- Add Figs 1-3 to webpage, along with discussion of them.
- Be sure to link to flickr albums and GitHub repositories from webpage
- Do we have any video renders from blender to add to webpage?

Ragin' Cajun 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 motivations behind the design choices and improvements made to the University of Louisiana at Lafayette's first entry to RoboNation's RoboBoat Competition that includes an Unmanned Aerial Vehicle (UAV). The overall function of this UAV is to act as a mobile sensor to survey the course~~s~~ with its onboard sensors, which include a Real-time Kinematic (RTK) GPS system System, a Raspberry Pi Camera Module, and an OAK-D machine vision system. The Ragin' Cajun RoboBoat is a catamaran-style autonomous surface vessel (ASV) equipped with four thrusters in an "X"~~-~~Configuration, enabling holonomic motion. The framework that controls the communication between various computers and the implementation of control and mapping algorithms is the Robot Operating System (ROS). The contributions to the ASV from the 2021 Ragin' Cajun RoboBoat team include finishing the upgrades from 2020 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 and 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.

LiDAR for machine vision feedback?

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

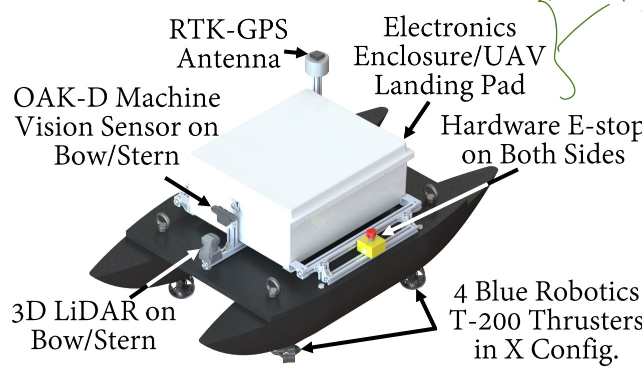


Fig. 1: 2021 Ragin' Cajun ASV CAD Model

The tighter line spacing on these labels looks much better.

Send me these (or point me to them on OwnCloud). I'll add to flickr for you to use on the webpage

For the 2021 RoboBoat competition, the team added an unmanned aerial vehicle (UAV) to be used as a mobile sensor to collect data to assist the ASV in mapping, navigation, and localization. The UAV, shown in Figure 2, is equipped with RTK-GPS, standard GPS, and an IMU for localization, OAK-D stereo camera for mapping, object recognition, and obstacle avoidance, and a Raspberry Pi Camera Module V2 (Pi Cam) for additional image data collection. It is also equipped with a 2D LiDAR lite for altitude augmentation, an RC receiver for manual flight, and flotation gear to ensure buoyancy. The total 2021 Ragin' Cajun 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

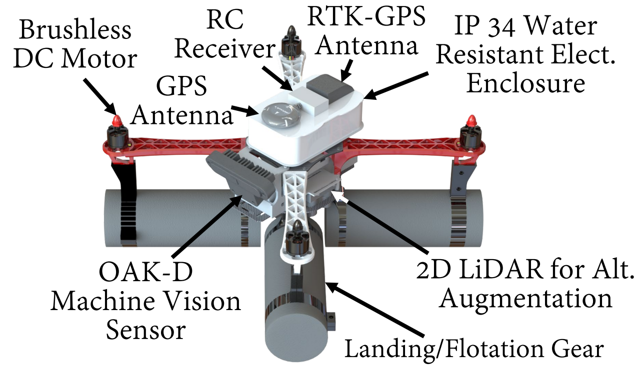


Fig. 2: 2021 Ragin' Cajun UAV CAD Model

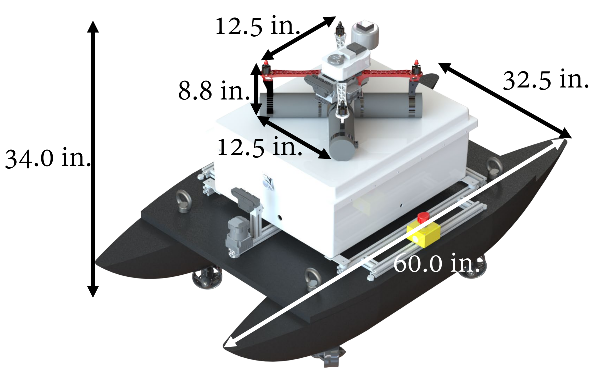


Fig. 3: 2021 Ragin' Cajun Autonomous System

that were gathered are presented. Next, Section V is the conclusion. Finally, in the Appendix, a list of components is presented.

II. Competition Strategy

The past competitions have shown that the Ragin' Cajun 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 obtain an approximate 1,200 additional points by adding a UAV that takes off of and lands on the ASV, as well as competes in the UAV specific task. The addition of this 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 course. This will allow the UAV to be one course ahead of the ASV to collect data and place waypoints for the ASV to navigate towards with increased accuracy due to the RTK-GPS system's correction data. The UAV is also equipped with an OAK-D machine vision stereo camera for object recognition and obstacle avoidance as well as a downward facing Pi Cam for further image data collection.

Should be next course a next section of the course?... or next task on the course?

task?

redundant with intro. How are you using those?

B. *System Software*

A mix of custom and standard ROS packages are used to aid in the control, perception, and mapping of the system. ~~Existing ROS software packages including *robot_localization* and *rtabmap_ros* (Real Time Appearance Based Mapping) are used on both the ASV and the UAV.~~ The *robot_localization* package combines GPS and IMU sensor data, and pose estimates from *rtabmap_ros*, and ArUco markers to compute a single state estimation of the system [1]. The global loop-closure Simultaneous Localization and Mapping (SLAM) package, *rtabmap_ros*, uses ~~allows for~~ the ~~a~~ stereo cameras ~~to create a 3-D point cloud of an environment~~ [2]. ArUco markers allow for the pose of a camera lens to be determined

is used on both the ASV and UAV. It

a little repetitive. Could be more precise.

True. But how does this system use it?

True, but how does this system use it?

just a point cloud? or a map?

on the ASV and UAV to create maps of the environment.

move map range sentence here

from a square marker that is similar to a quick response (QR) square [3].

Move to right after (Aas map part is revised) paragraph. Finish mapping discussion before moving on to ArUco markers.

The maps created by the data collected from the UAV and ASV are compiled using a ROS package called *multirobot_map_merge*, which allows multiple maps from different robots to be compiled into one map [4]. ROS is also used to aid in the overall communication of data for the system by allowing for multiple peripherals to share information across computers connected to the network. The *MAVROS* package is used to communicate with the flight controllers in addition to publishing the data that is being recorded by the flight controllers [5].

still true?

Due to the amount of information that is being collected and processed, multiple computational devices are needed. Furthermore, the imagine processing tasks that are required for this competition ~~is~~ are best handled by ~~the use of~~ dedicated Graphics Processing Units (GPUs). ~~The simpler tasks can be handled by Central Processing Units (CPUs).~~ For this ~~system, this challenge is accomplished by using~~ reason two Jetson TX2s are used on the ASV. These devices are used for image processing and mapping that is needed throughout the competition. The ASV also has one Raspberry Pi 4 Model B that handles basic sensor reading and low-level thruster control ~~for this system~~. The UAV utilizes a Raspberry Pi 4 Model B as the companion computer for its Pixhawk 4 flight controller. ~~It also handles the communication needed for the UAV to share data with the ASV.~~

To speed up the initializing, booting, and configuring process, 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, ~~and necessary libraries~~ 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 for the code that is needed to execute certain tasks to be mounted to the container at time of use. This allows ~~for~~ the code to be saved to a platform like GitHub, so the development can be split across a team.

redundant with packages

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' Cajun RoboBoat is equipped with a forward facing stereoscopic camera that provides images to an image classifier trained by a Convolution Neural Network (CNN). 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 made available~~ passed to a state machine that determines how the ASV should maneuver.

Judge Q
Which one? Why?

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 is sent to the navigation stack to maneuver to the middle of the gate and orient the vessel to be in-line with the channel. 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. As the vessel reaches this waypoint, it completes the Navigation Channel and prepares to launch the UAV.

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 nonlinear 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, restricted maximum velocities, and data collected by the UAV helps the Ragin' Cajun RoboBoat complete

haven't explained this yet

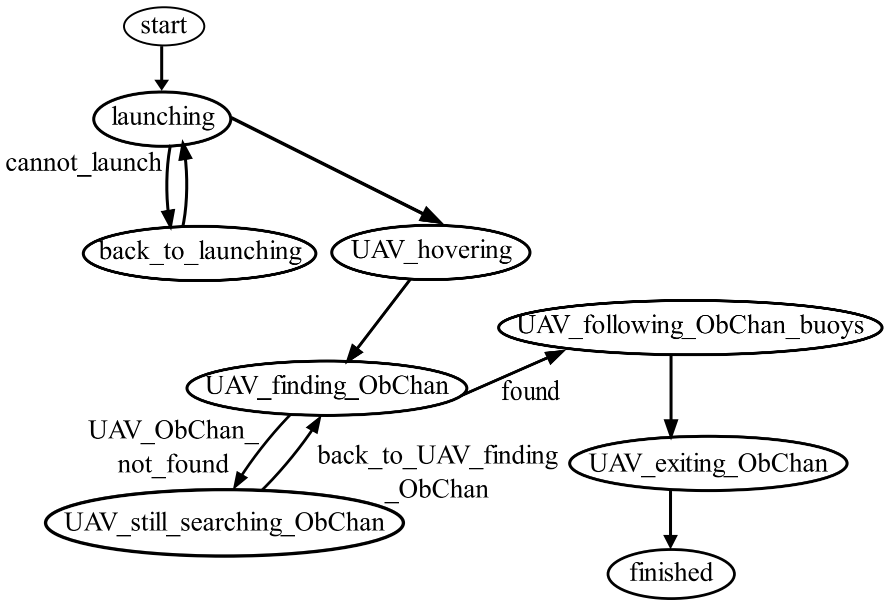


Fig. 4: State Machine for the UAV in the Obstacle Channel Task

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 4. 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 set limit that it can enter one of these redundancy states. This will stop the UAV from getting stuck in a continuous loop. Once the UAV finds the Obstacle Channel, it uses the image classifier 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.

Awkward

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 before it gets to the channel as well as a desired path to follow through the channel. The vision feedback system provided to the navigation stack prevents the ASV from colliding with the green, red, and yellow buoys. ~~Similar to the Navigation Channel, the image classifier looks for green buoys in the right side of the image and red buoys in the left. If red and green buoys in the Obstacle Field are misinterpreted as gates, the path planner will prevent the ASV from entering a region that it cannot fit because it knows the base footprint area of the ASV.~~

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 ~~once it exits as well~~. The UAV uses the GPS coordinate as well as the CNN to determine the location of this course. It will identifies the pill buoy by identifying a buoy that has an orange section on the top and a blue section on the bottom. Then, the UAV proceeds to circumnavigate this buoy while updating the distances between the buoys to find the largest entrance and place a waypoint at that location. ~~This waypoint will be~~ is ~~accessible to the ASV's state machine to motivate it to move towards this opening.~~

task?
the Obstacle Channel task on the course?

Judge Q
Can you do this with just colors? What about your CNN?

How?

When the ASV reaches the Obstacle Field, it moves towards the opening specified by the UAV.

From Section II.B, I thought your maps were combined?

To ensure that it can fit through that opening, the ASV also circumnavigates the Obstacle Field. The ASV saves the location of the entrance and enters. Next, the path planner uses both maps 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 image classifier 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. Then, it 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. The state machine overrides the velocity restrictions placed on the ASV by the path planner, 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 locates the ASV via its location in the course. The UAV will then fly towards this location. 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 5. 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.

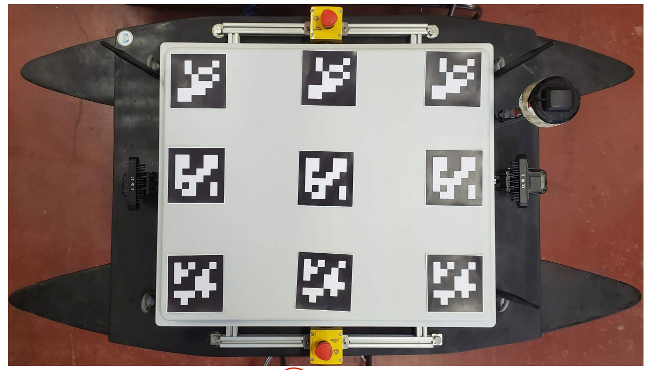


Fig. 5: ArUco Markers On Top of ASV to Aid in UAV Landing

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 by recording all of the obstacles' location and features. Along with the ASV's localization and path planning capabilities, this map is used to guide it back to the starting position without encountering any of the obstacles or exiting the course. Once the ASV reaches the dock, it holds its position with the help from its navigation stack to allow for the UAV to land on top of it. With the Pi Cam's estimation, the increased accuracy of the RTK-GPS system, and the 2D LiDAR for altitude augmentation, the UAV will slowly approach the lid of the enclosure and land here.

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' Cajun 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.

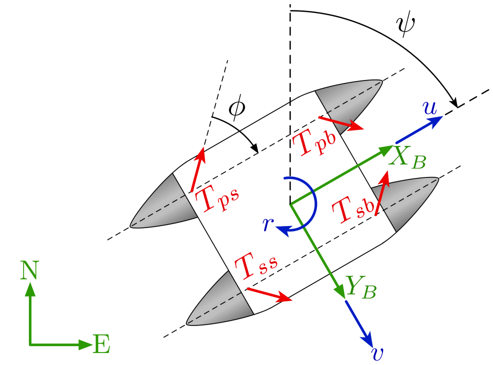


Fig. 6: Free-Body Diagram of RoboBoat Thruster Configuration

The 2021 Ragin' Cajun RoboBoat team declined to develop the required mechanism to complete this task; however, the frame selection and custom frame plates for the UAV were designed with additional mounting holes to allow for further development. This will allow future teams to develop subsystems as needed.

III. Design Creativity

A. Thrust Configuration

The thrusters are configured in an "X" pattern, and mounted at 45° angles, relative to the bow. A free-body diagram of this thruster configuration is shown in Figure 6. This configuration enables holonomic motion, or motion in all three degrees of freedom independent of each other. This improves the maneuverability of the ASV through tasks like the Obstacle Channel and Obstacle Field. ~~It also assists in docking since the ASV can move in the positive or negative sway directions and does not have to do a forward-reverse maneuver, such as a car parallel parking~~.

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, a~~llows for~~ improves the precision of the ASV ~~to increase its~~ localization c~~apabilities~~ [7]. This specific thruster configuration allows the ASV to take full advantage of the increased localization precision.

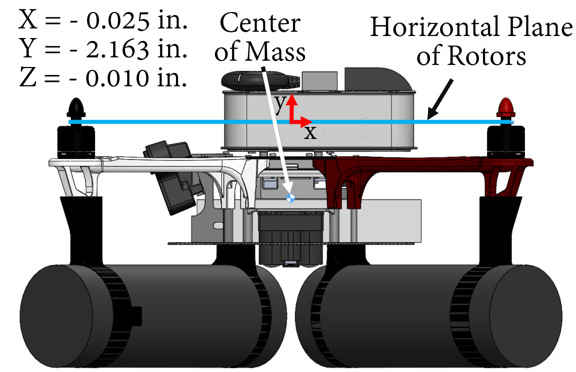


Fig. 7: UAV Center of Mass

B. UAV Center of Mass

~~To achieve desired stability during flight, the UAV's center of mass (COM) had to be considered.~~ 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 the stability of forward flight and the UAV's ability to combat wind disturbances [8]. For this application, a COM below the horizontal plane in which the rotors lie is desirable, because it increases forward flight stability [8]. 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 7.

C. UAV Enclosures

The RoboBoat competition takes place regardless of weather conditions. Therefore, it is imperative to keep all electronics safe from rain 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 [9]. To ensure that the main electronics would not overheat or be introduced to CPU throttling, a heat transfer analysis was completed. The problem is modeled as a thermal resistance problem, as seen in Figure 8. The hottest temperature that

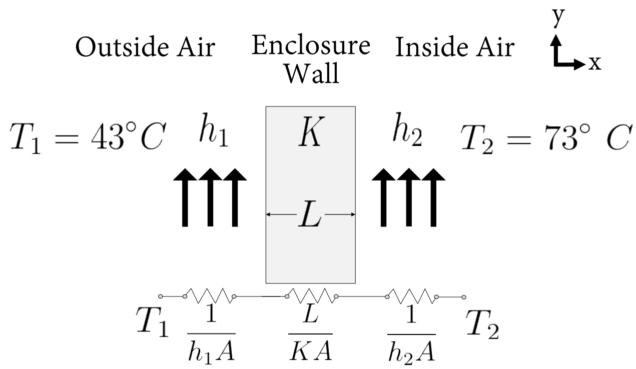
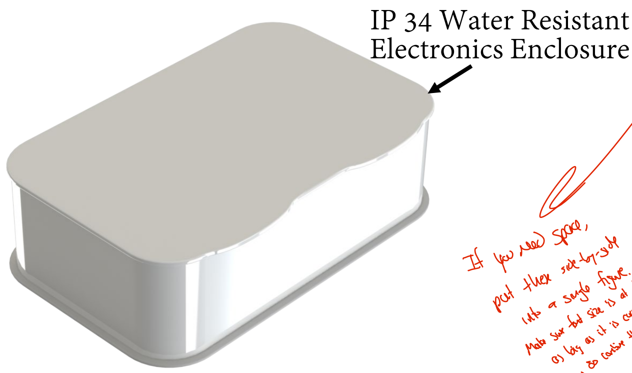


Fig. 8: Model of Simple Heat Transfer Problem



If you need space, put these side-by-side into a single figure. Make sure font size is at least as big as it is currently, if you do combine them.

Fig. 9: Main UAV Electronics Enclosure

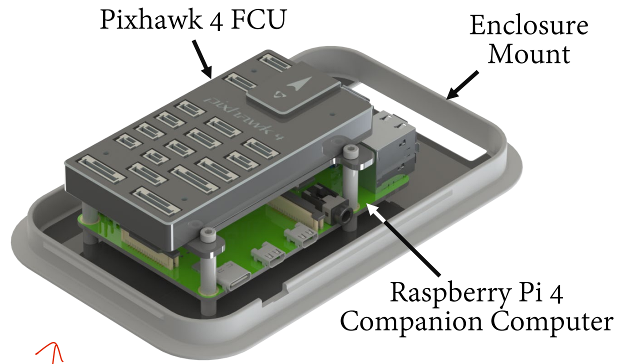


Fig. 10: Main UAV Electronics Mount

need a bit more space

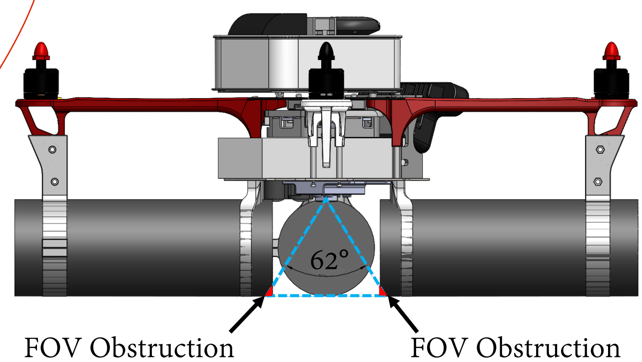


Fig. 11: Raspberry Pi Camera Module V2 FOV

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 [10], [11]. 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.

Move before thermal analysis, so reader can picture the enclosure during the analysis.

The main electronics enclosure for the UAV, shown in Figure 9, 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.

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 11, a sketch showing the analysis of the horizontal (largest) field of view (FOV), 62.2 degrees, of the PiCam can be seen. ~~As seen in Figure 11,~~ A ~~a~~ small amount of the polyethylene foam is ~~was expected to be~~ in the camera's FOV. However, this amount of interference was later determined to be insignificant.

2) OAK-D FOV: As seen in Figure 12, 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. ~~During the design process, the placement of all image sensors used on the UAV needed to be closely analyzed to insure that all necessary image data needed for object recognition, obstacle avoidance, and landing procedures is collected.~~

Doesn't add anything

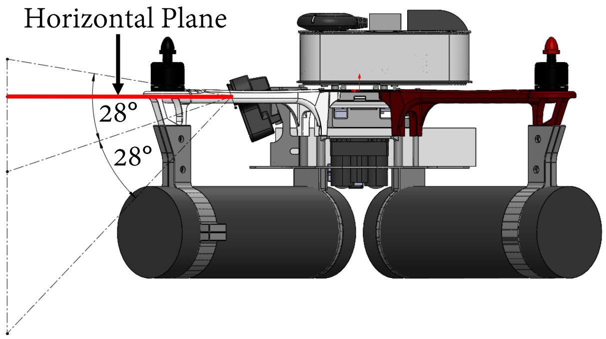


Fig. 12: OAK-D Field of View on UAV

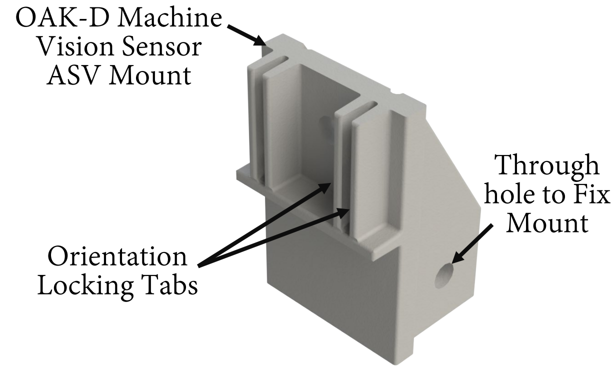


Fig. 14: OAK-D Camera Mount for ASV

If you need space, this is a candidate to cut too.

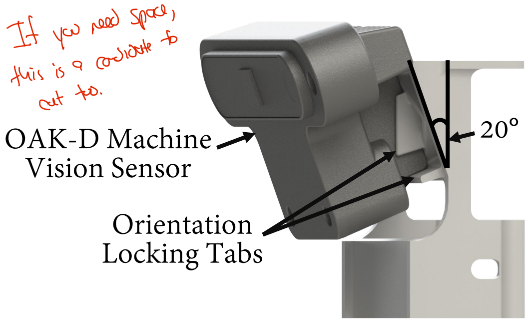
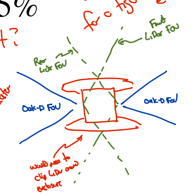


Fig. 13: OAK-D Camera Mount for UAV

~~The image sensor's FOV, mounting, and data type governed the orientation and location of these sensors. Moreover, these factors drove the design of their mounting hardware~~. As seen in Figure 12, the vertical FOV of the OAK-D stereo camera is 56 degrees. It was determined that the forward facing camera mount which holds the OAK-D on the UAV should be designed with a 20 degree angle from the vertical, as seen in Figure 13. This angle gives the OAK-D 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. ¶ The ASV's stereoscopic sensors were upgraded from the previous ZED Stereo Camera, to keep the implementation of code across the system uniform. The OAK-D machine vision sensor was added to both bow and stern on the ASV. 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, as seen in Figure 14. ~~This is imperative when performing SLAM operations.~~

3) Planar LiDARs: The ASV is also equipped with two Hokuyo planar LiDARs for ~~increased~~ obstacle avoidance ~~capabilitie~~s and for ~~assistance in creating a~~ mapping of ~~its~~ the environment. Theses LiDARs are positioned directly ~~in the middle of the ASV~~ on both the bow and stern. They give the ASV a 270° FOV of its surroundings, thus increasing its ability to utilize its holonomic motion. These LiDARS also give the ASV an approximate 68.15% increase in horizontal perception. increased from what?

270° at front and 270° at rear, right? Do you have room for a figure?



IV. Experimental Results

A. System Simulation

a little awkward. Not really imported. "A model of the ASV was developed for use in the rotors_simulator ROS package..."

The UAV was imported into a simulator called RotorS because of its use of a physics-based ROS application called Gazebo [12]. This allows for the UAV to experience real world forces in a simulated environment. ~~Moreover, noise can be introduced into the simulation to more accurately model a real-world environment~~. 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. ~~The base environment was used.~~

In this simulated environment, all of the sensors were configured to accurately represent their specifications. In RotorS, flight paths can be set, scripts that control the perception and mapping

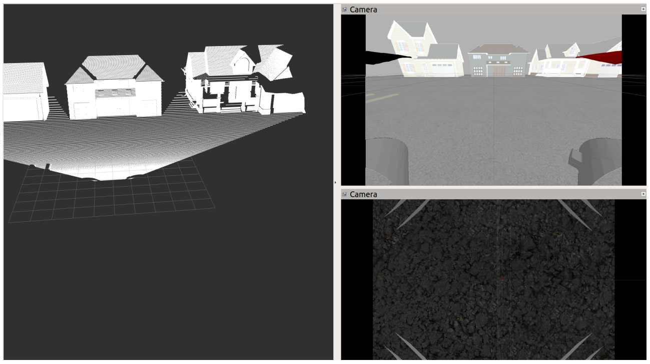


Fig. 15: UAV Peripheral Simulation in RViz

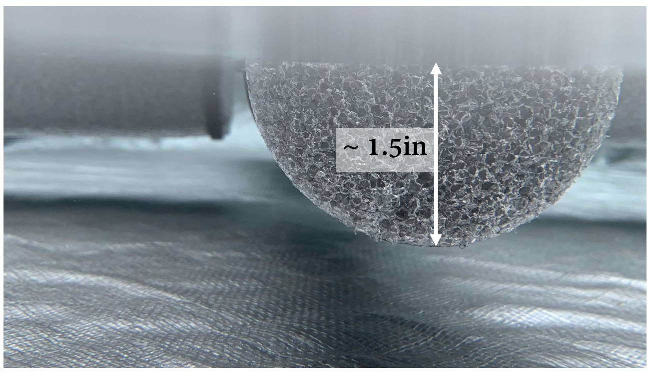


Fig. 16: Submerged Flotation/Landing Gear

algorithms can be executed, and data can be collected. For example, Figure 15 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 15 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 ~~analyze if it would float.~~ Figure 16 shows the submerged amount of the flotation device was approximately half of the cylinder, 1.5in. This result exceeds the preliminary analysis which showed that the cylinders needed to have approximately $\frac{2}{3}$ of their volume submerged in water.

Validate the buoyancy calculation

Fig. 17: Approximate IP34 Standard Test Results

If you need space, cut this figure

C. IP Rating Test

Since the electronics enclosures on the UAV were designed to approximate IP34 standards, a simulated test was conducted to ensure ingress protection from splashes of water in all directions. Typically, devices are tested with controlled machines to spray the enclosures with a precise streams and steady angles. However, the test was conducted by first adding a piece of a hydrophilic material within the enclosures to ensure any water that could leak in would visible after testing. Then, a silicone bead was placed around the lip of the encloses and around any exposed wires to seal the openings. Next, a 12oz. water bottle was modified to allow for a stream of water to be sprayed from it. Finally, the enclosures were placed on a pedestal and sprayed from all directions ensuring water touched every surface. The results for the main electronics enclosure are displayed in Figure 17. With the test results showing no water ingress, the design approximations of IP34 standards were satisfied.

Be more concise here. I don't think this is worth the space you are currently giving it.

D. RTK Accuracy

To increase the 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' Cajun RoboBoat consists of one base station

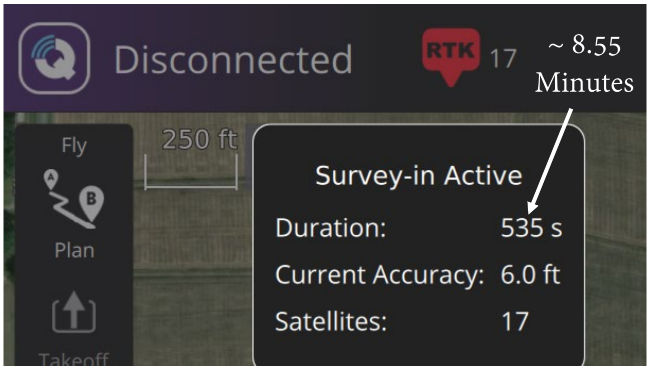


Fig. 18: RTK Accuracy with QGround Control


Fig. 19: System Testing

that has a fixed location on-shore and stations on each of the autonomous systems. A test was done to see how accurate the base station could get in a 10-minute time frame. As seen in Figure 18, 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.

Did you only do this once? If not do you have statistical analysis of what kind of precision you could expect?

include ~ between Figure~\ref{} to prevent latex from separating number from figure text

E. System Test

Once the Ragin' Cajun RoboBoat System was completed, it was taken to a local pond to run some general tests on the system's new components including the new navigation stack, as well as testing the manoeuvrability of the ASV with its new enclosure mounted. In Figure 19, the system is on the water and testing was being conducted on the operation of the system. Unfortunately, a thruster mount broke and hit the side of the ASV causing a small divot in the hull, as shown in Figure 20. 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 ~~is due~~.


Fig. 20: Malfunction During Testing

V. Conclusions

This paper presented the design of the 2021 Ragin' Cajun RoboBoat system, including key software, hardware, and strategic improvements. This design builds on the Ragin' Cajuns' 2019 and 2020 entries into the RoboBoat Challenge and now includes a UAV to aid in data collection, upgraded OAK-D machine vision sensors, and a RTK-GPS system for increased localization accuracy. Contributions made by this team to furthering the Ragin' Cajun 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 competition strategies employed and documented, upgrades to sensors, and the addition of the UAV by this team will also be useful to future Ragin' Cajun RoboBoat teams.

awkward

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] 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.
[4] J. Horner, "multirobot_map_merge," https://github.com/hrnr/m-explore.git, 2021.
[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] 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.
[8] 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.
[9] "IEC 60529 Ingress Protection IP Code," International Organization for Standardization, Geneva, CH, Standard, 2001.
[10] L. Lam, "Hottest temperatures ever recorded in all 50 states," 2016. [Online]. Available: https://weather.com/news/climate/news/hottest-temperature-recorded-50-states
[11] M. Schmitt, "The operating temperature for a raspberry p," 2020. [Online]. Available: https://technologisttips.com/raspberry-pi-temperature/
[12] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," 04 2004, pp. 2149 – 2154 vol.3.

APPENDIX

TABLE I: Ragin' Cajun 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	Pi 4B+	Raspberry Pi	ARMv8, 1.5 Ghz 8GB DDR4 RAM	1	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' Cajun 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' Cajun 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