Design of the Minion Research Platform for the 2018 Maritime RobotX Challenge

Jamie E. Barnes, Nate D. Bloom, Stephen P. Cronin, Grady C. Delp, Juan L. Halleran, Matthew R. Helms, James J. Hendrickson, Nicholas R. Middlebrooks, Nicholas D. Moline, James B. Near III, Jefferson S. Romney, Marco A. Schoener, Nicholas C. Schultz, David J. Thompson, Timothy A. Zuercher
Dr. Charles F. Reinholtz, Dr. Eric J. Coyle, Dr. Patrick N. Currier, Dr. Brian K. Butka, Dr. Christopher J. Hockley

***Abstract*— Embry-Riddle Aeronautical University (ERAU) has made significant improvements to their fully autonomous research platform, Minion. To complete mission tasks, Minion uses sophisticated sensory and perception algorithms fusing data from a suite consisting of four LiDARs, two wide-angle cameras, and a high precision GPS/INS. This data feeds path planning and decision-making algorithms that include neural network visual detection and tracking, 3D Multi-Variate Gaussian classification, and dynamic path planning.**

Taking lessons learned from the 2014 and 2016 competitions, the Minion platform was developed emphasizing refinement of existing systems. This allows it to meet the objectives of the 2018 RobotX Challenge and the demands of the team's research and teaching interests. This emphasis on refinement led to major improvements in controls, vision, and propulsion. This also allowed easy integration of other mission requirements, such as the racquetball turret and the autonomous underwater vehicle (AUV) deployment system.

All of Minion's systems are rated to survive operations in adverse weather conditions, including high temperature, high humidity, and heavy precipitation, and they have been tested in these environments. In the course of development, Minion was thoroughly tested using simulations, recorded data, and over 100 hours of in-water testing. The result of this is an advanced platform that is robust, reliable, and readily upgradable.

I. INTRODUCTION

A. Background and Vehicle Overview

Embry-Riddle Aeronautical University's (ERAU) Team Minion includes students ranging from undergraduates to Ph.D. candidates with backgrounds in Software, Electrical, and Mechanical Engineering. The team draws from experiences with many autonomous platforms, including entries in the AUVSI Foundation's RoboSub and RoboBoat competitions, as well as the previous two Maritime RobotX competitions.

From its inception in 2014, Team Minion has worked to create a platform that is rugged, customizable, and easily upgradable in order to meet mission requirements. All components are designed to withstand harsh environmental conditions including precipitation, humidity, and heat. In 2016, the Minion autonomous surface vessel (ASV) showcased the MAST (Minion Autonomous Systems Tray), which allowed mission-critical hardware to be suspended under the deck, enabling greater modularity.

For the 2018 Challenge, Team Minion further worked to improve the performance of the ASV by upgrading the propulsion system to allow for holonomic maneuverability. This allows Minion to sway (i.e., move sideways) to fully control position and heading simultaneously. These upgrades, along with software package updates, increased testing time, and a combination of improved custom and commercial-off-the-shelf (COTS) hardware complete a system ready for competition.

B. Software Overview

Software onboard Minion is broken into individual process modules that execute in parallel and communicate asynchronously using a publisher-subscriber messaging system. This enables modules to run at different rates and be selectively activated and deactivated, improving overall system efficiency. The competition software architecture is shown in Fig 1.

Sensing is handled primarily by a combination of a LiDAR-based Perception module, a camera-based Vision module, and a GPS/IMU-based State module. These modules leverage the strength of each sensing modality to detect and classify objects to create a world map for the autonomy modules.

The MinionTask mission tracker aggregates data from various modules and determines the best current objective to complete the mission. It communicates the objective to the Path Planner which calculates the optimal path, which the Controls module then executes. MinionTask also communicates the objective to the sensory modules, enabling and disabling processing algorithms based on the current objective.

Additional modules enable Minion to localize underwater acoustic targets with hydrophones, deploy and control an autonomous underwater vehicle (Anchor), and interface with a stand-alone racquetball turret (Bodyguard II). A custom Ground Station operator control unit enables efficient mission control over low-bandwidth datalinks.

The modules interact asynchronously through the MinionCore inter-process communications suite. The function of each module is discussed in Vehicle Design and in the Appendices.

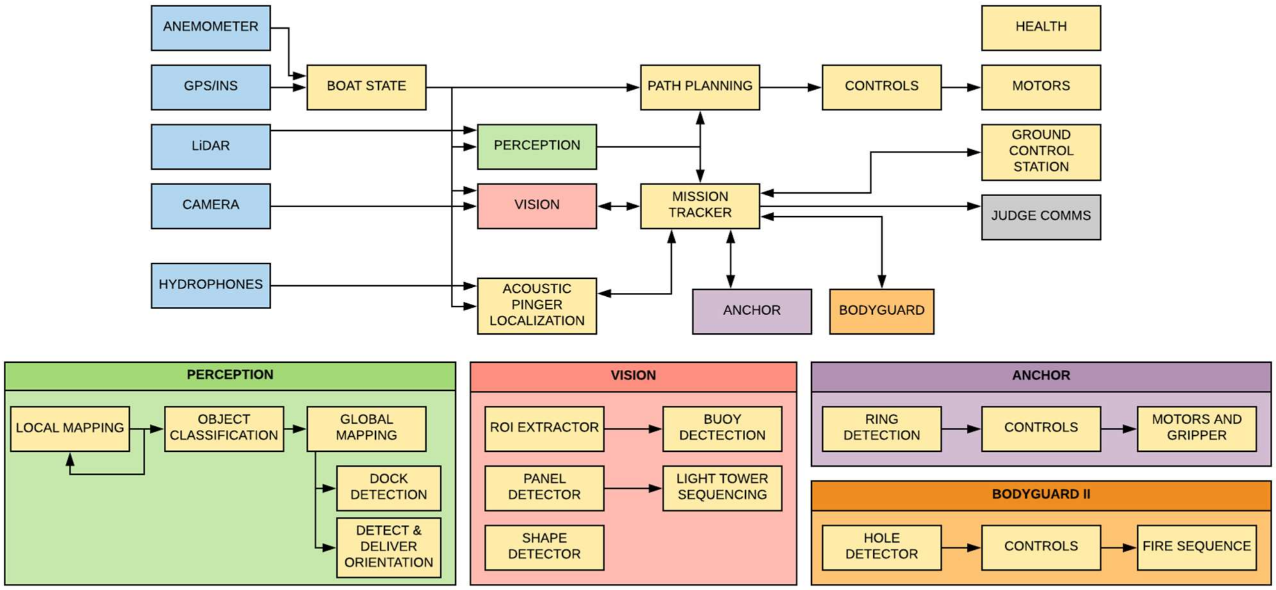

Fig 1. Minion ASV Software Architecture

II. DESIGN STRATEGY

Learning from the experiences of the 2016 competition, several goals were created to refine the platform's hardware and software to deliver a more robust system that shaped the redesign of the 2018 platform.

One key goal was the implementation of a simulation environment. This simulation allowed integration of the MinionTask, which drives the actions of the ASV, in a virtual environment. While a version of MinionTask was created for the 2016 competition, without the simulation capabilities, it was not robust.

Robust controls were a requirement for 2018 competition. The 2016 competition revealed problems in interfacing a controls system with a path planner and path following system. For example, the prior implementation did not have the ability to follow paths in reverse, posing issues with the docking challenge. The 2018 implementation needed to address these issues while also improving overall system robustness through failover techniques. In case of motor or azimuthing servo failure, the controls module was designed to adjust to the scenario to retain positive control of the system.

The prior perception suite primarily relied on LiDAR sensors for object detection and classification. For color and shape-related tasks such as Scan the Code, the LiDAR sensors provided the camera system with a bounding box to consider for sequence detection. This created a heavy reliance on the correct LiDAR classification of objects and an accurate transformation between LiDAR and camera sensors. In a maritime environment, the constant motion requires the accuracy of the transformation be very high. While one goal was to improve this transformation, another was to use neural networks to allow the cameras to detect the Scan the Code sequence from raw images.

A. Simulation Environment

For the purpose of testing mission and path-planning software without access to the boat, a rudimentary simulation environment was developed during the 2016 Maritime RobotX Challenge. This software was developed using hard-coded maps and variables, but it proved its usefulness and showed the utility of having a more robust and versatile software package in developing autonomy for the 2018 competition.

While a simulation environment running in Gazebo was available for all RobotX teams to utilize, the RobotX Gazebo model was released late in the competition cycle and did not match the capabilities of Minion, including the extensive perception suite and azimuth-capable thrusters.

MinionSim was developed through a series of intermediary milestones, allowing it to become immediately useable by the other modules in the software stack while slowly increasing its usefulness in further developing the other modules. Some of these milestones include producing a synthetic state (position and pose), interpreting the received control messages, sending objects the vessel discovers in the virtual environment, and describing the objects' visual features in ways that are useful to the other software modules.

The simulation software for the 2018 RobotX Challenge includes a mapmaking module for generating fields of virtual objects, and a simulation engine that interprets the files exported from the map maker that contain the object fields. Arbitrary maps can be created that contain configurations of objects and tasks expected in 2018 in order to test the ability of MinionTask. The simulation engine generates and distributes synthetic versions of the messages that other modules expect to receive from the physical ASV, allowing those other modules to be tested without needing the boat to be operating on the water. Other functions, including a hardware-in-the-loop mode, allow the simulation of virtual objects in the surroundings of the physical boat, which can be used to test path-planning and controls software.

B. MinionTask

Minion's MinionTask provides a unique capability designed to push the system towards the realm of true autonomy. The MinionTask does not script missions; instead the missions are designed through a series of tasks, each of which has defined start conditions, point values, and times to complete. The tasks also encode the object classifications and rule requirements for each mission in a common modular format that can be called by the mission engine. Tasks exist as separate compiled code and can be modified, added, or subtracted without modifying the mission engine.

Minion begins each mission in a search state with no knowledge of the course element locations and no predetermined task order. The search area encompasses the entire operating area and is searched in a pattern seeded by priority locations. As objects are discovered, a ready check function is run for each task to determine if the necessary conditions for initiating the task have been met along with an estimated execution time and point value. Tasks are selected and dynamically launched by the mission engine in real time to maximize the points scored per second of operation time.

C. Controls

Minion's 2018 control algorithms allow for robust handling of the new azimuthing system and improves upon 2016's path following algorithm. The control module is set up in a system that cascades control from the mission objective down to actuator allocation. This cascaded system, in combination with a new nonlinear optimizer, provides a far more robust control system with multiple advantages to the 2016 design. The most significant advantage is the ability to introduce failure, or "limp" modes. These modes enable the platform to operate on a reduced set of actuators and still accomplish the mission objectives. For example, if the port azimuthing were to fail, the system could account for full operation of the starboard actuators and operation of the port actuator at an arbitrary fixed angle.

D. Vision

Minion's vision module, which uses the visible imagers on both Minion and Anchor, supplements the information supplied by the perception module, which relies on LiDAR sensing. In 2016, Minion relied almost solely on the LiDAR system due to efficiency and reliability concerns with the Vision module. As many competition tasks require vision, it was a key to the 2018 strategy to address this deficiency.

Improving vision was addressed by using convolution neural networks (CNNs), which are a type of deep learning network, to increase the speed and accuracy of the vision classification and detection networks. The computational burden of these networks is also offloaded to the system GPUs to prevent slowdown of the other critical systems on Minion.

Ultimately, CNNs are trained using Tensorflow V1.5 to accomplish the Scan the Code task, identify buoy colors, and identify the shape and color of the target on the dock. These CNNs were all created from retraining already constructed networks. This design feature allows crossover of code between different networks and tasks. For the 2018 competition, the Mobilenet V1, Inception V2, and Inception V3 networks are all used to allow for a trade-off between speed and accuracy as well as classification and detection. Using this approach allowed for the easy implementation of a new network for the sub deployment task as well, since the code to run the networks is already compiled and all that is needed is retraining a network.

E. Electrical

To power and control the improvements to the propulsion, sensor, and payload systems, much of the electrical system has been improved from 2016.

The largest changes to the electrical system are the changes made for the new propulsion system. New RDPs demanded a change in motor controllers; new degrees of freedom in the propulsion and vehicle controls system required additional actuators and their associated power distribution and communications circuits. An off-the-shelf motor controller was integrated with the existing safety systems and a custom circuit board was designed to power and drive both the azimuth degrees of freedom and the thruster retraction actuators.

Each payload system, including the turret and the AUV deployment system, also required control circuitry. To improve maintainability of the whole system, the circuit board designed for the azimuth and retraction actuators included some extra peripherals that allowed its common use for the control of the payload systems, increasing field maintainability.

A new feature that has been added to the electrical systems is an upgraded external indicator system. Supplementing the basic light tower of the previous competition, a system of large LED arrays that is an order of magnitude brighter has been added to allow bystanders and operators to easily know the status of the system.

A problem that was found in the previous competition was that the vehicle's motor noise far exceeded the amplitude of the pinger signal, rendering the hydrophones useless while the motors were in operation. This problem has been remedied with the addition of a 5-stage analog filter and gain circuit that attenuates the motor noise.

Please see Appendix H for more electrical information, and Appendix F for more information on the hydrophone.

III. VEHICLE DESIGN

A. Design Process

The design process for Minion incorporates techniques from AGILE [1] for both hardware and software development. A 2-week sprint cycle was adopted, as well as weekly stand-up meetings. Stand-up meetings require every member of the team present their progress in a fast-paced manner with few technical details. This permits every member of the team to be versed in the progress of the entire project. A 2-week sprint cycle concludes with an in-water test on the final day. This provides a visible metric of progress for all team members, as well as incentive to accomplish all goals within a sprint cycle. A 2-week sprint may also conclude with a design review for projects that can't be tested within a 2-week sprint.

B. *Major Changes*

1) *Propulsion Redesign*

Following the 2016 RobotX Competition, Minion's propulsion system was redesigned to address weaknesses that were limiting the maneuverability of the platform. Two "motor pods" mount to existing hardpoints on the WAM-V, one at the aft of each pontoon. The pods serve a dual purpose of housing the propulsion system and providing required buoyancy to the WAM-V. Each motor pod houses a RDP thruster. These are a type of marine electric motor in which a brushless motor is built around the propeller to improve efficiency and reduce noise compared to electric trolling motors. It also minimizes the risk of tangling with seaweed or anchor lines and ensures the platform does not disrupt wildlife. Minion's 2016 Torque-Jet RDP thrusters been replaced with Copenhagen VM Asymmetric thrusters in 2018 for improved efficiency, thrust, and reliability.

The focus of the propulsion system redesign was to add azimuthing and beaching capabilities to the selected thrusters. Allowing the thrusters to rotate, or azimuth, improves maneuverability because it allows control over the magnitude and direction of force from each thruster, rather than just magnitude. Independent azimuthing also improves robustness as it allows the platform to operate and maneuver even if only one thruster is functional. To limit the impact of the additional complexity on reliability, the azimuthing thrusters can be mechanically locked, returning the platform to differential thrust. Azimuthing is achieved with Volz DA-30 servos, allowing the thrusters to rotate ± 85°. These powerful servos were chosen due to their environmental ratings and ability to rotate at 150°/s while producing 70.8 lbf-in of torque to smoothly and quickly azimuth the thrusters, even under maximum thrust.

The thrusters can also be retracted from the water, reducing the human intervention required when launching, retrieving, and beaching the ASV. Linak LA-36 linear actuators raise and lower the thrusters. Due to their worm drive, these actuators lock in position when not powered. Retraction and deployment each take under 10 seconds. Appendix D further details the design and analysis behind the new propulsion system. The final propulsion system is shown in Fig 2.

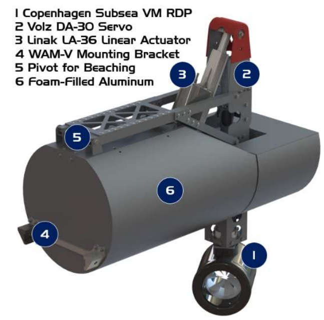

Fig 2. Azimuth and beaching enabled propulsion system.

2) *Racquetball Turret*

For the purpose of completing the Detect and Deliver task, Team Minion opted to develop a logistically simpler but more functionally robust solution than the air-powered turret used in 2016. The turret, Bodyguard II, is independently powered from the rest of the vehicle by a 6-cell lithium-polymer battery in a waterproof housing, rather than requiring a source of compressed air. This simplifies the beach operations that will be necessary to recharge the turret, and is accomplished by changing the method of firing from a compressed-air cannon to two sets of counter-rotating wheels.

The 2018 turret, shown in Fig 3, is mechanically simpler than in 2016. Bodyguard II operates on two four-bar linkages, powered by HiTec D845WP Waterproof servos. These servos, when operating at 6V, can produce 35 in-lb of torque through their 180 degrees of motion. The mechanical advantage of the four bar linkages, which reduces the azimuth range of motion to 60 degrees, and the altitude range of motion to 20 degrees, amplifies the torque, offering smoother and stronger responses to the changing positions and angles of the WAM-V deck and target.

The active targeting system built into Bodyguard II is enabled by a Microsoft Lifecam, with processing taking place onboard the turret using a Nvidia Jetson TK1. Once triggered by a task running on the primary Minion computer system, the video feed from the LifeCam detects the center of the Detect and Deliver target. The servos then move to position the target in the center of the camera view, and the turret fires.

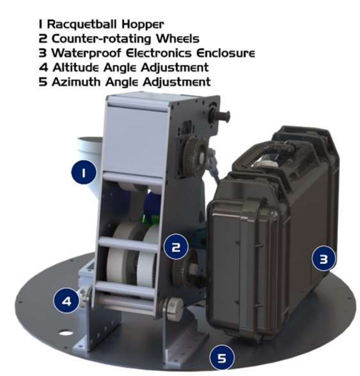

Fig 3. Bodyguard II Racquetball Turret.

3) Underwater Capability

a) AUV

Anchor, Minion's deployable AUV, is a BlueROV2 from Blue Robotics, and was selected because of its robust design, compact size, and open-source software. The BlueROV2 uses two Blue Robotics T200 thrusters to control depth and an additional four thrusters for holonomic control in the horizontal plane. Anchor is controlled by Minion via a Mavlink stream by processing a video stream from Anchor's onboard low-light 1080p USB camera.

b) Deployment

Anchor is linked to the ASV with the Blue Robotics Fathom Slim Tether that carries 2-wire ethernet and includes Kevlar, so it can act as both a physical link and data link. A custom ratcheting winch spools the tether to deploy and retrieve the BlueROV2. The winch itself is a large spool driven by an Ampflow A28-150-F48 48V motor through a 3-stage gearbox. The winch is mounted on Minion's modular under-deck payload tray, the MAST, shown in Fig 4.

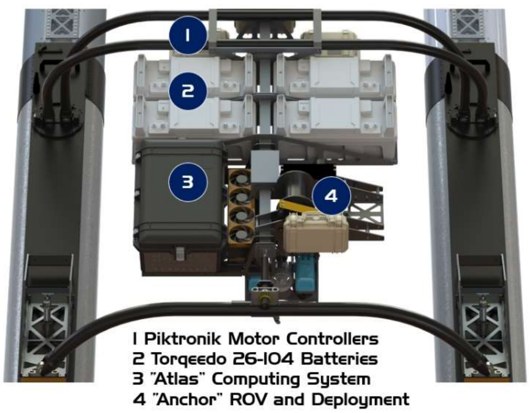

Fig 4. Submarine deployment module mounted on the MAST.

Since the deployment system's location on the MAST is port of the ASV's centerline, a spring-loaded swing-arm transfers the sub from the deployment tray to the centerline of the platform. This minimizes the risk of the sub hitting Minion's pontoons during retrieval, seen in Fig 5. More details on Anchor and its deployment system are in Appendix B.

Fig 5. BlueROV2, Anchor, and deployment showing swing-arm.

4) Path Planning & Controls Approach

Two modules are responsible for acting on autonomous behaviors: the Path Planning and the Controls modules. The Path Planning module is responsible for taking objectives from MinionTask and producing a trajectory that accomplishes the objective. The Controls module takes those trajectories and commands the vehicles actuators. For details on controls and path planning, see Appendix L.

The Path Planning module takes input of different objectives (waypoint, path with constant heading, station keeping, circling, docking) that produce trajectories for the Controls module. The other four, (stop, heading hold, point hold, direct) are special cases that do not produce trajectories and disable parts or all of the Controls module.

The Path Planning module coverts the nine tasking objectives into three possible controls modes: stop, direct, path. In the stop mode the vehicles actuators are disabled. In the direct mode the Controls module is receiving surge, sway, and yaw targets directly. The path mode causes the Controls module to follow a trajectory to an objective.

The trajectory controller in the Controls module is time-based leader-follower technique. Trajectories are smooth, continuous functions of time. The controller first calculates an error using the current vehicle state and the target state at time t. Then, the controller calculates an error between a future predicted state and the target state at time $t + t_{lead}$. A weighted average of the resulting control outputs is used to command the body controller.

The body controller consists of a set of gain-scheduled PID algorithms to control the following states: yaw, yaw rate, sway speed, and surge speed. The PIDs also have ramped inputs and output rate limits. The output of the body controller stage are a set of desired forces and moments. These are given to the actuator allocation stage.

The last stage of the Controls stack is actuator allocation. The allocation stage attempts to produce those forces and moments with the available actuators. The allocation problem is solved with a nonlinear optimization using quadratic sequential programming. The nonlinear optimizer considers the command limits of both the azimuthing actuators and the thrusters, as well as minimizing the amount of change in setting for the actuators. If an exact solution cannot be found, then a best fit solution is produced instead using a relative weighting of the objectives. The currently set modes are differential (no azimuth), full (complete azimuth), crutch mode (only one available azimuth), limp mode (only one motor/azimuth pair), and "twerk" mode (only a single, non-azimuthing thruster).

IV. Experimental Results

A. Test Approach

1) Simulation & Playback

In between tests, the software team uses simulation and playback tools to develop and improve algorithms.

2) Vehicle Shakedown

A vehicle shakedown takes place the day before a test to verify changes and ensure compatibility between modified code throughout the past sprint. This procedure helps ensure that time is not wasted during a test debugging incompatibility between the software modules.

3) Test Days

A test day occurs the final day of each 2-week sprint. In a standard AGILE method, this would be the member demo. During a test, the platform is deployed in the river with a chase boat and may be run in tele-operated or autonomous modes. These tests may be used to find discrepancies between simulation and the real-world environment. Similarly, it offers a good opportunity for logging data from the sensors to improve algorithms in the next sprint.

4) Logging

The Minion platform has multiple methods of logging to ensure all the relevant data is captured while on the water. Each module automatically logs all MinionCore messages or incoming sensor data while open. However, for some modules like vision, these logs are compressed and may not provide the best information for new algorithms. There is also a method for taking uncompressed manual logs.

5) Test Debrief & Sprint Planning

After a test, there is a debrief for all team members. This debrief covers everything that was accomplished during the test, as well as anything that needs to be accomplished for the next test. These debriefs are used to begin planning for the next 2-week sprint cycle.

B. Capabilities Testing

1) Perception

The Perception module is responsible for the detection of waterborne objects, mapping those objects, and determining if any of the detected objects are a competition object. The goals for this system are:

1. Detect objects within 25m to bow, port, and starboard, 10m to stern
2. Identify object location and size to within 0.5m
3. Map an area up to 1 sq. mile
4. Classify competition objects within 5 seconds of detection
5. Classify objects with over 90% accuracy and less than 20% false positive rate.

The detection, mapping and classification methods of the perception module are all detailed in Appendix E.

The accuracy of the detection and mapping system was found to be approximately 20cm for stationary objects within 20m of the vessel. This accuracy can be seen in Fig 6 where the pier pylons can be easily distinguished. However, all competition objects are floating, and can therefore have their point clouds distorted by the wave induced motion. The point clouds can still be easily recognized as the associated object, as evidenced by the TaylorMade Buoy and Light Tower object in Fig 7. Both Fig 6 and 7 were created using empirical data collected from on-water testing and then re-played through Minion's perception module.

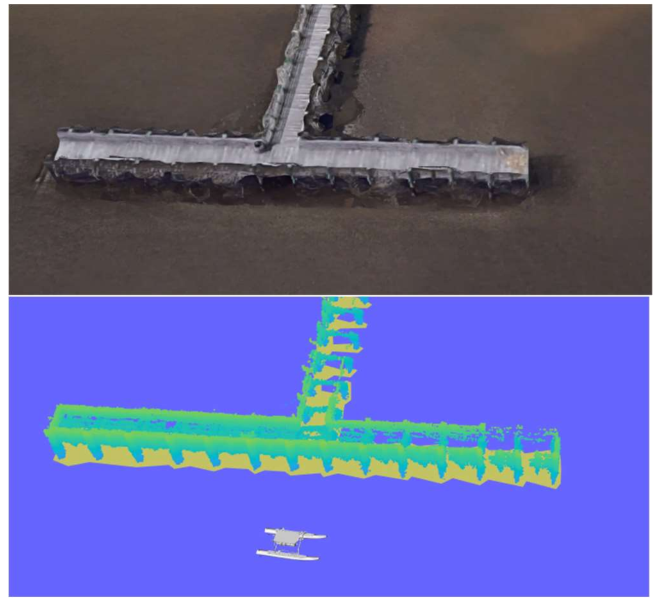

Fig 6. Satellite view of a pier compared to the 3D point cloud captured by Minion. The yellow polygon surrounding the bottom of the pier represents the mapped object boundaries used by the path planner.

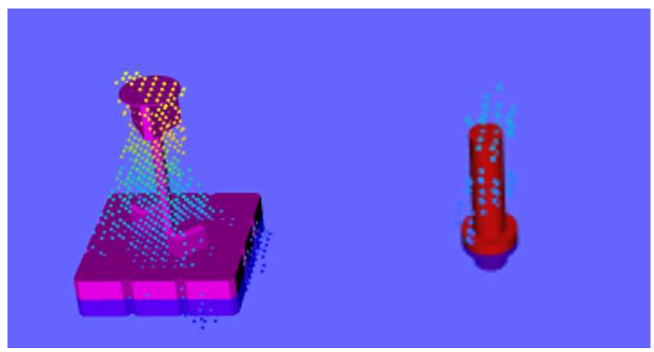

Fig 7. Point clouds for a floating TaylorMade Buoy and Light Tower. The CAD Model for each object has been placed on the figure in the location the object was detected. The CAD model for the light tower does not include the guide ropes on each corner of the base which are connected to the pole just below the panels.

Object features used in classification were collected from four competition objects during in-water testing, which are unique in spatial capabilities or near infrared reflectivity. The confusion matrix of Table I shows the accuracy of classifying these objects across a total of 1863 samples. It should be noted that a tall buoy with reflector refers to the green and red TaylorMade buoys, while the general Tall buoy class is a white TaylorMade buoy or totem.

Table I

CONFUSION MATRIX OF CLASSIFICATION RESULTS

		Test Class				
		Tall Buoy	A3 Buoy	Light Tower	Tall w/o Reflector	Unknown
Predicted Class	Tall Buoy	97.7%				17.4%
	A3 Buoy		97.2%	1.1%		9.2%
	Light Tower			95.8%		3.2%
	Tall w/o Reflector				98.5%	20.9%
	Unknown	2.3%	2.8%	3.1%	1.5%	49.3%

The results show that the goal of over 90% accuracy has been achieved with the lowest classification accuracy being 95.8%. Similarly, the false positive rate is under 20% for every class with only unknown objects even exceeding a 2% false positive rate. While not a specific goal, false negative rates are also below 4%. These results, combined with the history-based filtering discussed in Appendix E, allows the MinionTask to trust the class label given by the perception module.

2) *Vision*

The vision module is responsible for identifying the color of objects and shapes used in competition. Color classification is required to complete Scan the Code, while shape and color detection are required for the Detect and Deliver task as well as the Docking task.

a) *Light Tower Task*

A combination of three different CNNs was used to determine the sequence of the tower. The first network ran using the Faster RCNN Inception V2 (Coco) detection model trained with 50 proposal regions to crop the raw image from the camera down to the light tower. From the tower image, the single-shot detector (SSD) mobilenet V1 (Coco) was used to crop the image down to the light panel. Once the light panel image was obtained, a final network uses the Inception V3 classification framework to output the color of the panel.

This task ran at approximately 7 frames per second to obtain this number of samples per sequence. Next, to determine the sequence, a detector was implemented which used a moving mode to determine the color of each panel. Using this method, the sequence was output successfully in all cases except when the sun washed out the colors on the panel. In this case, the boat would rotate around the light tower and attempt to classify the panel in better lighting conditions. Fig 8 shows an example of the proposed regions (blue bounding box) of the light tower and panel.

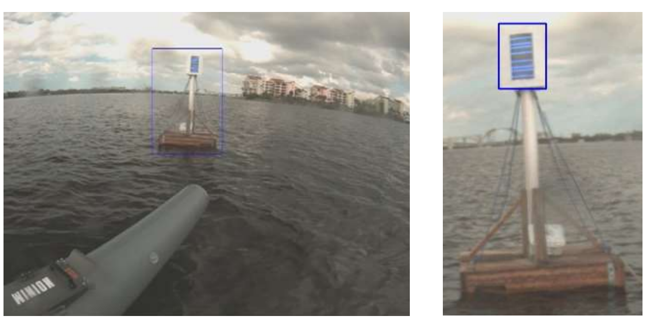

Fig 8. Light tower and panel prosed areas of interest.

The results in Table II show a robust system that can effectively crop down to the light panel when the light tower is in range during the task. These networks worked in bad lighting conditions as well since the CNNs did not rely heavily on color as a determining factor in the detection process. This was not true, however, for the panel color detection as this CNN was almost solely based upon color of the panel, since that is the only difference between red, green, and blue.

Table II

COMBINED SPEED AND ACCURACY OF LIGHT TOWER NETWORKS

Network	Purpose	Type	Speed [ms]	Accuracy [%]
Inception V2	Crop Tower	Detect	100-150	100
Coco Mobilenet V2	Crop Panel	Detect	20-40	99.7
Inception V3	Identify color	Classify	20-40	98.2
Combined	All	Both	140-230	97.9

An overall results and confusion matrix is shown in Table III for an 855-image set over two different sequences. This data shows that the color detector is robust as well and only fails to work well when the sun washes out the panel.

Table III

LIGHT PANEL COLOR CLASSIFICATION NETWORK RESULTS

		Test Class			
		Black	Blue	Red	Green
Predicted Class	Black	100.0%			
	Blue		96.4%		3.0%
	Red			100.0%	
	Green		3.6%		97.0%

The final step in predicting the sequence is the use of the sequence detector, which uses the results of the CNNs in real time. The detector uses a moving mode of the last five predictions to vote on the actual prediction. This is repeated until a sequence is detected; a black panel, 3 non-black panels, and a final black panel. A sequence is output once it has been voted for three times in the allotted time. If no sequence is detected in this time, the boat will circle around the tower and try once more.

b) *Object Classification*

The other use of vision for Minion was classifying the different colored buoys and the signs for docking. Both tasks were also completed using the Inception V3 classification network. These networks were retrained individually for the different tasks they were applied to.

For the buoys, the network was segmented into classifying the buoy as either red, green, blue, yellow, white, or black. Only color was of importance, so the type of buoy did not matter and was not accounted for in the classification. This was possible since the LiDAR data was accurate enough to classify the type of buoy.

A confusion matrix is shown in Table IV for the general performance of this network. For this task, if a color could not be identified with over a 70 percent confidence and a majority vote over a range of 20 images, the color is returned as unknown. This was done because a false positive is almost always worse than having an unknown color, as other logic can be applied to determine the corrective action to take.

Table IV

CONFUSION MATRIX FOR BUOY COLOR CLASSIFICATION

		Test Class					
		Red	Green	Blue	Yellow	White	Black
Predicted Class	Red	99.9%					
	Green		99.5%				
	Blue		0.2%	100.0%			
	Yellow				100.0%		
	White	0.1%	0.2%			100.0%	
	Black		0.2%				100.0%

The second classification network that was trained was for the docking signs. This network was used to detect the shape and color of the different docking signs. The same process was used for this task as was for the buoys. The network identified the color (red, green, or blue) and the shape (cruciform, circle, or triangle). Due to time limitations, there was no available data for testing this network.

3) Acoustics

Localization of the pinger was accomplished utilizing an ultra-short baseline array of four hydrophones and a multilateration processing algorithm. These sensors were arranged in a tetrahedron, allowing for the position of the source to be calculated as opposed to the bearing, which was employed in the prior implementation on the platform. This approach allows for more robust means of rejecting invalid returns, as the location can be compared to the buoys defining the gate. The specifics of the system as well as a detailed discussion of the algorithms implemented are in Appendix F.

At a high level, the goals of the system were to provide a more robust system than the prior iteration of the technology that could leverage the positioning information with enough accuracy. This translated into requirements would be:

1. Bearing accuracies (on surface plane) of ±5 deg
2. Positional accuracy within ±1m on (surface plane)

To validate whether or not the algorithm achieved these results, two forms of testing were employed. Signal simulation models allowed for rapid validation of the behavior of the full array; pool testing allowed for a controlled environment where the source could be easily moved and its location measured. These two forms of testing allowed for a confident deployment of the system on the platform.

To process the data, the raw waveform, Fig 9, is taken and the pulse, Fig 10, is extracted using frequency analysis. This allows the clean signals at the front of the incident pulse, Fig 11. At this point, phase analysis can be performed in order to compute the position.

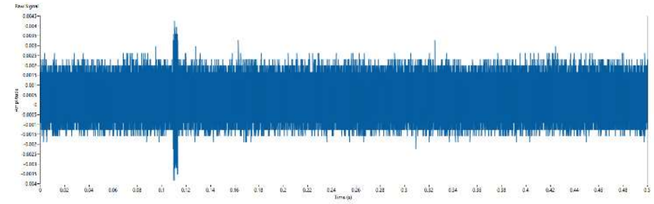

Fig 9. Raw waveform – 0.5 second capture.

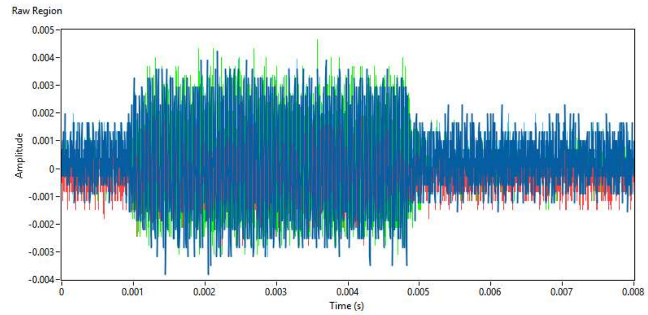

Fig 10. Extracted pulse (seen at t-0.15s above)

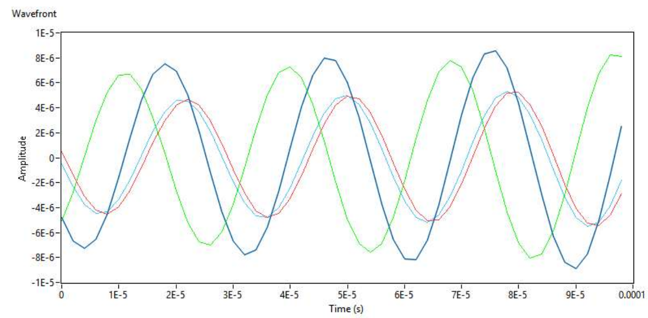

Fig 11. *Wavefront Signals (seen at t=0.001s above)*

Evaluating the position involves a numerical solution to the location of the pinger in XYZ space. For the signals above, doing so yields a position offset of (10,11,1) m. Based on the measurements of the pinger and arrays location, this results in a positional error of 0.2 and 0.18 m and an angular error of -0.1 deg in the surface plane. These performance results were repeatable in subsequent testing, achieving errors less than that of both the angular and positioning requirement.

C. Mission Testing

The break-down of each mission's testing, both in simulation and in-water, can be seen in Table V. The result of this extensive simulated testing was 58 total tests run across 5 different tasks, including: Navigation Gates, Acoustic Gates, Scan the Code, Obstacle Field and Totem Circling, and Docking. This extensive simulation testing helped ensure that the limited in-water test time was used to tune task run times and thresholds. As a result, the approximately 13 hours of in-water testing were used to refine the task parameters and to find edge cases to further test in simulation.

Table V

MISSION TESTING RESULTS

Mission	Simulation Test Hours	In-Water Test Hours
Navigation Gates	20.	5
Scan the Code	20	5
Acoustic Gates	10	2
Obstacle Field and Totems	8	-
Docking	-	1
Total	**58**	**13**

1) Simulation Tracker Results

Of the five tasks that were tested, four of these were able to be tested in MinionSim. The tasks that were tested in the simulator were the Navigation, Scan the Code, Acoustic Gates, and Obstacle Avoidance and Totem Circling tasks.

The Navigation task was the most heavily tested task in the simulator. Since it is the entry key to all testing that will need to be done on the course, it was critical that this challenge would be able to be robustly and reliably completed. In the simulator, the team was able to test several edge cases such as the gates being severely out of spec compared to the listed dimensions in the task outline. This included gates that were upwards of 40 meters apart and in skewed configurations, such as the case shown in Fig 12. Testing in the simulator also allowed testing of the platform's ability to complete this challenge both with and without color classification information being applied to the buoys. Through over 20 hours of simulator testing throughout the logic development phase, this task was proven to be highly reliable.

Testing of the Scan the Code challenge was fairly limited in MinionSim. Without the ability to simulate the sequence in a way that would allow the vision module to be tested, simulation was limited to checking the movement routines of the platform throughout the task. However, this did allow for around 20 hours of behavior and waypoint testing in simulation.

Similar to the Scan the Code task, limitations in MinionSim prevented the team from simulating pinger data, which would be used to determine the start gate to cross through. As a result, the platform would randomly guess and then transit through one of the three gates before completing the rest of the task.

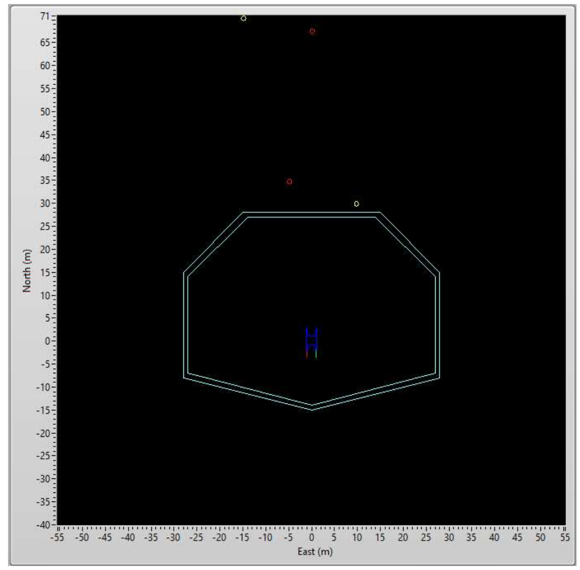

Fig 12. Navigation gate simulation using 40+ meter long, skewed gates

Although the pinger could not be simulated, MinionSim was more than capable of testing of edge cases and of situations that would be out of specification for competition requirements. Some of the edge cases tested included unevenly spaced gates, gates with the start buoys positioned above and below the line of fit (see Fig 14) and elongated start gates. The robustness of the module was also tested by simulating objects with incorrect or missing classifications as well as missing color identifications.

While testing this task, the circling of a specified totem was also tested, which allowed for behavior validation for the circling that would be done for the totem task. This also included testing the behaviors of the platform when only one buoy was detected in the region where a totem would be expected for circling. As a result, nearly 10 hours of successful simulator testing for the acoustic gate task was performed.

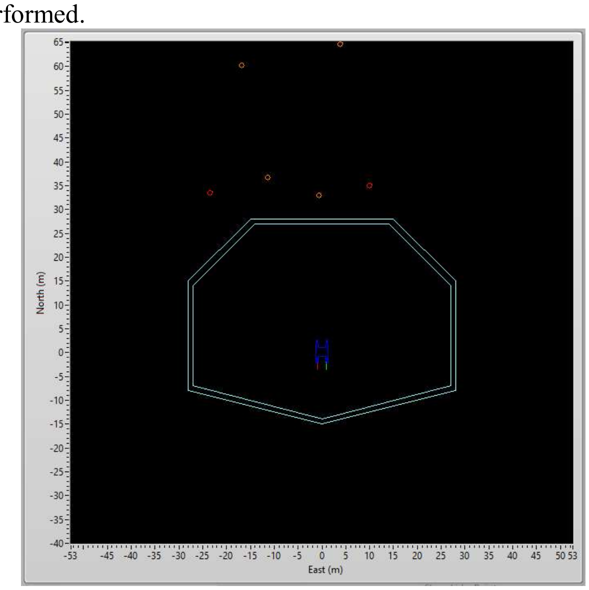

Fig 13. Acoustic gates with skewed, unevenly spaced entry gates.

2) In-Water Tests

As was to be expected, in-water testing revealed logic bugs and edge cases that were not initially considered when testing each of the tasks. During the time that was available for testing each challenge, four of the five tasks were attempted on the water. These included the Navigation gates, Scan the Code, Acoustic gates, and Docking challenges. Through all this testing, an impressive 13 hours of tasks testing was accumulated.

In-water testing of the Navigation gates challenge showed that several edge cases needed to be accounted for in the task logic. The first of these edge cases discovered was when the start gates were skewed. This was solved by adding a check in the logic for possible skewed gate conditions, further described in Appendix I. However, after this fix was implemented, it was found through in-water testing that it causes the boat to plot waypoints away from the end gate location to correct for what it thought was a skewed gate when only one of the end gate buoys had been classified. This problem was then solved by making the skewed gate case toggleable in the configuration file. Real world testing also showed that, due to potentially slow classification times, there was a need for this task to be able to find both the start and end gates with limited, and in some cases incorrect, classification information from Vision and Perception. After these changes were made, the navigation gates challenge was again attempted in the water. The platform was able to successfully navigate through the gates during 10 hours of in-water testing.

Testing of the Scan the Code challenge in real world conditions proved to be highly successful. This testing showed that scan angles, which allowed the sun to appear in front of or behind the ASV, hindered the sequence accuracy. The configuration files were changed to allow the platform to approach the tower at more ideal angles. Testing of this task also showed that the ideal scanning distance to get fast, reliable sequence returns was anywhere from 10-15 meters from the light tower, which was also edited in the configuration file. Testing of this task proved to be extremely successful with around 10 hours of successful in-water testing.

The Acoustic Gates task in-water testing revealed that the details about the ASV's real-world handling characteristics, primarily Minion's turning radius, were not accurately modeled in the simulation environment. As a result, it was determined that the circling radius for the end buoys was too tight, so this parameter was added into the configuration file so it could be tuned. It was also discovered that the intermittent waypoint generated after the ASV crossed through the gate, but before it went to search for the buoy to circle, was too close to the gate buoys. This would cause the ASV to make large, circular paths that would often put the ASV back through one or more gates while attempting to achieve the intermittent waypoint. This was then corrected in the Acoustic Gate task code by making the parameter for how far out the waypoint was placed past the gates a tunable parameter. Unfortunately, the only element of this task that was unable to be tested on the water was the gate detection via the hydrophones. However, even without this element, the ASV was able to successfully detect the gates, navigate through a randomly selected gate, find the required buoy to circle, and circle that buoy in the correct direction. As such, there were only 5 hours of successful attempts at this task on the water.

The Docking challenge was one that was only able to be simulated on the water through hardware-in-the-loop simulation of the dock. However, this did allow the logic for this challenge to be refined and tested. Through the in-water testing that was done, it was determined that the object growth that was done by the path-planner in order to ensure the ASV did not ram into obstacles prevented the ASV from being able to successfully complete the docking challenge. Thus, it was noted that there needed to be a direct mode in the Path Planner that would ignore obstacles in the way and simply drive to a point. It was also noted that a fall-back option that would be able to directly command the controls module, regardless of obstacles in the path, would need to be developed or revived. This resulted in emergence of Direct Mode, which overrides the path-planner and sends direct messages to the controls module. Both additions would allow the ASV to successfully complete the docking challenge. Unfortunately, this resulted in only around an hour of in-water testing for the docking challenge.

V. CONCLUSIONS

ERAU Team Minion has improved on its 2016 RobotX entry by improving the electrical, propulsion, controls, tasking, simulation, and vision capabilities to address the 2018 challenges. Additional new capabilities include azimuthing control and a deployable AUV. The system has been extensively tested with hundreds of hours of simulation development and over 100 hours of in-water testing. The result is a highly capable autonomous maritime system capable of competing successfully in RobotX 2018.

VI. ACKNOWLEDGEMENTS

ERAU Team Minion acknowledges the RobotX sponsors, AUVSI and ONR. The team also wishes to acknowledge its sponsors: Glenair, Velodyne Lidar, TORC Robotics, Volz Servos, Teledyne Dalsa, Solidworks, Mathworks, ERAU COE, ERAU SGA, Copenhagen Subsea, Pelican, the Paul B. Hunter and Constance Charitable Foundation and all other generous sponsors. Finally, the team would like to thank all those individuals who have helped us in this project.

VII. REFERENCES

[1] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. (2001). Manifesto for Agile Software Development Manifesto for Agile Software Development.

VIII. APPENDICES

A. Situational Awareness
B. Autonomous Underwater Vehicle
C. Racquetball Turret
D. Propulsion Systems
E. Perception Systems
F. Acoustic Systems
G. MinionCore Inter-process Communications
H. Electrical Systems
I. MinionTask Mission Planner
J. MinionSim Simulator
K. Vision Systems
L. Controls & Path Planner