Ragin' Cajuns RoboBoat 2021

Captain:
Joseph Stevens
Members:
Nathan Madsen, Brennan Moeller, Adam Smith, Benjamin Willis
Faculty Advisors:
Yasmeen Qudsi and Joshua Vaughan[1]

Abstract— **This report discusses the design choices and improvements made to the University of Louisiana at Lafayette's first entry in to RoboNation's RoboBoat Competition that includes an Unmanned Aerial Vehicle (UAV). This UAV is to be a mobile sensor for the ASV using its onboard sensors, which include a Real–time Kinematic (RTK) GPS system System, a Raspberry Pi Camera Module, and an OAK–D machine vision sensor. The Ragin' Cajuns RoboBoat is a catamaran-style autonomous surface vessel (ASV) equipped with four thrusters in an "X"–configuration, enabling holonomic motion. The contributions to the ASV from the 2021 Ragin' Cajuns RoboBoat team include finishing the upgrades to the new electronics enclosure that were hindered due to COVID–19, adding a RTK–GPS system, and upgrading the previous vision sensors to OAK–D machine vision sensors.**

I. Introduction

The 2021 RoboBoat competition requires teams to build an Autonomous Surface Vessel (ASV) capable of performing various tasks that require several subsystems to function together. The ASV shown in Figure 1 is equipped with two planar LiDARs for depth perception, two OAK–D stereo cameras for machine vision feedback, and a RTK–GPS and an IMU for localization. The vessel is equipped with four thrusters mounted in an "X"–configuration, enabling holonomic motion. This vessel is equipped with two Lithium polymer (LiPo), one 10Ah battery for its thrusters and another 10Ah battery for the electronics. This two battery configuration allows the E-Stops to kill power to thrusters without removing power to the electronics.

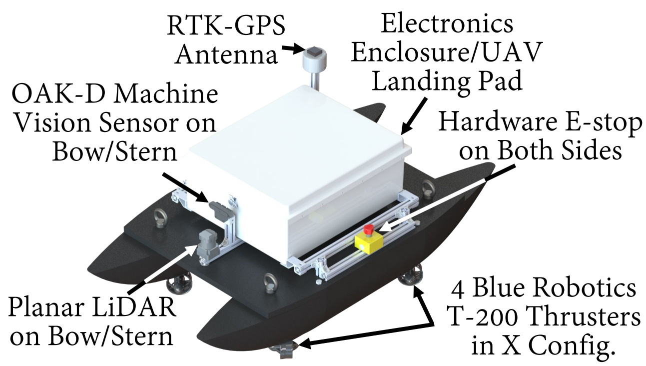

Fig. 1: 2021 Ragin' Cajuns ASV CAD Model

For the 2021 RoboBoat competition, the team added an unmanned aerial vehicle (UAV) to be used as a mobile sensor to assist the ASV in mapping, navigation, and localization. The UAV, shown in Figure 2, is fitted with an OAK–D machine vision stereo camera to differentiate between the various buoys that will be encountered during the competition and to avoid possible obstacles and a downward–facing Pi Cam to find ArUco markers on top of the ASV's electronics enclosure when landing and object recognition. It is also equipped with RTK–GPS, standard GPS, and an IMU for localization. It is also equipped with a downward–facing single point LiDAR–Lite for altitude augmentation, an RC receiver for manual flight, and flotation gear to ensure

[1]Department of Mechanical Engineering, University of Louisiana at Lafayette, Lafayette, LA 70504, USA joshua.vaughan@louisiana.edu

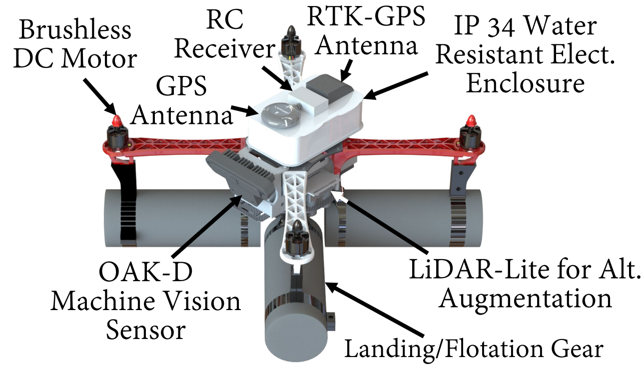

Fig. 2: 2021 Ragin' Cajuns UAV CAD Model

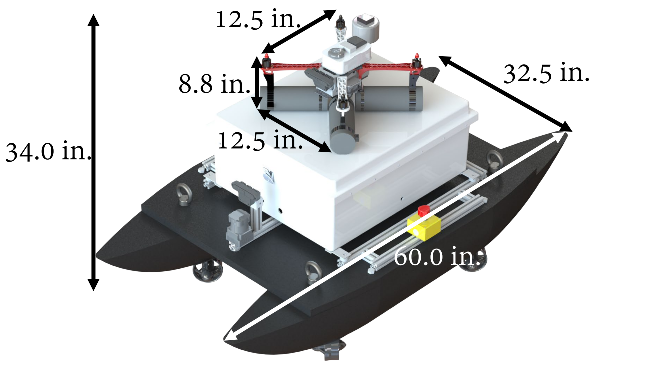

Fig. 3: 2021 Ragin' Cajuns Autonomous System

buoyancy. This UAV is powered by one 9.2Ah, 100C discharge rated LiPo battery. The size of the battery was chosen to give an approximate 17 minute flight time. This is a sufficient amount of time for the UAV perform its necessary tasks. If the battery life is detected to be $\leq 35\%$, the UAV will make an emergency landing. The total 2021 Ragin' Cajuns autonomous system is shown in Figure 3 with its key dimensions highlighted.

In the next section, the competition strategy for the 2021 competition is discussed. The design creativity of the ASV and the new UAV are discussed in Section III. In Section IV, the experimental results that were gathered are presented. Finally, Section V is the conclusion. A list of components is also presented in the Appendix.

II. Competition Strategy

The past competitions have shown that the Ragin' Cajuns ASV is an adaptable and capable vessel for RoboBoat's challenges. However, there was room for improvements including utilizing a UAV in the competition, upgrading localization and sensing components, and improving the control architecture of the system. This year, integrating a UAV and upgrading the localization of the system were selected as the main foci. The following sections will discuss the strategic approach for the UAV integration and how the system will operate in the competition.

A. UAV Integration

Through an analysis of the past competitions, it was determined that the team could potentially obtain an approximate 1,200 additional points by adding a UAV that takes off of and lands on the ASV, and completes the UAV–specific task. The addition of a UAV to the current ASV requires tight integration of software and mechanical design. To facilitate the integration of the UAV's and ASV's sensing, mapping, perception, and controls, the Robot Operating System (ROS) is used. To ensure the designed UAV could be integrated to the ASV, computer–aided design (CAD) models of the UAV and ASV were created and updated throughout the process. While this integration increased the complexity of the system, the ASV can operate independently of the UAV, allowing for each run to continue even if the UAV fails.

For the 2021 competition, the UAV will be deployed after the mandatory Navigation Channel is completed and move to the next section of the course. This allows the UAV to be one task ahead of the ASV to collect data and place waypoints for the ASV to navigate to.

B. System Software

A mix of custom and standard ROS packages are used for the control, perception, and mapping. The *robot_localization* package is used both on the ASV and UAV. It combines GPS and IMU sensor data and pose estimates from *rtabmap_ros* and ArUco markers to compute a single state estimation of each system [1]. The global loop-closure Simultaneous Localization and Mapping (SLAM) package, *rtabmap_ros*, uses the data

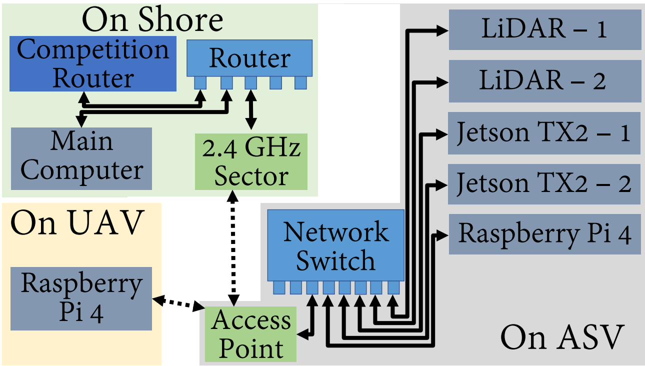

Fig. 4: 2021 Ragin' Cajuns Network

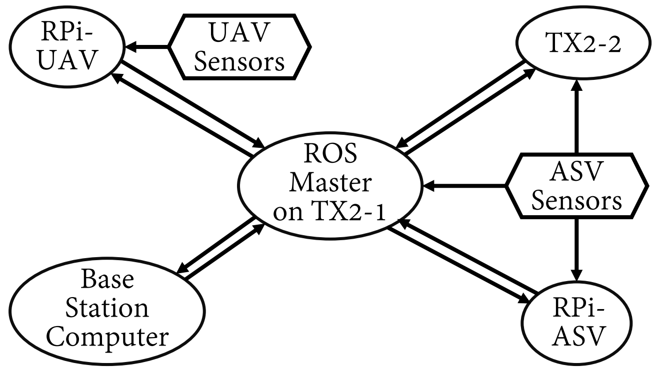

Fig. 5: Graph of ROS Network

collected from the OAK–D stereo cameras and the planar LiDARs on the ASV, as well as the OAK–D on the UAV to create 3D point cloud maps of [2]. The maps created by the UAV and ASV are compiled into a single map using a ROS package called *multirobot_map_merge*, which allows multiple maps from different robots to be compiled into one map [3]. ArUco markers allow for the pose of a camera to be determined from a square marker that is similar to a quick response (QR) square [4]. As stated above, the ASV's lid is fitted with an ArUco marker array. This gives the UAV a more accurate pose estimation when landing on the ASV.

ROS is also used to aid in the overall communication of data for the system by allowing multiple peripherals to share information across computers connected to the network. as shown in Figure 4. The 2.4 GHz sector, or antenna, creates a networking bridge from the on shore router to the access point on the ASV. This access point also gives the UAV access to the network. Figure 5 is a high–level overview of the ROS network for this system, which allows all the information gathered from both the ASV's and UAV's sensors to be shared. For example, the *mavros* package is used to communicate with the flight controller on the UAV and publishes the data to the ROS master that it is recording [5].

Due to the volume of information that is being collected and processed, multiple computational devices are needed. Furthermore, the imagine processing tasks that are required for this competition are best handled by dedicated Graphics Processing Units (GPUs). For this reason two Jetson TX2s are used on the ASV. These devices are used for image processing and mapping. The ASV also has one Raspberry Pi 4 Model B that handles basic sensor reading and lower–level thruster control. The UAV uses a Raspberry Pi 4 Model B as the companion computer for its Pixhawk 4 flight controller.

To speed up configuration, initialization, and booting, Docker was used to create containers that could be implemented on multiple devices [6]. A Docker container contains all of the dependencies, packages, system tools, and settings needed to execute an application. This allows the same container to be run on different operating systems, such as the operating systems contained in the two TX2s or the two Raspberry Pis. Docker also allows the code that is needed to execute certain tasks to be mounted to the container at time of use. This allows the code to be saved to a platform like GitHub, so the development can be split across a team [7].

C. Navigation Channel

Completing the Navigation Channel task is mandatory and must be done before attempting any other tasks in the competition. The complete system must pass through two sets of gates consisting of two buoys at least six feet apart. The two sets of gates are at least 50 feet apart. The Ragin' Cajuns RoboBoat system is equipped with forward facing stereoscopic cameras on both the

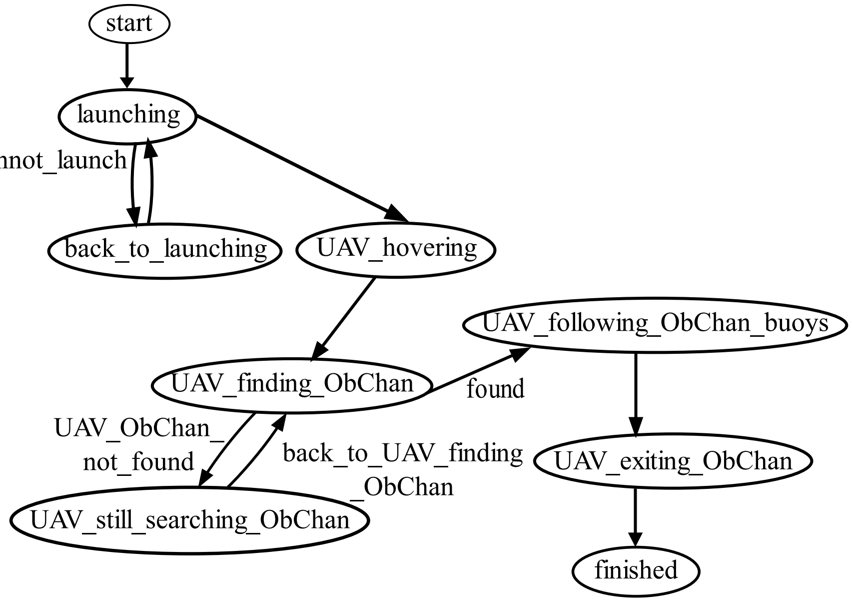

Fig. 6: State Machine for the UAV in the Obstacle Channel Task

UAV and ASV that provide images to a "You Only Look Once" (YOLOv3) Convolution Neural Network (CNN), object detection in real time [8]. The training set for this CNN consists of manually-labeled images from the 2019 competition, as well as images from the 2016 Maritime RobotX Competition. The output of the image classifier is passed to a state machine that determines how the ASV or UAV should maneuver.

For the Navigation Channel task, the state machine directs the ASV to find a gate, identified with green in the right side of the frame and red in the left. A waypoint goal to maneuver to the middle of the gate and orient the vessel to be in–line with the channel is sent to the navigation stack. The state machine instructs the ASV to maintain the initial heading as it continues to drive forward and look for the exit gate. Another waypoint between the exit gate buoy is sent to the navigation stack as a target location. Once the exit gate is found, and the vessel reaches this waypoint, it completes the Navigation Channel and prepares to launch the UAV. Furthermore, if the UAV cannot find the exit gate it will continue to the next provided GPS location, and the ASV's peripherals will guide it through the channel.

D. Obstacle Channel

The Obstacle Channel requires the ASV to maneuver through a series of gates with obstacle buoys along the trajectory. Unlike the Navigation Channel, these gates are arranged in a non–linear path. This task may be difficult to complete because of the ASV's relatively large footprint. However, the holonomic thruster configuration allows the ASV to exert forces and moments in each degree of freedom independently increasing its maneuverability. Furthermore, it is equipped with a RTK–GPS system to help improve localization. This increased maneuverability, increased localization precision, and data collected by the UAV helps the Ragin' Cajuns RoboBoat complete this task.

Once the ASV exits the Navigation Channel, the UAV launches, hovers, and then is tasked with finding the Obstacle Channel via GPS coordinates and object recognition. An outline of the UAV's state machine for the Obstacle Channel is shown in Figure 6. The state *back_to_launching* is invoked if the UAV cannot launch for any reason. Similarly, a redundancy is in place if the UAV cannot find the obstacle channel. There is a limit to the number of times a redundancy state can be entered; stopping the UAV from entering a continuous loop. If this limit is reached for any state other than launching the UAV it will then continue to the next task's GPS location and begin surveying. If the UAV reaches this limit during the launching process, it will send a a message to a computer on shore and reboot its companion computer and re initialize the state machine. Once the UAV finds the Obstacle Channel, it uses the CNN to follow the channel, similar to how the Navigation Channel was identified, green in the right side of the frame and red in the left. The UAV places waypoints approximately in the middle of the channel as it flies above. This information is available for the ASV through the ROS network.

Once the ASV arrives at this task, it utilizes the map that was created by the UAV with its path planning package, *roboboat_navigation*. This allows the ASV to have an understanding of its environment and a desired path to follow through the channel before it arrives at the task. The vision feedback system provided to the navigation stack prevents the ASV from colliding with the green, red, and yellow buoys.

E. Obstacle Field

Once the UAV exits the Obstacle Channel, it moves to the Obstacle Field to create a map for the ASV to use. The UAV uses the GPS coordinate and the CNN to determine the location of the Obstacle Field on this course. It will identify the pill buoy by using the YOLOv3 CNN to determine if the pill buoy is in the frame. Then, the UAV circumnavigates this buoy while updating the distances between the buoys to find the largest entrance and place a waypoint at that location.

When the ASV reaches the Obstacle Field, it moves towards the opening specified by the UAV. The ASV saves the location of the entrance and enters. Next, the path planner uses the map created by *multirobot_map_merge* to plan a trajectory around the pill buoy. Once the pill buoy has been circled, and the ASV has changed its heading by at least 360°, the ASV will exit the Obstacle Field using the recorded position as the last waypoint and head towards the Speed Gate task.

F. Speed Gate

After the UAV finishes mapping the Obstacle Field, the GPS coordinates are sent to the navigation stack to motivate the ASV to move toward that location. As it approaches, the CNN is used to locate the one standalone gate that consists of one green and one red buoy. The UAV then hovers and looks for the marker buoy, a single blue buoy. Once the buoy is found, the UAV places a waypoint at the gate and flies toward the blue buoy where it places another waypoint.

Once the ASV completes the Obstacle Field, it proceeds toward the waypoint that was placed at the entrance of the speed gate by the UAV. The state machine overrides the velocity restrictions that were in place for the earlier tasks where speed was not a priority, so this task can be completed as quickly as possible. Then, the state machine instructs the ASV to pass through the gate at maximum speed. Once the target buoy has been located, waypoints are placed around it so it can be circumnavigated. The initial position at the gate is used as the final waypoint to guide the ASV out of the Speed Gate Challenge.

G. Return to Dock

Once the UAV has finished all of the tasks, it will then fly towards the ASV, guided by their shared localization data. As it approaches the vessel, the downward facing Pi Cam identifies an ArUco tag array that is on top of that ASV's enclosure, as shown in Figure 7. These markers are used to determine the UAV's location in reference to the ASV. Then, the UAV follows the ASV to the starting position and prepares to land.

After the ASV finishes the Speed Gate challenge, it navigates back to the starting location that was recorded upon entering the course. Throughout the competition, the ASV was creating a map of the course. This map is used to guide it back to the starting position while avoiding any of the obstacles or exiting the course. Once the ASV reaches the dock, it holds its position to allow the UAV to land on top of it. With the Pi Cam's estimation, the increased accuracy of the RTK–GPS system, and the LiDAR–Lite altitude

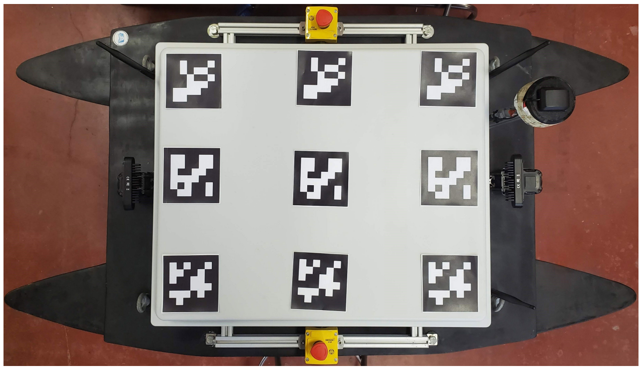

Fig. 7: ArUco Markers On Top of ASV to Aid UAV in Landing

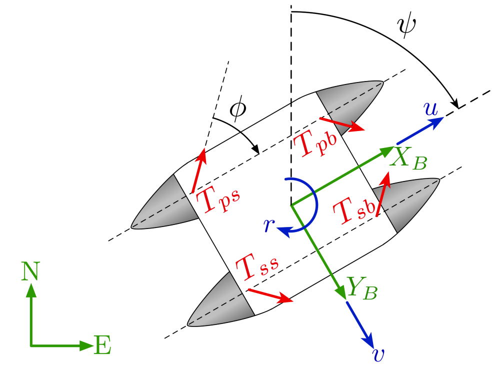

Fig. 8: Free-Body Diagram of RoboBoat Thruster Configuration

augmentation, the UAV will slowly approach the lid of the enclosure and land.

H. Acoustic Docking

The Acoustic Docking task requires the vessel to localize a docking signal and dock itself in the specified docking bay. The 2021 Ragin' Cajuns RoboBoat team declined to continue to develop upon the previous system used for this task. The focus was on to the design, manufacture, and integration of the UAV to the ASV system.

I. Object Delivery

The Object Delivery task requires the ASV to deliver up to four objects to a specified area in the course. The task may be completed solely by the ASV or by a combination of an ASV and UAV. The 2021 Ragin' Cajuns RoboBoat team declined to develop the required mechanism to complete this task. However, the frame selection and custom frame plates for the UAV that were designed with additional mounting holes allowing for future development.

III. Design Creativity

A. Thrust Configuration

The thrusters are mounted to the hull of the ASV in a "X"–configuration, mounted at 45° angles relative to the bow. A free–body diagram of this thruster configuration is shown in Figure 8. This configuration enables holonomic motion, or motion in all three degrees of freedom independent of each other. This improves the maneuverability of the ASV, which is especially valuable in tasks like the Obstacle Channel and Obstacle Field.

Furthermore, the addition of the RTK–GPS system, that receives RTCM (Radio Technical Commission for Maritime-services) data to increase the accuracy of standard GPS signals to centimeter levels, improves the precision of the ASV's localization [9]. This specific thruster configuration allows the ASV to take full advantage of the increased localization precision.

B. UAV Center of Mass

Throughout the design of the UAV, component placement was driven by the effects that the location and mass of the components would have on the COM of the UAV. The location of the COM in reference to the horizontal plane in which the rotors lie has an affect on both the stability of forward flight and the UAV's ability to combat wind disturbances [10]. For this application, a COM below the horizontal plane in which the rotors lie is desirable, because it increases forward flight stability [10]. In its current state, the location of the UAV's COM, relative to the center of the horizontal plane in which the rotors lie, is (-0.025, -2.163, -0.010) in, as shown in Figure 9.

C. UAV Enclosures

The RoboBoat competition takes place regardless of weather conditions. Therefore, it is imperative to keep all electronics safe from rain

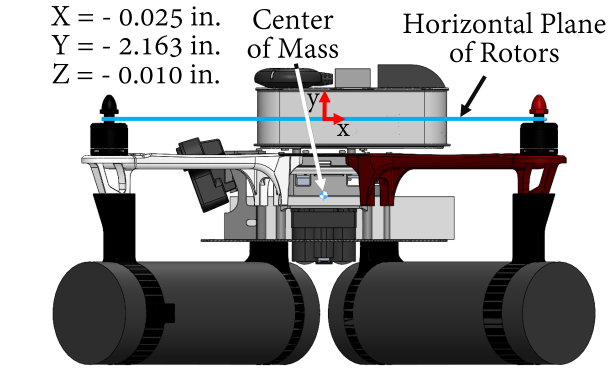

Fig. 9: UAV Center of Mass

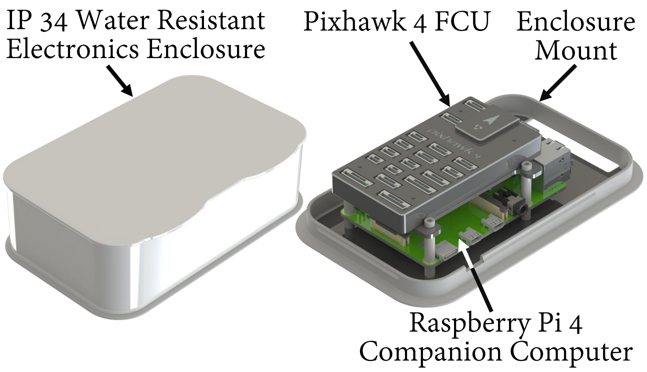

Fig. 10: Main UAV Electronics Enclosure and Mount

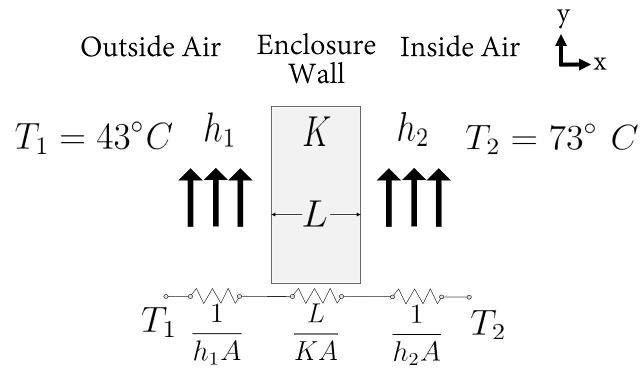

Fig. 11: Model of Simple Heat Transfer Problem

water. This is achieved by fully enclosing them in 3D printed enclosures. These enclosures are also used to directly mount the electronics to the UAV. It was determined that all enclosures must meet Ingress Protection (IP) 34 standards, water resistant from splashes of water in all directions [11].

The main electronics enclosure for the UAV, shown in Figure 10, is made of acrylonitrite butadiene styrene (ABS) because of its UV and weather resistant properties. The main electronics in this enclosure are a Pixhawk 4 Flight Control Unit (FCU) and a Raspberry Pi 4 companion computer as shown in Figure 10. Further considerations were made to ensure other electronics be protected from water ingress such as the Pi Cam, RTK–GPS, and power module. The LiDAR–Lite V3HP equipped on the UAV already exceeded the desired qualifications with an IPX7 rating. To ensure that the main electronics would not overheat or result in CPU throttling, a heat transfer analysis was completed. The problem is modeled as a thermal resistance problem, as seen in Figure 11. The hottest temperature that Florida has ever reached is T_1, 43°C, and T_2, 71°C, represents the average operating temperature of a Raspberry Pi, which is 20-30°C above the ambient operation temperature [12], [13]. This calculation determined that the system would dissipate heat to the environment through the enclosure, but more importantly that the Raspberry Pi's CPU would operate approximately 16.5% under its throttling temperature, 85°C.

D. Data Acquisition

1) Raspberry Pi Camera Module V2: A Pi Cam was positioned on the bottom plate of the UAV to be used as a downward facing image sensor to collect data for object detection and landing procedures. In Figure 12, a sketch showing the analysis of the horizontal field of view (FOV), 62.2 degrees, of the PiCam can be seen. A small amount of the polyethylene foam is in the camera's FOV. However, this amount of interference was later determined to be insignificant.

2) OAK–D FOV: An OAK–D stereo camera will be used by the UAV to capture distance and RGB image data for purposes of mapping, object detection, and obstacle avoidance. As seen in Figure 13, the vertical FOV of the OAK–D stereo camera is 56 degrees. The forward facing camera mount which holds the OAK–D on the UAV is designed with a 20 angle from the vertical, as seen in Figure 14. This angle gives the OAK–D

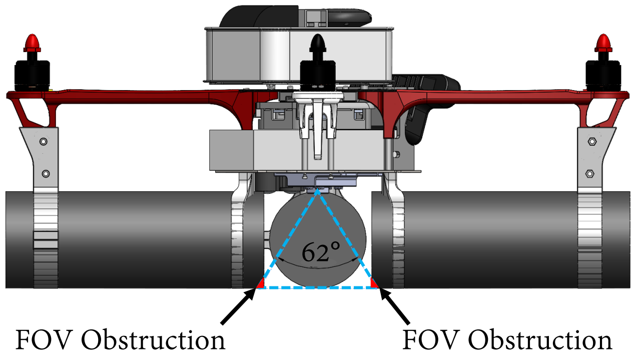

Fig. 12: Raspberry Pi Camera Module V2 FOV

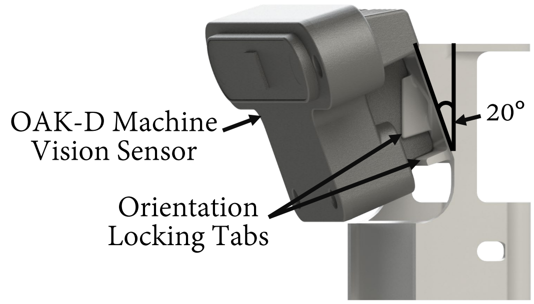

Fig. 14: OAK–D Camera Mount for UAV

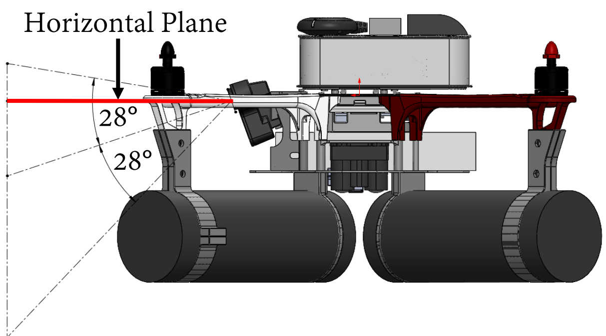

Fig. 13: OAK–D Field of View

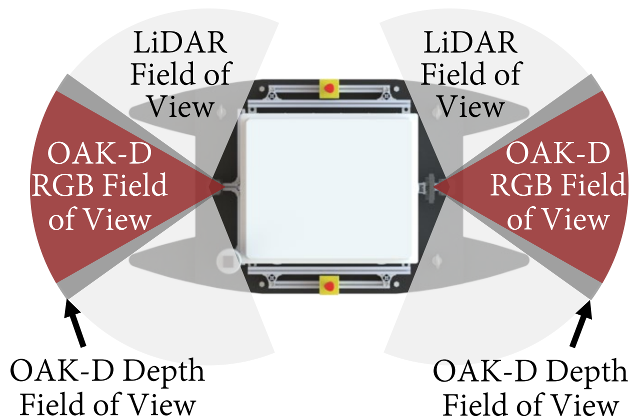

Fig. 15: ASV LiDAR FOV

the capability of capturing image data above and below the horizontal plane providing data for both obstacle avoidance and object recognition, respectively. This mount also locks the cameras position on the UAV allowing for the exact location to be known when the camera is collecting data.

To keep the implementation of code across the system uniform, the ASV's stereoscopic sensors were upgraded to OAK-D machine vision sensors. The OAK–D machine vision sensor was added to both bow and stern adding a 81° RGB FOV. This allows the ASV to take full advantage of its holonomic motion capabilities. Similar to the UAV, the mounting brackets for these cameras allow for their exact position to be known when they are collecting data.

3) Planar LiDARs: The ASV is also equipped with two Hokuyo planar LiDARs for obstacle avoidance and mapping. These LiDARs are positioned in the middle of the ASV on both the bow and stern and offer a wider FOV than the OAK-D's RGB and depth sensing FOVs, as seen in Figure 15. They give the ASV an approximate 230° FOV of its surroundings, front and rear, thus increasing its ability to utilize its holonomic motion. These LiDARS also give the ASV an approximate 64.8% increase in horizontal perception from only having stereo vision.

IV. Experimental Results

A. System Simulation

Models of the ASV and the UAV were developed for use in the *rotors_simulator* ROS package because of its use of a physics–based ROS application called Gazebo [14], [15]. Each sensor that is on the UAV can also be simulated, such that the FOV and data being collected represent that of the actual sensor.

In this simulated environment, all of the sensors were configured to accurately represent their

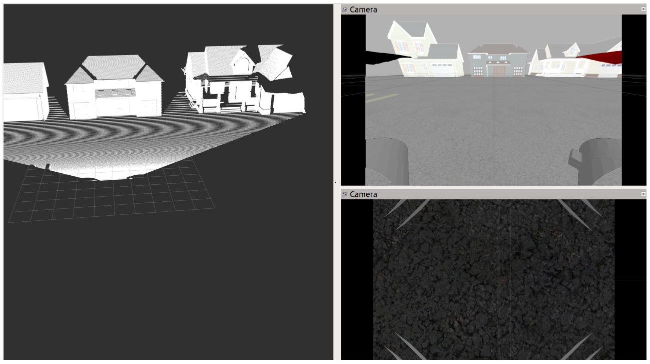

Fig. 16: UAV Peripheral Simulation in RViz

Fig. 17: Testing of Flotation Device in Controlled Environment

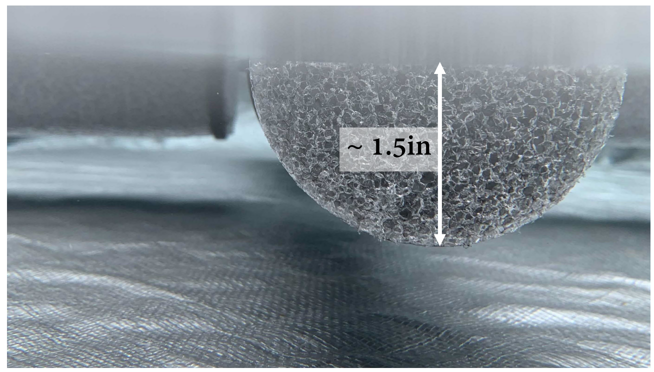

Fig. 18: Submerged Flotation/Landing Gear

specifications. In RotorS, flight paths can be set, scripts that control the perception and mapping algorithms can be executed, and data can be collected. For example, Figure 16 shows depth data collected by the simulated OAK–D stereo camera on the left and RGB image data collected by the simulated OAK–D digital camera on the top right. The bottom right image in Figure 16 displays RGB image data collected by the simulated Pi Cam. This simulation also allows for safer testing during future development of the UAV.

B. Buoyancy Test for the UAV

To ensure the UAV's flotation device was properly designed, testing was conducted in a controlled environment. The completely loaded system was placed in a controlled body of water to validate the buoyancy calculations as seen in Figure 17. Figure 18 shows the submerged amount of the flotation device was approximately half of the cylinder, 1.5in. This result validates the preliminary analysis which showed that the cylinders needed to have approximately $\frac{2}{3}$ of their volume submerged in water.

C. RTK Accuracy

To increase the localization accuracy of the UAV and ASV, Real-time Kinematic (RTK) positioning was used. A typical RTK system can achieve centimeter–level accuracy, as apposed to a standard GPS, which has roughly 16ft. of radial error. The accuracy associated with an RTK system is a function of time and the number of satellites and base stations in the area. The RTK system for the Ragin' Cajuns RoboBoat consists of one base station that has a fixed location on shore and stations on each of the autonomous systems. A test was done to see what accuracy could be achieved in a 10–minute time frame. As seen in Figure 19, in under 9 minutes, the system's radial error was reduced to 6 feet. The system did not reach its full potential of centimeter level accuracy, but increasing the run time of the RTK system would allow improved accuracy to be achieved. Moreover, the RTK–GPS system was tested throughout this year. On average, the RTK–GPS system's error would decrease between 1.5–2ft after approximately 4–6 hours. During this time, the number of visible satellites was 17–20.

D. System Test

Once the Ragin' Cajuns RoboBoat System was completed, it was taken to a local pond to run some general tests on the system's new com-

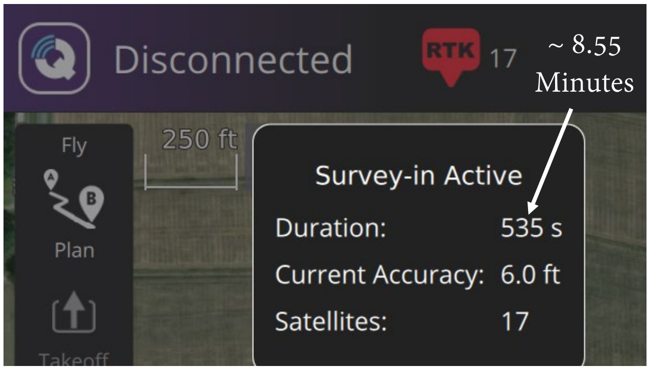

Fig. 19: RTK Accuracy with QGround Control

Fig. 21: Malfunction During Testing

Fig. 20: System Testing

ponents including the new navigation stack, as well as testing the manoeuvrability of the ASV with its new enclosure mounted. In Figure 20, the system is on the water and general testing was being conducted on the operation of the system. Unfortunately, during the test, a thruster mount broke and hit the side of the ASV causing a small divot in the hull, as shown in Figure 21. Plans to fix the hull, mount, and finish tuning the UAV's controller are scheduled before the June 13th deadline for the Optional Operational Video.

V. Conclusions

This paper presented the design of the 2021 Ragin' Cajuns RoboBoat system, including key software, hardware, and strategic improvements. This design builds on the Ragin' Cajuns' 2019 and 2020 entries into the RoboBoat contest and now includes a UAV to aid in data collection, upgraded OAK–D machine vision sensors, and a RTK–GPS system for increased improved accuracy. Contributions made by this team to furthering the Ragin' Cajuns RoboBoat development at the University of Louisiana at Lafayette include a Gazebo simulation of the UAV, an updated CAD model of both systems, and a platform to attempt more tasks in the future, including the Object Delivery task. The upgrades to the sensors and the addition of the UAV by this team will be useful to future Ragin' Cajuns RoboBoat teams.

VI. Acknowledgments

The 2021 team would like to thank Chapman Consulting Inc. for allowing us access to their facilities and resources. Also, the team would like to thank Microsoft Azure and Intel for sponsoring the OpenCV OAK–D AI Competition which allowed us to win 10 free OAK–D machine vision sensors.

References

[1] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system." Springer, July 2014.

[2] M. Labbe, "rtabmap_ros," https://github.com/introlab/rtabmap_ros, 2021.

[3] J. Horner, "multirobot_map_merge," https://github.com/hrnr/m-explore.git, 2021.

[4] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, p. 2280–2292, 06 2014.

[5] V. Ermakov, "mavros," https://github.com/mavlink/mavros, 2021.

[6] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux journal*, vol. 2014, no. 239, p. 2, 2014.

[7] S. of the CRAWLab, "roboboat," https://github.com/CRAWlab/roboboat, 2021.

[8] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[9] T. Takasu and A. Yasuda, "Development of the low-cost rtk-gps receiver with an open source program package rtklib," *International Symposium on GPS/GNSS*, 01 2009.

[10] P.-J. Bristeau, P. Martin, E. Salaün, and N. Petit, "The role of propeller aerodynamics in the model of a quadrotor uav," *2009 European Control Conference, ECC 2009*, 08 2009.

[11] "IEC 60529 Ingress Protection IP Code," International Organization for Standardization, Geneva, CH, Standard, 2001.

[12] L. Lam, "Hottest temperatures ever recorded in all 50 states," 2016. [Online]. Available: https://weather.com/news/climate/news/hottest-temperature-recorded-50-states

[13] M. Schmitt, "The operating temperature for a raspberry p," 2020. [Online]. Available: https://technologisttips.com/raspberry-pi-temperature/

[14] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," 04 2004, pp. 2149 – 2154 vol.3.

[15] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Robot operating system (ros)," *Studies Comp.Intelligence Volume Number:625*, vol. The Complete Reference (Volume 1), no. 978-3-319-26052-5, p. Chapter 23, 2016, iSBN:978-3-319-26052-5.

Appendix

TABLE I: Ragin' Cajuns RoboBoat Specifications

Category	**Item**	**Vendor**	**Specifications**	**Quantity**	**Price ($)**
Battery	4S Li-Po	Turnigy	16V 5200 mAh 450g	4	53.96
Battery	3S Li-Po	Floureon	12V 4500 maH 324.5g	2	33.29
Battery	3S Li-Po	Zeee	12V, 100C 9000 maH 560g	2	71.00
Comm.	TL-WA901ND	TP-Link	2.4-2.4835 GHz 270m range 12V, 1A 5.8W	1	37.99
Computing	Raspberry Pi 4 Model B	ARMv8, 1.5 Ghz 8GB DDR4 RAM	2	90.00	
Computing	Jetson TX2	NVIDIA	256 CUDA Cores 2-Core Denver 2 4-Core Cortex-A57 8GB DDR4 RAM	2	629.99
Control	NAVIO2 HAT	EMLID	5V, 150 mA Cortex-M3 IMU, Barometer	1	168.00
Control	PX4 FMU	Holybro	5V 2 Accel/Gryo Barometer	1	245.00
Enclosure	PJ24208RT	Hammond MFG	0.064 m^3 Fiberglass 11 kg	1	Donated

TABLE II: Ragin' Cajuns RoboBoat Specifications Cont.

Category	**Item**	**Vendor**	**Specifications**	**Quantity**	**Price ($)**
Hull	Fiberglass Cloth	TotalBoat	$6\ \frac{\text{oz}}{\text{yard}^2}$	10.56 yard^2	56.01
Hull	Epoxy	TotalBoat	$1.18\ \frac{\text{g}}{\text{cm}^3}$	4.31 kg	126.99
Hull	Fairing Compound	TotalBoat	$1.32\ \frac{\text{g}}{\text{cm}^3}$	2.27 kg	56.99
Propulsion	T-200	Blue Robotics	$[-4.1, 5.25]$ kgf 76mm Propeller 156g (in water) 390W, 24A (max)	4	169.00
Propulsion	Speed Controllers	Blue Robotics	16.3g 7–26V 30A (max) $[1100, 1900]\ \mu$s	8	25.00
Propulsion	MT 2213 Motor 1045 Propeller	EMAX	935KV, 860g thrust 1045 Propeller	4	67.00
Sensing	GPS-RTK Board	SparkFun	5V, 35mA 5Hz-RTK NEO-M8P-2	3	199.95
Sensing	LiDAR Lite V3 HP	Garmin	1m-40m, 5V 85mA	1	150.00
Vision	UTM-30-LX-EW	Hokuyo	270° FOV 2D Projection 30 meter range 100 Hz	2	4900.00
Vision	Camera Module V2	Raspberry Pi	8 MP Sony IMX219 62.2°, 48.8°	2	25.00
Vision	OAK–D Stereo Camera	Luxonis OpenCV	1MP 1280x800 120 FPS 900mA / 5V 115g 81°, 71.8°	2	Won

TABLE III: Ragin' Cajuns RoboBoat Specifications Cont.

Category	**Item**	**Vendor**	**Specifications**	**Quantity**	**Price ($)**
Vision	OAK–D RGB Camera	Luxonis OpenCV	12MP 4@K30 FPS 900mA, 5V 115g 81°, 68.8°	2	Won
Frame	F450 Quad-rotor	DJI	450mm footprint	1	72.00