Design of Vessel to Compete in the 2016 Maritime RobotX Competition

Matthew Begneaud, Ethan Davidson, Ronald Kisor,
Chandler Lagarde, and Joshua Vaughan
University of Louisiana at Lafayette
Department of Mechanical Engineering
Lafayette, Louisiana, USA

***Abstract*—This paper covers the systems and techniques developed by the University of Louisiana at Lafayette team to enable autonomous navigation and task completion in the Maritime RobotX competition.The competition involves developing an autonomous marine system using a pre-existing marine craft as a development platform: Marine Advanced Research's Wave Adaptive Modular-Vessel. The vessel must autonomously navigate environments while avoiding obstacles and identifying targets which dictate goals and give instructions for other tasks. The system must be equipped with various sensors and subsystems to be capable of completing the competition tasks. These capabilities include environment mapping, object and color detection, acoustic-pinger localization, projectile launching, and underwater vision.**

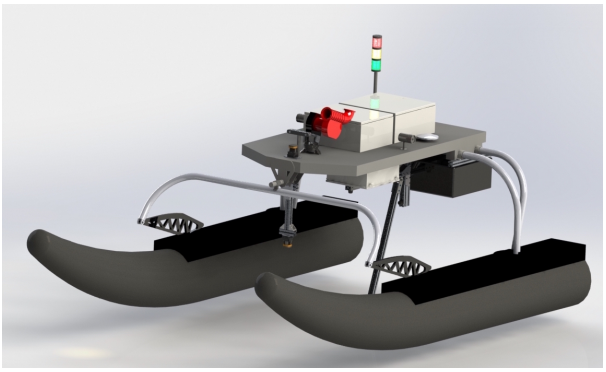

Fig. 1: 3D Render of the UL Lafayette WAM-V (Propulsion System Not Shown)

I. Introduction

The University of Louisiana at Lafayette (UL Lafayette) team primarily consists of four undergraduate seniors in Mechanical Engineering, whose senior project is to develop and program the WAM-V system to compete in the Maritime RobotX competition. The competition gives students the opportunity to gain experience in the field of autonomous, unmanned vehicles. Teams may consist of college undergraduate and graduate students, faculty advisors, and even high school students. The experience of working on a complex system, such as an autonomous marine vessel, that is capable of completing the tasks in this competition is invaluable to students and exposes them to one of the quickest growing fields in engineering and technology.

The UL Lafayette team began work on the system in the Summer of 2016. The systems were designed and developed using commercially available products as well as custom designed parts by utilizing rapid production methods such as water jet fabrication. A 3D CAD model of the UL Lafayette WAM-V can be seen in Figure 1.

Because the system is operating in a marine environment, the components were designed to be watertight and weather-proof. The system was also designed for future research use in mind. In order to take full advantage of the capability of the vessel's dynamics and potential maneuverability, the system utilizes propeller angling as well as propulsion differential to steer the vessel. To enable environment mapping, the vessel has two LiDAR sensors, a forward-facing stereo camera, and mono cameras in the starboard, port, and stern directions.

The system utilizes a distributed computer architecture with all computers communicating over a network utilizing the Robot Operating System (ROS) [2]. Utilizing ROS allows for the use of many previously developed packages which speed up the development of the system and supports the use of a distributed architecture. Keeping computing systems separate also minimizes system computation lag and increases robustness by isolating failure points in the system.

In the next section, the strategy behind the overall system design will be discussed. Then, in Section III, the physical design of the system will be covered. Section IV will discuss the computing architecture and software design of the system. Testing methods will be covered in Section V. Finally, conclusions will be made in Section VI.

II. Design Strategy

The systems that were added to the vessel were designed to be simple and modular. For example, the onboard launcher system is separate from the rest of the computing and electronic systems on the vessel. Using this approach prevents electrical, hardware, and/or software failure from inhibiting the other systems in operation, contributing to the robustness of the system.

The approach for completing the competition tasks is to use environment mapping packages provided by ROS while using OpenCV [1] and machine learning to identify targets of interest relative to the tasks. The navigation control system utilizes GPS and IMU sensors for positioning and heading feedback, as well as LiDAR sensors for point-cloud mapping to supplement localization of the vessel in its environment. ROS navigation packages are utilized for general navigation, while more specific movements and trajectories are handled by custom algorithms. The utilization of pre-written packages from ROS and OpenCV allow for quick development using tested and proven methods, allowing the team to make quick progress.

III. Physical System Design

The components of the system are kept modular and water-tight to allow for robust and safe operation. The system is also designed to allow for modularity for future use of the vessel as a research platform. The vessel utilizes Torqeedo electric motors for propulsion, taking advantage of propeller angling and propulsion differential to increase maneuverability. The propulsion and computing systems are powered separately, which minimizes potential damage in the case of electrical failures and minimizes system reinitialization time after the use of a system kill-switch. The mechanical subsystems on the boat are also powered separately; these systems include the projectile launcher system and the retractable underwater camera system. The payload plate configuration of the systems are shown in Figure 2. Each subsystem is described in more detail in the following subsections.

A. Enclosures

The WAM-V has five main enclosures onboard which can be seen in Figure 3, including a permanent enclosure, a removable computing enclosure, two enclosures housing the propulsion system power, and an enclosure housing components necessary for operation of the launching system. All enclosures are IP-67 rated and utilize waterproof wiring glands to seal out moisture. The permanent enclosure houses components that are typically always used on the vessel, such as: power for the computing systems, power for actuation of other sub-systems, GPS/INS sensor, IMU sensor, networking

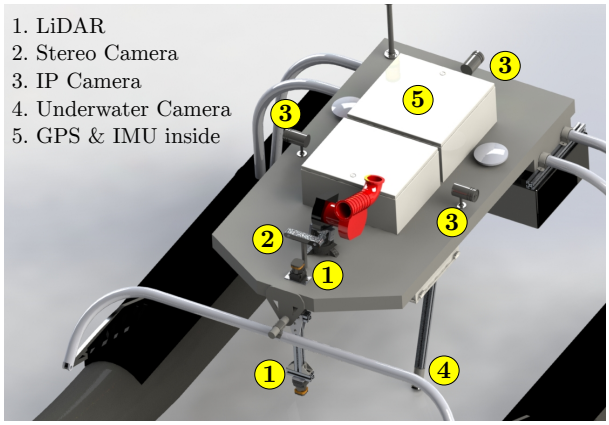

Fig. 2: 3D Render of Payload Plate Layout

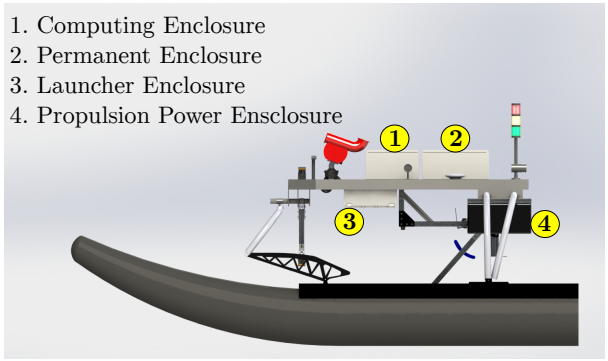

Fig. 3: 3D Render of Enclosure Layout

routers, and motor control computers. The computing enclosure houses most of the computers for the system, including four NVIDIA Jetson TX1 computers, one Raspberry Pi computer, and the audio interface used for the hydrophone system.

The launcher enclosure houses a Raspberry Pi computer for aiming the launcher and an AC power inverter to power the launcher motor. The two propulsion enclosures each house two twelve volt batteries configured to supply twenty-four volts to each Torqeedo motor.

B. Sensors

The configuration of sensors can be seen in Figure 2. Sensors used on the WAM-V include a Hemisphere Vector VS131 GPS/INS and a MEMSIC 350 Series IMU. Environment mapping and object identification are achieved using two Hokuyo UTM30LX-EW LiDAR sensors, a forward-facing Stereolabs ZED stereo camera, and three Foscam IP cameras for starboard, port, and stern vision. One of the Hokuyo sensors is used for general contour detection for mapping, while a second Hokuyo LiDAR is mounted directly below the first in

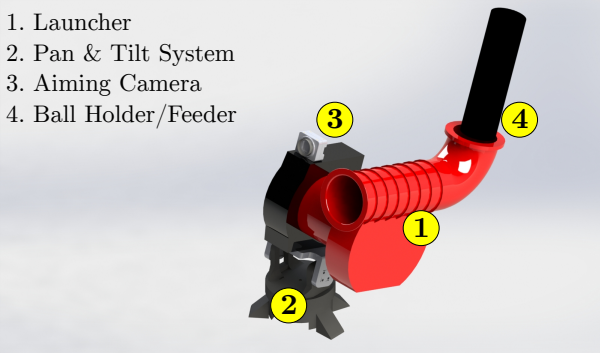

Fig. 4: 3D Render of Launcher System

a lower plane of view for detection of small buoys and docks. The ZED stereo camera aids in point-cloud mapping ahead of the vessel, as it is capable of depth perception.

C. Launcher System

The launcher system shown in Figure 4 utilizes a commercially available pitching machine, modified to shoot racquetballs, mounted onto an outdoor surveillance pan and tilt system. The system aims by use of visual servoing with OpenCV, via a small camera mounted on the barrel of the launcher. This allows the system to shoot projectiles accurately without needing to move the vessel and maintain a stable orientation for aiming. This also increases accuracy by compensating for movement of the vessel caused by waves. The launcher enclosure houses an AC inverter used to power the launcher motor and its pan and tilt control system, as well as a Raspberry Pi to control aim and firing of the launcher with a set of relays.

D. Underwater Vision System

The retractable underwater camera system, shown in Figure 5, consists of a large-profile 80/20 pivoting arm and frame. The arm is extended and retracted by use of a winch motor. Retraction of the underwater system reduces the overall drag on the system. Retracting and extending movements are limited by frame-mounted limit switches, which are actuated by a cam on the arm. A submersible camera is mounted on the end of the arm.

E. Acoustic Pinger Localization System

The hydrophone-based acoustic pinger location system consists of two hydrophones, one on the port side and one on the starboard side, as shown in Figure 6. As traveling sound waves emitted by a pinger encounter the

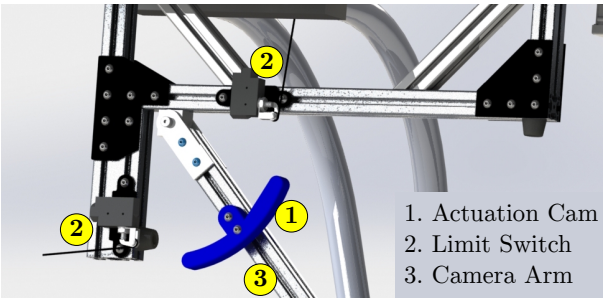

Fig. 5: 3D Render of Retractable Underwater Vision System

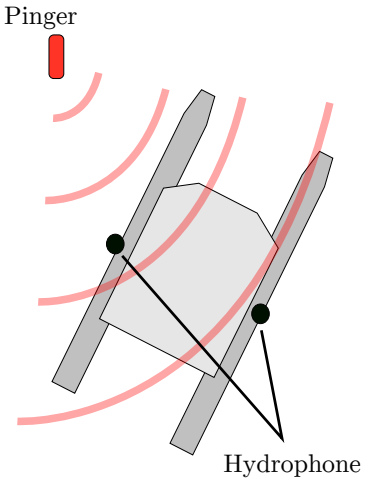

Fig. 6: Acoustic Pinger Localization System

vessel, the signal will be received by the hydrophones at different times if the vessel is not heading directly to or away from the pinger, as can be seen in Figure 6. The system is able to determine the direction of an acoustic pinger by measuring the time difference between signal peaks on each hydrophone. Once the time difference is sufficiently small, the pinger is either directly ahead or behind the vessel. The direction can be checked by turning slightly and seeing which hydrophone is leading the other in signal peak time.

IV. Computing Architecture & Software Design

Subsystems on the vessel are mostly controlled on separate computers communicating across a ROS network, allowing for isolation and parallel development of systems. The use of distributed architecture also allows for faster processing. A diagram of the computing architecture can be seen in Figure 7.

The processing of sensor data from the GPS/INS and IMU and the main algorithm computing is handled

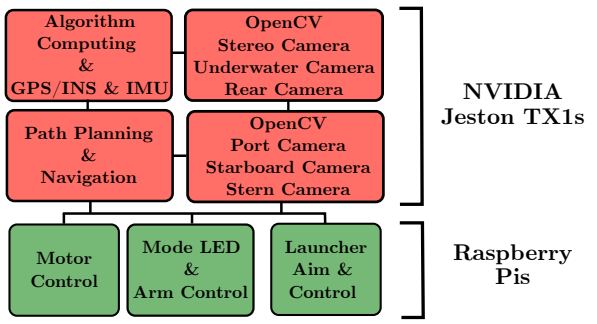

Fig. 7: Diagram of Distributed Architecture

by one NVIDIA Jetson TX1. Image processing with OpenCV from all cameras is split across two Jetson TX1 computers. LiDAR sensors are processed by the same TX1 handling path planning and navigation in order to minimize system lag from transferring large data arrays at high rates over the network. Minor functions such as control of the LED mode indicator and actuation of the underwater vision arm are controlled by a Raspberry Pi computer.

Keeping the software modular allows for easier debugging and allows nodes to be restarted individually without interfering with other systems and functionalities. It is also simpler to develop modular programs, which can then be used to create systems in the form of building blocks which can be recycled from other systems.

A. Visual Processing Methods

Using various OpenCV [1] and machine learning methods, the system is able to visualize its surroundings. The system has been trained to detect buoys and other objects of interest using machine learning. Shapes are also detected using simple contour detection. The color of the object is then detected using OpenCV to correlate the object to the current task and identify targets of interest. Colors are detected using a combination of color blob detection, masks, and thresholds utilizing the Hue-Saturation-Value (HSV) color space to produce binary images [1].

Horizon detection is used to filter background noise in the image, which detracts from object-detection accuracy. Due to the abundance of blue in the competition environment from the sea and sky, it can be difficult to detect blue objects. The sky is handled mostly by the use of horizon detection, and water is filtered out with ripple detection.

Object centroids are compared to the center of the image to localize the targets relative to the vessel. This is useful, for example, in the task of passing through buoy gates. The median of the two buoy centroids can be used

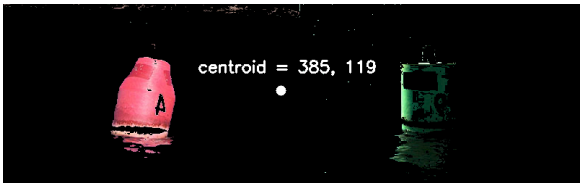

Fig. 8: Processed Test Image of Buoy Gate Median Detection

to identify the middle of the gate as a navigation target, as shown in Figure 8. Using object centroid detection to localize targets with respect to the vessel is also used by the system's custom navigation and path-planning algorithms.

B. Path Planning & Navigation

The WAM-V utilizes the ROS packages for tasks such as environment mapping, odometry estimation, and navigation. The mapping package takes in depth data from the LiDAR sensors and ZED stereo-camera to create a local costmap of the vessel's surroundings. The odometry estimation is achieved by use of a localization package which takes in depth data from sensors as well as data from the IMU and GPS/INS systems to estimate where the vessel is located in its local map.

The ROS navigation package, referred to as the Navigation Stack, receives commands in the form of goal positions in the vessel's odometry frame. The Navigation Stack utilizes costmap and odometry data while also considering the dimensions of the vessel in order to plan the shortest path to the goal that avoids hitting objects. The path planner generates velocity commands which are sent to a base controller that attempts to keep the vessel on the path [2].

Once the vessel reaches or senses targets of interest, the system bypasses the waypoint-based Navigation Stack to directly send velocity commands to the base controller, allowing for more-direct control over the vessel's movement. For example, when circling buoys for tasks, the system switches to using visual processing to dictate movement commands for the vessel. Side cameras are used for feedback to adjust the angular velocity of the vessel with a PID controller to keep the buoy in the center of the camera feed, as seen in Figure 9. The same visual-processing-based approach is used for other task-specific movements such as passing through buoy gates, parking in docks, and approaching targets like the LED-panel-sequence buoy.

V. Experimental Results

The team was unfortunately unable to coordinate significant on-water-testing of the vessel before shipping it

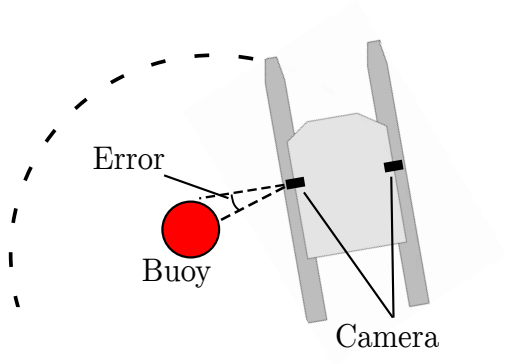

Fig. 9: Depiction of WAM-V Circling a Buoy

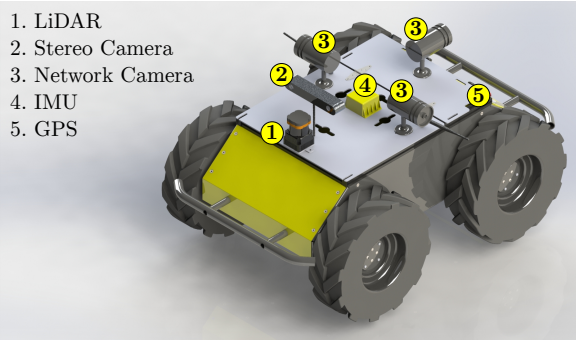

Fig. 10: 3D Render of Clearpath Husky Testing Platform

to the competition site; however, testing of algorithms and navigation was still made possible via a Clearpath Husky rover, which is also programmed with ROS.

The Husky rover was outfitted with a similar sensor configuration to the boat, as shown in Figure 10, and was utilized for algorithm testing. This allows for all algorithms to be written to publish velocity commands and goal positions to the system. The only difference between testing and competition performance is the control of the movement of the vehicle, as the rover and boat differ in their dynamics and motor control in order to achieve desired velocities.

VI. Conclusion

UL Lafayette team's design for the Maritime RobotX competition focuses on modularity of subsystems to increase overall system robustness. The design of the system was highly influenced by a short development period and, therefore, benefitted from modular design due to the ease of parallel development in such systems. The modularity of the systems also aids to the system's overall robustness. The use of a distributed computing architecture isolates main software systems, decreasing the impact of failure points in one system on another. The vessel will also be used for future research.

Acknowledgments

The team would like to thank Mr. Donald Mosing, the UL Lafayette College of Engineering and Department of Mechanical Engineering for providing funding for this project, ASV Global for their technical guidance and lending of equipment, and Road Narrows Robotics and Base Engineering for their sponsorship of this project.

References

[1] Christopher M. Barngrover. Computer vision techniques for underwater navigation. Master's thesis, University of California, San Diego, 2010.

[2] William D. Smart Morgan Quigley, Brian Gerkey. *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*. O'Reilly, 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2015.