The vehicle is expected to enter the test course and perform a lawnmower pattern, autonomously classifying task regions, and then use the proper code to complete them in a particular order.

The team decided that to avoid the Detect and Deliver task. Not only does emitting this task simplify the vehicle design, it streamlines the logic when the vehicle detects the task elements associated with the dock.

A. Entrance and Exit Gates

1) Beacon Localization: The acoustic signal received by a hydrophone can be described by:

$$s_i(t) = s_0 sin(2\pi f_p(t - t_i)) \tag{4}$$

where i denotes the specific hydrophone. Whereas time difference of arrival methods can be derived [9], the method used relied on the an analogous phase difference to determine the bearing of the acoustic signal. If there is a pair of hydrophones, then the fast fourier transform (FFT) of each received signal can be calculated (F_1 and F_2). The ratio of the FFTs can be expressed as:

$$\frac{F_1}{F_2} = |\frac{F_1}{F_2}|e^{j(\phi_2 - \phi_1)} \tag{5}$$

The phase differential can therefore be calculated by solving:

$$\phi_2 - \phi_1 = arctan[\frac{Im(\frac{F_2}{F_1})}{Re(\frac{F_2}{F_1})}] \tag{6}$$

The bearing of the acoustic signal is typically expressed for time difference of arrival methods as:

$$\theta_{12} = arcsin[\frac{c_w(t_2 - t_1)}{L}] \tag{7}$$

where c_w is the speed of sound in water and L is the distance between hydrophones, or the baseline length. L should be the half the minimum wavelength.

The phase difference is related to the time delay via:

$$\phi_2 - \phi_1 = 2\pi f_p(t_2 - t_1) \tag{8}$$

Therefore, substituting Eq. 8 into Eq. 7 creates the following equation:

$$\theta_{12} = arcsin[\frac{c_w(\phi_2 - \phi_1)}{2\pi f_p L}] \tag{9}$$

Using two pairs of hydrophones and performing least squares analysis allows the system to resolve a 2D bearing to the acoustic signal. Fig. 14 shows a 2D projection of the hydrophone array.

In previous competitions, the motors have produced frequency disturbances within the 25 - 40 kHz range. While in competition, the motors would be turned off and the vehicle was allowed to freely drift during the acoustic bearing calculation. In order to avoid this, software and hardware filtering solutions were explored.

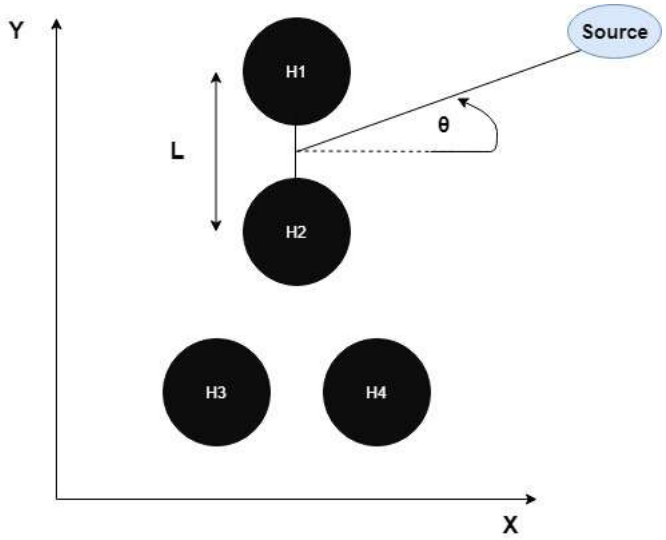

Fig. 14. A diagram of the hydrophone set-up (not drawn to scale). The distance between hydrophones should be set to half the minimum wavelength seen by the system to ensure it stays within - π to + π wrapping

2) Objection Detection: The data from the LiDAR sensor enters the AHC algorithm to provide detection of important objects such as buoys associated with the challenge. These buoys are isolated from the larger point cloud.

For the isolated objects, the parallel camera system projects the RGB values onto the object point cloud, creating a sparse representation of object with it's correct color. This is be important to detect the buoys associated with the gates (green, white, red) as well as for the circumnavigation points (black, yellow, light tower). In the semifinals and finals, the light tower shape is differentiated from the buoys using AHC. Furthermore, the vision system detects if the light tower is on by looking for a change in color content.

3) Autonomous Navigation: Once the objects have been detected, in this case the buoys for the gates, the midpoint can be determined. Using this midpoint, the track for the vehicle can be determined to enter the given gate. Once through the gate, the vision system is used again to detect the object that must be circumnavigated.

During the qualifying rounds there are two different colored buoys. A camera will be used to provide the color of the buoys. Once the desired buoy is determined, waypoint navigation is used to control the track for the vehicle around the buoy. In order to determine the waypoints around the buoy half of the total width of the WAM-V is used, which is then multiplied by 1.5 for safe measure. Using this value, four waypoints are created around the buoy in a clockwise fashion.

During the semi-final and final rounds one of the colored buoys will be replaced with a light tower. The light tower will be checked first. If the detection system determines that the light tower is on, then the waypoint navigation system will be

used to circumnavigate the light tower. However, if the light tower is off, then the vehicle will circumnavigate the colored buoy using the same waypoint determination above.

B. Avoid the Obstacles

The application of algorithms for object detection and autonomous navigation (described in the previous section) is rather straight forward for this task. Classification can be used to distinguish between different types of buoys so that canned buoys can be recognized apart from spherical ones. The current strategy would therefore also hold up in the finals, when this task is integrated throughout the course.

C. Underwater Ring Recovery

The winch system described in Section II will be used to deploy the AUV into the water after the LiDAR has been used to locate the buoy associated with the challenge. After deployment, the disparity map described in section II.B.3.b can be used with blob detection to locate the rings. By creating separate blob detectors for different inertia values, the angles of the ring can also be determined.

Then the small passive gripper with angled teeth is shown in 11 is used to collect a localized ring by moving the vehicle forward until a ring is captured. As designed, the teeth prevent any captured ring from falling back out. Therefore the AUV can be reversed, trigger the pin to release the ring from the structure, and then be recovered via the winch subsystem.

D. Find the Totem Poles

Combining color detection of the camera with the detection and navigation techniques described in the entrance and exit gates section, the circumnavigation of the totem poles in the proper order can be completed.

E. Scan the Code

The object detection and classification techniques allow the vehicle to navigate to the light tower, where it will apply the same change of color detection algorithm that is being used in the entrance and exit gates challenge. Combining and fine tuning logic, the light tower pattern can be recorded.

F. Identify Symbols and Station Keeping

Unlike previous techniques relying on the LiDAR, this classification will rely predominately on the camera, which will leverage shape and color classification. Once the proper symbol is determined, the vehicle will use it's LiDAR based objection detection to generate a waypoint within the docking bay and then will switch to the sliding-mode controller for station keeping. If there is an issue with the symbol classification, a default case will take over and cause the WAM-V to arbitrarily pick a dock to station keep within.

IV. Experimental Results

A. Test Course Construction

Before going to the competition, systems had to be tested. Therefore, based on the information given by RobotX, a test course was created replicating the various challenges. Buoys, a docking station, and a light tower were needed. An easy and cost-effective way to get them was to make them. The can buoys, the dock, and the light tower were mainly made of PVC, as shown in Fig. 15. For the Docking station, PVC couplings were also used so that the large structure could be dismantled and assembled as needed. Then using the information for the RobotX forums, the light for the light tower was built using RGB LED matric panels and Arduinos Pro Mini. The can buoys and the light tower used pool-noodles and weights for buoyancy control.

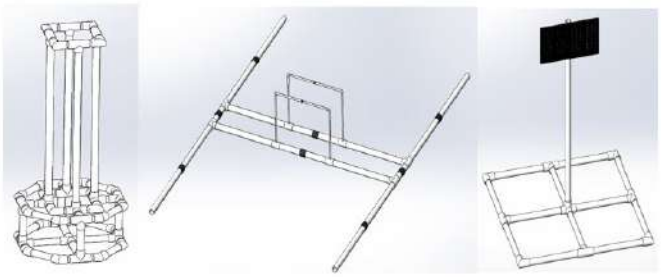

Fig. 15. From left to right: canned buoy, dock, and light tower constructions

B. Simulation

The simulated environment was used to integrate and continuously test mission planning along with path planning, object detection, and obstacle avoidance. In Fig. 16, Vehicle has used the simulated LiDAR to detection two buoys, designated by the black dots. The blue circles that radiate from them are margins of safety to ensure the vehicle width does not cause a collision.

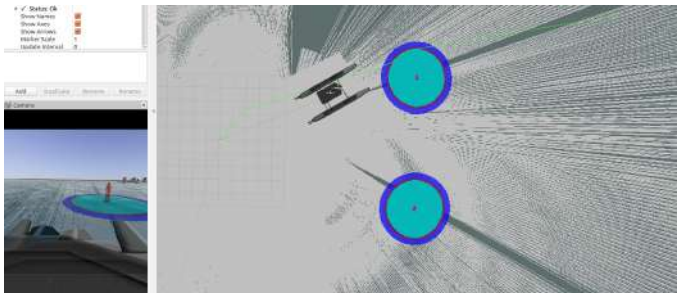

Fig. 16. Path planning in simulation environment, generated path shown in green

C. On-Water Testing

Several tests surrounding the controls systems and basic mission planning were tested at North Lake in Dania Beach, FL. Fig. 17 shows a line of sight controller, which failed to reach steady state under the 15 to 17 knot winds. Since strong

winds caused issues in the previous competition, there was a need for a more reliable controller in such environmental conditions. Fig. 18 shows a solution through enclosure based style heading [2], which exhibited much better control in response to wind disturbance. The success allowed the vehicle to complete a proper lawn mower pattern, which is crucial for our approach to the RobotX competition.

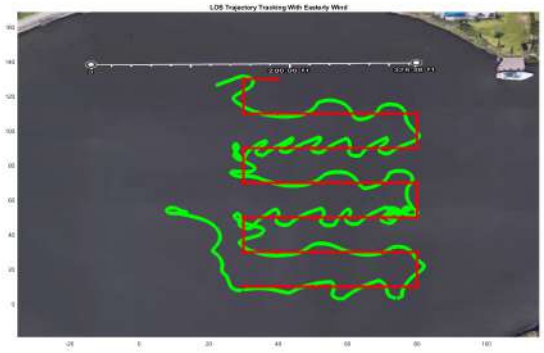

Fig. 17. Line-of-sight style heading and velocity control with 15-17 knot westerly winds

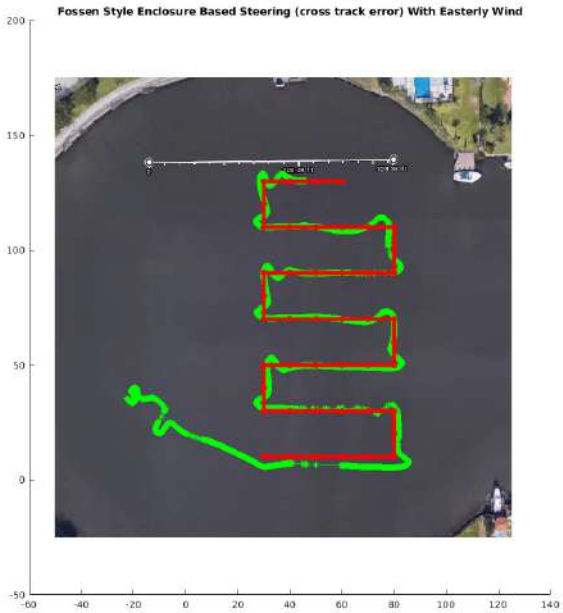

Fig. 18. Enclosure based steering style heading and velocity control with 15-17 knot westerly winds

V. Conclusion

Preparation for the 3rd biennial RobotX Challenge produced some satisfying results for our competition team - especially in electrical and software subsystems. Besides updating some sensors, the subsystems were integrated and tested together more so than in previous years, especially when considering navigation, obstacle avoidance, and path planning with the WAM-V vision system.

Other notable accomplishments were updates to the controls system, test course construction updates, and some updates to the acoustic localization logic. Overall, the team felt confident that its systems were developed with the ability to show high-level autonomy in this year's challenge.

References

[1] Bianca Mesa, Travis Moscicki, et. al, "Team worx: Journal paper for the 2016 robotx competition," Boca Raton, FL, 2016.

[2] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley Sons, Ltd, 2011.

[3] W. H. E. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of Classification*, vol. 1, no. 1, pp. 7–24, Dec 1984. [Online]. Available: https://doi.org/10.1007/BF01890115

[4] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1355–1361.

[5] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 305 – 324, 1979.

[6] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 300–307.

[7] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics Automation Magazine*, vol. 4, no. 1, pp. 23–33, March 1997.

[8] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, Nov 2011, pp. 155–160.

[9] A. K. Tellakula, "Acoustic source localization using time delay estimation," Master's thesis, Indian Institute of Science, Bangalore, India, 2014.