Design and Development of the NaviGator Autonomous Maritime System

Alex Perez, Alex Kuzmicki, Andres Castrillon, Cameron Brown, Mehron Talebi, Jarrod Sanders, Andrés Pulido, Andrew Knee, Aditya Ramesh, Kayleigh Beron, Luka Bjellos, Dr. Eric Schwartz, Dr. Carl Crane

***Abstract*—The NaviGator Autonomous Surface Vehicle (ASV) is part of a larger group of collaborative autonomous aerial, surface, and subsurface vehicles known as the NaviGator Autonomous Maritime System (AMS). The NaviGator AMS has been designed to compete in the Association for Unmanned Vehicle Systems International (AUVSI) Foundation's 2022 Maritime RobotX Challenge in Sydney, Australia. This paper describes design approaches taken to complete challenges presented in the 2022 Maritime RobotX Challenge, as well as the electrical, mechanical, and software infrastructure upon which the NaviGator AMS has been designed.**

I. Introduction

The University of Florida's (UF) Team NaviGator AMS is a multidisciplinary group composed of undergraduate and graduate students from the departments of Electrical and Computer Engineering and Mechanical Engineering. This project is primarily sponsored by the Machine Intelligence Lab (MIL), which has 24 years of experience in competing in the AUVSI Foundation's robotics competitions, including past championships in the RoboSub and RoboBoat competition, as well as in the RobotX Maritime Challenge. Due to the larger scale of the Maritime RobotX Challenge, MIL has partnered with the Center for Intelligent Machines and Robotics (CIMAR), a lab that has competed in three DARPA challenges and has extensive experience with developing highly intelligent large-scale autonomous vehicles. Between MIL's experience in autonomous maritime systems design and CIMAR's experience in autonomous ground vehicles, Team NaviGator AMS feels that they have created a winning combination and look forward to competing in the 2022 Maritime RobotX Challenge.

II. Design Strategy

Development of NaviGator AMS began prior to the 2016 Maritime RobotX Challenge in Honolulu, Hawaii. This initial version of NaviGator included sensors and actuators that were necessary to accomplish all the tasks for that competition. These systems were updated in preparation for the 2018 Maritime RobotX Challenge, also in Honolulu, Hawaii. In the past four years, we have expanded on the opportunity to add more features and capability to NaviGator, while ensuring that we balance the restoration of previously working functional designs. The impact of COVID-19 prevented us from physically working intensely on the boat throughout a significant proportion of 2020 and 2021, giving us a further challenge in the development of this year's submission.

III. Vehicle Design

This section of the paper will describe the hardware and software that was developed for this competition, as well as the motivations behind these choices. This will include descriptions of early iterations of hardware and software that may have failed, what was learned in that process, and how that knowledge was integrated to improve on the designs.

A. Mechanical Systems

The mechanical platform used for the NaviGator ASV is a modified WAM-V research vessel developed by Marine Advanced Research. Significant developments were made to the mechanical design of NaviGator AMS, including the development of a landing pad and drone support system to ensure reliability for the drone aspect of this year's challenge. Due to the novelty of this aspect of the challenge, significant time was invested to ensure proper development of a drone support system. To develop systems, the mechanical team used SolidWorks for part design and GrabCAD for file management. Several of the mechanical modifications that the team has made will be detailed in this section.

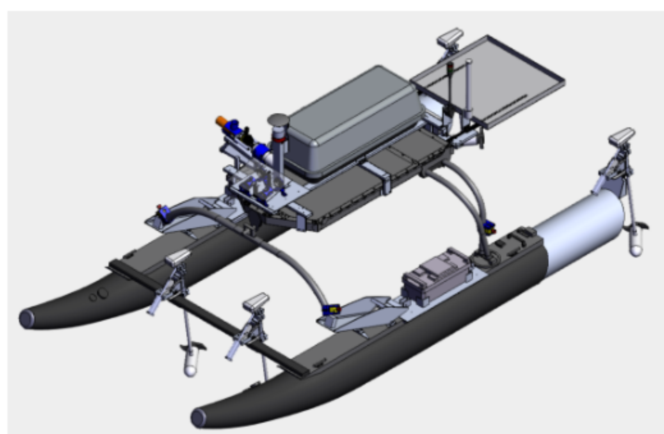

Fig. 1. 3D model of the NaviGator assembly created on SolidWorks.

1) Propulsion: NaviGator ASV's propulsion system began as two forward-facing stern thrusters, providing the ASV with a skid-steer configuration. After a short time of testing, it became apparent that adding more thrusters and mounting them at an angle would simplify the vectoring of the thrust to achieve a desired motion, as well as adding the capability of lateral motion. The current configuration features two bow and two stern thrusters oriented at a fixed 45 degrees. This is a thruster configuration that the team used in the 2013 RoboBoat Competition with much success, earning first place. In addition to improved maneuverability, using four thrusters provides redundancy in the system, allowing the ASV to still have maneuverability even if either both bow thrusters or both stern thrusters fail. This feature was invaluable when a motor driver died minutes before a qualification run in the 2013 RoboBoat Competition. With a quick modification to the thruster mapper program, the ASV was able to operate with just three thrusters, saving the run. The major disadvantage of this configuration is that the fixed angles of the thrusters means that it is not particularly efficient moving in any direction. However, for the tasks that the Navigator ASV is designed to perform, maneuverability is significantly more important than efficiency.

Mounting the thrusters posed many challenges and required several design iterations, especially for the bow thrusters. For the ASV to be deployed from a trailer, the bow thrusters had to be either removed or raised during deployment so they would not collide with the trailer structure. The transom clamps on the trolling motors accommodated this function. 3D printed polycarbonate clamping blocks that interfaced with the clamps on the trolling motors kept them fixed in place. While the mounts held the motors securely, the 3D printed parts began to crack and eventually failed. To solve this issue, the clamping blocks were machined from aluminum.

2) Sensor Mast: The need for a stable sensor platform is paramount in machine vision applications. The preliminary design utilized an 80/20 aluminum rail truss, which did not provide the required stiffness and resulted in smearing of the vessel's detection data. The initial sensor platform also did not raise the LIDAR system high enough to permit detection of obstacles in immediate proximity to the pontoons, a problem rectified in the final design. This year, a new 3D printed mount for the LIDAR scanner was made to tilt it's inclination 15 degrees downward. This off-set ensures the scanner directs it's beams towards the objects in front of it rather than trying to detect objects in the sky. This maximizes the number of useful point-cloud data points generated by the scanner [1].

As previously mentioned, the cameras, LIDAR, and GPS antenna require a rigid support. The need for an unobstructed GPS antenna guided the design towards a mast structure. For transport to the competition site, the assembly had to fit within the prescribed envelope of a Pelican Products transport case, requiring a modular assembly process. These target specifications led to a base-and-tree assembly, where the mast is simply welded to a plate that then fastens to the payload tray via a superstructure. For corrosion resistance and manufacturability, 6063 aluminum was chosen. To simplify the assembly process, fastener types were standardized. The mast is centered laterally on the ASV, which helps create a well-defined coordinate system that permits simpler software transformations.

3) Electronics Enclosure: NaviGator ASV's electronics are housed in a Thule Sidekick cargo box. The team originally considered commercial waterproof boxes, but began looking for other options due to their high costs. One student suggested the idea of using a cargo box after being inspired by family road trips they had taken when they were younger. While traditionally used to mount on the top of cars to provide additional storage, the cargo box was an ideal electronics enclosure due to its watertight integrity, aerodynamic form factor, low cost, and a side-opening mechanism that makes it very easy to access all the electronic components.

The box's watertight integrity prevented the team from using air circulation for cooling. Instead, a combination of techniques are used to cool the box. First, an adhesive reflective covering was applied to the lid of the box to reflect heat generated by solar radiation. Second, the box has an active water cooling system that is used to remove the heat generated from the electronic components inside the box.

Fiberglass inserts were used to mount the components inside the box. These inserts add rigidity to the relatively flimsy box and make it easy to add or remove components from the box. The components that need to be frequently removed, e.g., the hard drives, are attached to the fiberglass with Velcro. The rest of the components are attached with traditional fasteners.

4) Racquetball Launcher: A system for delivering the racquetballs into the target for the Detect and Deliver task was developed by attempting to reduce the complexity of the previous year's versions of the launcher. Our two previous designs utilized a counter-rotating flywheel launcher and a pneumatic linear actuator respectively. For this iteration we attempted to reduce control complexity and avoid inconsistency in the mechanical systems by implementing a "slingshot" launcher. Using stretched surgical tubing as our launching force, triggered by servos that pull a firing pin for each ball we can achieve similar launch ballistics to our previous versions with less points of failure and variability.

5) UAV Landing Pad: A landing pad was mounted on the back of the boat to create a landing area for the drone used in the "Search and Report" task. The placement of the landing pad was selected so that it would stand clear from other objects, such as antennas and sensors, thus, giving the drone an unobstructed space to land. The landing pad consists of an 80-20 aluminum structure and a flat panel where the drone lands. Originally, the panel was planned be made out of sheet metal, but after the electrical team raised concerns that the large metal panel could interfere with the antennas on NaviGator, the material for the landing pad was changed to acrylic. The strength of the landing pad was verified by performing FEA simulations in SolidWorks that mimicked the forces the assembly would experience when the drone lands on it, as shown in Fig. 2.

B. Electrical Systems

Robustness and simplicity were the primary motivating factors behind the design of the NaviGator ASV's electrical system. The team focused on these aspects in order to get a testable system built quickly and minimize any downtime due to electrical failure.

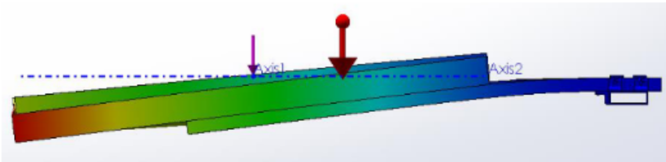

Fig. 2. FEA Analysis of the platform done in SolidWorks. Force applied: 20lb. Deformation scale: 22.3657.

1) Power Delivery: The salient features of the power system are the dual battery power supply and the power merge board. The NaviGator ASV's power requirements surpassed those of MIL's other projects in both the amount of power and required power source durability. Solving these challenges required looking outside the typical battery suppliers MIL used in the past. To be able to supply adequate power to four thrusters, computers, sensors, and communication hardware, the ASV uses two Torqueedo Power 26-104 batteries. Each battery powers two of the thrusters and contributes to the power rail that powers all other devices on the NaviGator ASV.

The power merge board is a student-designed printed circuit board assembly (PCBA). It uses two Texas Instruments LM5050 High Side OR-ing FET controllers as ideal diode rectifiers to balance and parallel the two batteries into one rail that supplies four output ports. This makes the system more fault-tolerant to a failing battery, a feature used in normal operation to switch batteries out without turning the system off. One of the strengths of MIL is the ability to design hardware and software that can be reused on other projects and vehicles. This is the third vehicle for which this board design has been utilized. The design was originally created for PropaGator 1 and then used on PropaGator 2, both of which have competed in the RoboBoat Competition.

2) Passive Sonar: The passive sonar is a student-designed PCBA that implements a signal conditioner with for four Reson hydrophones. The timing information from the hydrophones is put in a buffer and sent to the computer for processing. Built for resiliency and long-term reliability, our hydrophone PCBA design has functioned in over six autonomous vehicles originating from MIL. The board provides conditioned data from the hydrophones to software systems over a serial connection for further processing.

3) Kill System: The hardware kill system consists of two student-designed PCBAs and four off-the-shelf twist to detent kill switches. The kill system for the vehicle also has a software component. The kill board monitors the status of six kill sources. When any of the six sources request a kill, the kill board cuts power to the thruster motor controllers. The six kill sources are the four off-the-shelf switches that are mounted around the vehicle, a remote kill switch, and the computer. The remote kill switch operates over a 915 MHz LoRa link and displays the hardware kill status of the vehicle. The kill board is also used to control the NaviGator ASV's indicator lights.

C. Software Systems

1) Object Detection: The lowest level perception service available on the NaviGator ASV is the Occupancy Grid Server. Occupancy grids are a two-dimensional grid-like representation of the environment generated by the sensor suite available on the ASV. The generated map contains both the occupied and unoccupied regions in the environment. This information is provided to the server via any range-detecting sensor onboard. On the ASV, the primary range-detecting sensor is a Velodyne VLP-16 LIDAR. A LIDAR uses lasers to provide relatively dense range information of the environment. This information is then segmented by regions containing dense clusters of relatively close points. These bounding regions are treated as obstacles, and are placed in the occupancy grid. This information is then provided to higher level services such as the motion planner and classification server.

2) Classification: In the classification system, the role is to attempt to label the clusters gathered from the object detection system. Our classification system contains a YOLOv7 model, which is one of the newest and fastest real-time object detection systems in the field computer vision. The YOLO model was trained on thousands of labeled images of buoy data from field testing, as well as, previous competition videos [2]. The model is constantly receiving a stream of images from our cameras, and it places bounding boxes on the image on where it believes the buoys to be. This can be seen in Figure 3.

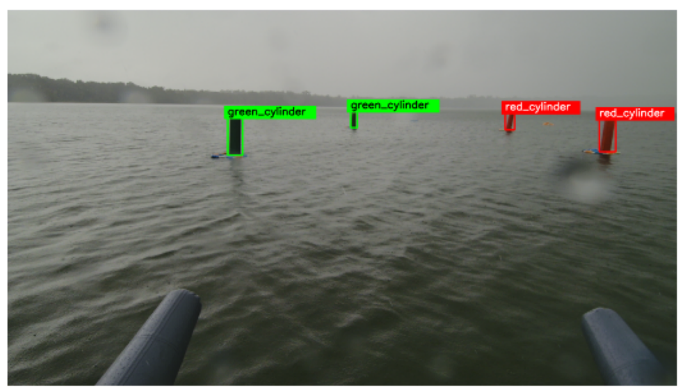

Fig. 3. Due to the volume and variety of images used to train the model, it is able to detect the buoys even in more complex scenarios (such as a rainy and cloudy day).

From here, provided the 3D bounding boxes from the LIDAR system, the classification system transforms the position of those bounding boxes into camera space, which gives us an estimate of where the object would appear in an image. Combining this with the YOLO bounding boxes, we can assign each YOLO bounding box with its closest camera space transformed cluster and label each cluster/buoy accordingly.

3) Motion Planning: For motion planning, we currently use a rapidly-exploring random tree (RRT) algorithm. The algorithm starts with a seed node at the ASV's initial state. It then randomly samples a state in the region of navigational interest. A nearness function is applied to every node currently in the tree, and then that node is extended or steered towards the random state following a policy function. The endpoint of that extension is added as a new node to the tree only if it is allowable, and the algorithm repeats. If an extension, or any intermediate state leading up to it, is not allowable, that iteration is simply abandoned. Once a node reaches the goal region, the tree is efficiently climbed from the goal back to the seed, and is classified as one solution to the planning problem. The best of the found solutions is defined as the one that takes the least amount of time. The goal region is likely to be reached because one can bias tree-growth towards it by shaping the probability density function from which random states are sampled. An example of the RRT algorithm in action can be seen in Fig. 4 [3].

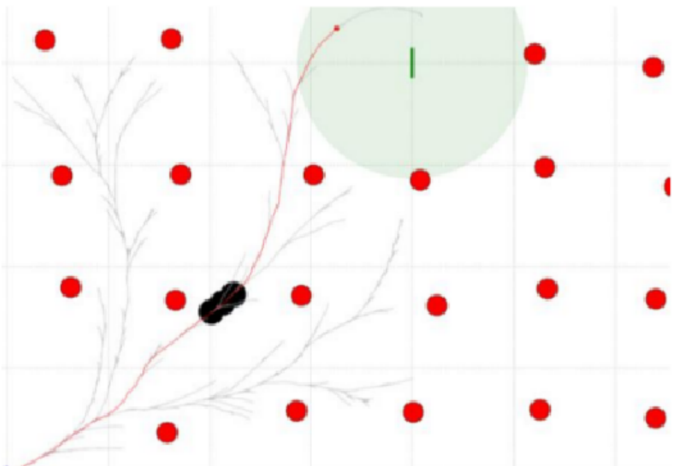

Fig. 4. An example of the NaviGator ASV's RRT planning towards a goal region

After selecting the RRT algorithm for safety and flexibility, the final step was to integrate the algorithm with a real-time system. One of the biggest difficulties in doing this was dealing with a highly nonstatic environment. Obstacles spontaneously appear when they get in range of the perception system. This means that a valid path can suddenly become invalid with only seconds to spare. To make efficient use of time, the planner should always be planning the next move so that the RRT has more time to get a better solution. To handle this, the planner had to be made asynchronously interruptible, and a lot of plan-reevaluation and crisis-aversion logic had to be built in to elegantly deal with spontaneously appearing and/or moving obstacles that cross the ASV's current path [4]. The ASV's real-time ROS-integrated RRT algorithm being run for an arbitrarily drawn, complicated occupancy grid can be seen in Fig. 5. Tree nodes can be seen in blue. The ASV was only given one second to plan its first move. It used its time during the first move to plan its second move, shown in red. While the paths generated using this method are safe and useful for solving the problem of navigation in the competition, the team is actively working on improved heuristics for smoothing out the paths [5].

4) Motion Control: Since the RRT motion planner uses a model of the ASV, in principle it would be possible to employ a model-predictive control architecture in which the ASV rapidly re-plans from its current state to steer it back onto the desired path. However, due to the randomness inherent to the RRT itself, such a method did not work well in practice. Thus, the team opted to make use of the sequence of states generated by the motion planner rather than the inputs to define the reference a feedback controller tracks. First, a simple manually-tuned full-state feedback PD controller was used. Tracking along straight paths was nearly perfect with this alone, providing a positional steady-state error of less than 0.25 meters. However, along curves, a larger positional steady-state error of a few meters would always emerge depending on the curvature. Even the introduction of a standard integral term did not fix this problem. The team figured that this was because an integral of the world-frame error alone would only be able to compensate for disturbances that are constant in the world-frame. Simulation revealed that the sources of the curved motion disturbances were centripetal-Coriolis effects and heading-dependent drag forces. A more intelligent integrator would be necessary to compensate for these state-dependent disturbances. Most marine and aerial systems accomplish this by using a model-reference adaptive control (MRAC) architecture. MRAC works very well on the ASV, bringing steady-state error to negligible amounts in all cases without introducing oscillations. Additionally, it does not wind-up as much as an ordinary integrator when unexpected disturbances are applied, such as humans pushing the ASV, since it is trying to adapt specifically to drag and inertial effects instead of constant external forces. Finally, with the controller outputting desired wrenches (i.e., forces and moments), the last operation needed is to map that wrench to a thrust command for each thruster. A surface vehicle would only need three thrusters to be holonomic, but with four, the ASV is more fault tolerant. This redundancy in the mapping can be solved as a regularized least-squares problem by evaluating a pseudoinverse.

Fig. 5. The NaviGator ASV's real-time ROS-integrated RRT algorithm being run on an arbitrarily drawn, complicated occupancy grid

5) Navigation and Odometry: The NaviGator ASV uses a global positioning system (GPS) and inertial navigation system

(INS) that is developed by Forrest Voight, a UF graduate and member of 2016 Team NaviGator AMS. It primarily consists of a circuit board with a Spartan-6 field programmable gate array (FPGA), radio frequency (RF) frontend, inertial measurement unit (IMU), magnetometer, and a barometer. The FPGA performs the correlation operations that enable tracking of GPS satellites. All the sensor measurements and correlations are passed to a computer via USB, into a pipeline of software modules that track and decode the signals from the GPS satellites and then fuse measurements using an extended Kalman filter into an estimate of the ASV's pose in both absolute world and relative odometry coordinate frames. Last, the resulting odometry is transformed so that it describes the ASV's coordinate frame and it is then passed to ROS. By using the sensors to aid the GPS solution and taking advantage of GPS carrier phase measurements, extremely precise relative odometry is possible, with noise on the order of centimeters over periods of seconds to minutes. This is the result of years of work, during which several iterations of the hardware were produced (and deployed on other MIL robots.)

IV. Strategy Design

A. Situational Awareness and Reporting

We establish a connection to the technical director server through a TCP/IP socket. A heartbeat node runs in the background and aggregates multiple information sources into the format required by competition; this string is then sent over the socket.

B. Entrance and Exit Gates

In order to detect where the acoustic beacon is, we are currently using a two hydrophone system to calculate the direction in which the ping is coming from. With a two hydrophone system, we can hear two instances in time when a sound is received. The delta produced by this difference gives us infinitely many solutions of the direction the sound is coming from. However, given the constraint that it is known where the boat will be relative to the entrance gate, and the fact that it is known that there 3 possible locations of where the beacon can be (in between one of the three gates), the hydrophone system associates certain ranges of deltas to certain gates. For example, using the point cloud system, the mission can position the boat directly in front of the middle gate. After this, the range the delta resides in after listening to the beacon will correspond to the gate that we need to go through. Given the clusters of where the buoys are the boat just needs to go through the middle of the chosen pair of clusters.

Once the boat goes through the entrance gate, for the preliminary round, it uses several way points to circumnavigate the black buoy, and the initial entrance gate way point is reused for the exit gate. For the later rounds, the entrance gate way point will be used later once exiting the obstacle course.

C. Follow the Path

For the Follow the Path challenge, the is constantly attempting to look at LIDAR clusters and associate these with the bounding boxes from the YOLOv7 model (as mentioned in the Classification section. Given this, the higher level mission then begins an algorithm to flow through the gates of buoys as smooth as possible. At a high level, the boat tasks the perception module if there are two white buoys present, if there are, the boat goes through the closest two white buoys. If there are not, the boat investigates the closest buoys until it finds two white buoys. If there are more than two, it goes through the closest two white buoys. This process loops on for red-green buoy pairs until it reaches another pair of white buoys.

D. Wildlife Encounter

For the Wildlife Encounter task, the drone lifts off the boat. There are several buoys that are candidates for the Wildlife Encounter investigation and are labeled as ”UNKNOWN”. The LIDAR system allows the vehicle to calculate the global position of the unknown buoy, which is sent to the drone. Once the drone arrives to the location, it notifies the boat, and the boat sends the signal to take a picture using the hyper-spectral imaging camera. The drone's computer does onboard image processing which allows it to compare the signature captured to signatures corresponding to each of the type of buoys. This process is repeated for the other two Wildlife Encounter buoy candidates. When the boat knows what each buoy is, it begins traversing the course by defining way points around the buoys it needs to circumnavigate (in the proper direction), and when traveling from buoy to buoy, it makes sure to avoid going within a certain radius of the buoy that needs to be avoided.

E. Scan the Code

For the Scan the Code task, the strategy is to approach the the buoy and use a combination of LIDAR and image to determine the light colors. The 3D bounding box associated with the scan the code LED buoy is converted to camera space, and a mask is created around the LED portion of the buoy. A YOLOv7 model was generated for detecting the buoy LED colors, so each masked image is passed through the model to generate a classification of the color appearing to the boat. The sequence of colors is then recorded and sent back to shore.

F. Detect and Dock

In the Detect and Dock challenge, our strategy behind it revolves around using the LIDAR system as the primary method to identify the dock and go into any of the docking bays. The first challenge with docking involves moving the boat to the correct side of the dock in order to view the color panels. Finding the correct side uses the occupancy grid and point cloud clusters generated by our LIDAR processing system. Using the occupancy grid, the ASV finds the correct side by using both a center of mass calculation and occupancy grid emptiness finder. First, the center of mass for the occupancy grid of the dock is calculated and compared to the centers of the four sides. Since we know the docking bays face the longest side of the dock, two center points can be ignored. From here, we see which of the longest sides the center of mass is closest to. The side closest to the center of mass is opposite of where we need to be to observe the tiles, so we move to the opposite side. Here, we use a second check to verify the side determined by the center of mass calculation. From the center of the bounding box of the dock, we draw a line to the midpoints of the two longest sides. In this line, we calculate how many occupied spaces are encountered and the line with the least occupied spaces is the correct docking side. If this calculation disagrees with the conclusion drawn by the center of mass calculation, then we trust this second calculation as once it is in line with one of the sides of the dock, it is generally a more accurate solution. Once we're at the midpoint of the correct side, we use the LIDAR to find the back plates of the dock so we can both segment images from our cameras to identify the colors, and so we can calculate a point to move to for docking. Finding the back plates of the dock involves removing the LIDAR points for the bottom of the dock and using K-means clustering to put the LIDAR points into three distinct clusters. With this cluster, we find the average point which then is transformed into boat-space and moved 5 meters in parallel towards the longest side of the dock and that point becomes the goal for the ASV to dock at. Finally, to determine the correct bay to dock in we use the segmented images from the LIDAR data. Within this, we use a combination of HSV masking, edge detection, and color gradient detection find the color tiles and classify them.

G. Find and Fling

In the Find and Fling task, the strategy is to use similar strategies as the "Detect and Dock" task for finding the correct square to park in front of. Once parked, the boat will once again use HSV masking and edge detection to calculate where targets are in camera space. Taking into account the fixed trajectory of the ball shooter mounted atop the boat, the boat moves slightly to aim and fire at the targets.

H. UAV Search and Report

In the UAV Search and Report, the plan is to perform an S-shaped search pattern and have a camera pointing down from the drone. The drone will have an onboard raspberry pi 4, which will handle the vision processing component of the task. As it hovers over the posters, it will identify where it is in camera space, transform that to global frame, and report the finding to the main computer off-board the drone. Once the entire search pattern has been completed, the drone will return back to the initial location.

I. UAV Replenishment

Before departing from the dock, the WAM-V is carrying an UAV on the drone landing pad, a flat wooden panel located behind the vessel's computer. When the mission begins, the drone receives a command to launch from the landing pad. Using vision processing from the drone's attached camera, the drone locates the landing pad, upon which it gracefully descends. Through descending, the drone uses shape and color recognition to identify the location of the tin device on the landing pad and center on the device. After touching the landing pad, the drone activates its gripper and picks up the tin can. With the gripper still active, the drone departs from the landing pad, now carrying the tin can. The drone reactivates its vision processing pipeline with the goal of detecting the second landing pad. Upon detecting a white square object against the blue lake, the drone centers over the square and begins descent. After touching base with the landing pad, the gripper releases the tin can and departs from the pad. The drone then begins colocation with the boat by activating its beacon and determining how close the beacon located on the WAM-V is. After flying in the direction of the WAM-V, the drone begins a simple maneuver to land on the landing platform.

V. Experimental Design

A. Simulator

The first phase for testing new software for NaviGator AMS is simulation. We use the Virtual RobotX simulation from the Open Source Robotics Foundation as a platform for testing the vehicle, as this platform provides an effective scoring system similar to the one that will be used in the official RobotX competition. This simulator uses similar technologies to modern 3D video games to render images for the simulated vision cameras and LIDAR (see Fig. 6.) Most of the challenges present in RobotX 2022 are modeled in the simulator, allowing each task to be tested independently and in sequential runs similar to the finals of RobotX. Architecturally, the simulator uses Gazebo, an open source robotics simulator designed to integrate well with the ROS middleware we use. This allows us to run the exact same software in simulator as on the life platform, as the TCP socket interfaces for hardware (sensors and actuators) are fulfilled by the simulator. We added additional plugins to simulate the protocols of our student designed boards used for the emergency stop and passive sonar systems. This allows the simulated software to interface with device algorithms, including Bancroft's algorithm for the passive sonar device [6]. The simulated hardware enables testing the integration of these systems into the higher level software without having physical access to the system. Simulation also makes the development of high level decision making programs, known as "missions", to proceed in parallel to perception software [7].

B. Field Testing

In addition to testing in the simulator, NaviGator ASV underwent significant lake testing. Nearly 100 hours of in-water testing were carried out in the form of day-long tests in the months leading up to the competition at a lake near UF. Over 40,000 labor hours were accumulated during lake testing. Lake testing offered real-life environmental factors that simulation cannot accurately provide, such as wind and current disturbances, various lighting conditions, and inclement weather.

Field testing also offered a chance to test the mechanical systems of the ASV, such as actuators like the racquetball

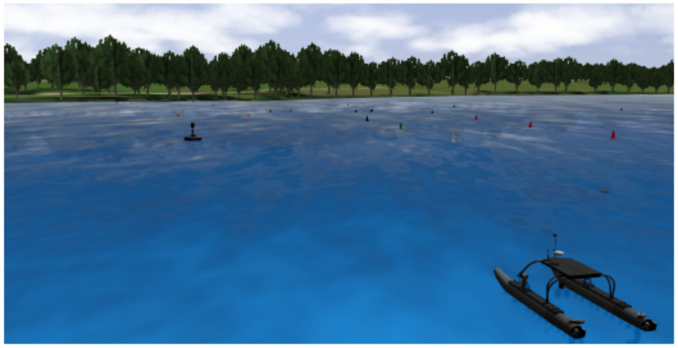

Fig. 6. Simulated NaviGator ASV in a realistic environment

launcher, the strength of team-manufactured components, and the efficiency of the computer cooling system. The frequency and duration of testing helped to expose failures that may have gone unnoticed until the competition. For instance, one issue we had was when the boat made sudden stops, its thrusters would stir up the water in front of it enough for the LIDAR system to detect it and perceive it as an object. Tuning the clustering algorithm parameters allowed us to avoid this issue without disrupting the functionality for other missions. The detection of this and other flaws during testing prevented what would have been catastrophic failures during the competition.

C. Field Element Construction

In order to take full advantage of the realistic testing environment that the lake provides, field elements similar to those that will be used in the competition were constructed. The field elements were designed to be simple in construction and easy to deploy. Many of the elements were made of a PVC pipe frame that allowed for modular construction and easy assembly and disassembly. Buoyancy was provided by foam sheets and pool noodles fitted around the PVC pipes. The simplicity and light weight of the course elements allowed for quick and easy setup and teardown of the course using only a few team members in a kayak (although recently we found that a paddle boat is even better than a kayak).

Attempting to recreate the field elements was a challenging task. With limited time and resources, the team reconstructed the dock and buoys as close as we could to the obstacles that would be present at the competition. Another challenge was testing the hyperspectral camera for the specific paint that would be used on the Wildlife Encounter tasks. The best we could do was to test on normal paint and attempt to re-identify certain type of paints [8].

VI. Conclusion

This paper presents the University of Florida's autonomous maritime system, NaviGator AMS (which includes our surface vehicle, NaviGator ASV, and our drone) for use in the 2022 Maritime RobotX Challenge. Sacrificing speed for maneuverability, the vessel's four thrusters give the ASV an additional degree of freedom when compared to traditional skid-steer vessels. The novel use of an automotive cargo box for housing electronics created an open layout design that allowed for easy access and rapid repairs. An iterative approach created a strong software foundation that was exhaustively tested with over 130 hours of in-water testing. Team NaviGator AMS is ready for the 2022 Maritime RobotX Challenge due to extensively tested software, simple mechanical design, and robust electronics.

VII. Acknowledgements

Team NaviGator AMS would like to acknowledge everyone who has supported the team since 2018, especially throughout the COVID-19 pandemic, including the University of Florida's departments of Electrical & Computer Engineering department and Mechanical & Aerospace Engineering, the labs of MIL and CIMAR, and our major industry sponsors: L3Harris Corporation, Texas Instruments, and Sylphase. The team would like to extend an appreciative thank you to our advisers: Dr. Eric Schwartz and Dr. Carl Crane. The latest Team NaviGator AMS developments can be found at www.NaviGatorUF.org.

References

[1] H. A. Lassiter, T. Whitley, B. Wilkinson, and A. Abd-Elrahman, "Scan pattern characterization of velodyne VLP-16 lidar sensor for UAS laser scanning," *Sensors*, vol. 20, no. 24, p. 7351, Dec. 2020. DOI: 10.3390/s20247351. [Online]. Available: https://doi.org/10.3390/s20247351.

[2] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[3] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "Lqr-rrt*: Optimal sampling-based motion planning with automatically derived extension heuristics," in *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, pp. 2537–2542.

[4] S. M. LaValle, "Rapidly-exploring random trees : A new tool for path planning," *The annual research report*, 1998.

[5] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation, bICRA 2014, Hong Kong, China, May 31 - June 7, 2014*, IEEE, 2014, pp. 1168–1175. DOI: 10.1109/ICRA.2014.6907001. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2014.6907001.

[6] M. S. Geyer and A. Daskalakis, "Solving passive multilateration equations using bancroft's algorithm," *17th DASC. AIAA/IEEE/SAE. Digital Avionics Systems Conference. Proceedings (Cat. No.98CH36267)*, vol. 2, F41/1–F41/8 vol.2, 1998.

[7] B. Bingham, C. Aguero, M. McCarrin, *et al.*, "Toward maritime robotic simulation in gazebo," in *Proceedings of MTS/IEEE OCEANS Conference*, Seattle, WA, Oct. 2019.

[8] Y. Mao, C. H. Betters, B. Evans, *et al.*, "Openhsi: A complete open-source hyperspectral imaging solution for everyone," *Remote Sensing*, vol. 14, no. 9, p. 2244, 2022.