Maritime Autonomous Surface Vessels: A Review of RobotX Challenge's Works

Xie Ming[1,a] and Sai Sahith Velamala[1,2,b]

[1]School of Mechanical and Aerospace Engineering, Nanyang Technological University Singapore, 50 Nanyang Ave, Singapore 639798

[2]Department of Mechanical Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore 575025, India

[a]<mmxie@ntu.edu.sg>, [b]<saivelamala96@gmail.com>

Keywords: USV, navigation, control, obstacle avoidance, ROV.

Abstract. The Maritime RobotX Challenge was created as a platform which showcases the advancements made in autonomous vehicle technology and to promote further research and development in the field. The competition requires participating teams to modify a 16' WAM-V USV in order to complete the challenge objectives including navigation control, obstacle avoidance, station keeping, code scanning, underwater symbol/shape identification and object delivery. This paper details the various hardware additions, control algorithms and software used by the participating teams on their competing vehicle in order to achieve maximum possible accuracy and efficiency of task completion.

1. Introduction

Launched by the Office of Naval Research (ONR) in 2012, the biennial Maritime RobotX Challenge was created to support research and development in autonomous vehicle technology. Hosted by RoboNation, a creation of the AUVSI (Association for Unmanned Vehicle Systems International) Foundation, the competition aims to engage, challenge, and educate students in the development of maritime autonomy through the principles of systems engineering.

The challenge consists of the following tasks:

1. *Demonstration of navigation and control* – The USV must navigate through two pairs of red and green buoys in a fully autonomous manner.
2. *Find totems and avoid obstacles* – The USV must detect and avoid a range of obstacle buoys while identifying distinctly colored floating totems and circling them in the correct direction.
3. *Identify symbols and dock* – The USV must autonomously locate the dock, identify the correct symbols/shapes and their associated docking bays, and then proceed into the correct bays in the correct order.
4. *Scan the code* – The USV must observe a light buoy to determine the sequential light pattern it flashes and report the color sequence.
5. *Underwater shape identification* – The USV must locate and identify objects/shapes on the seafloor in a specified area relative to a reference buoy.
6. *Find the break* – The USV must scan colored markers placed on or near the seafloor and count the segments between gaps indicated by other distinct underwater markers.
7. *Detect and deliver* – The USV must propel or insert objects through target holes on the correct face of a four sided floating platform after identifying colored shapes corresponding to the holes.

8. *Acoustic pinger based transit* – The USV must pass through assigned entry and exit gates which are marked by underwater acoustic pingers active at specific frequencies.

Table 1. Abstract functions required for the USV to perform given tasks.

Function			Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Manual
For task	Surface	Shape Detection			✓				✓		
		Color Detection	✓	✓	✓	✓			✓	✓	
		Relative Pos. Detection	✓	✓	✓	✓			✓	✓	
		Ball launch system							✓		
		Sound source Detection								✓	
	Underwater	Shpae Detection					✓				
		Color Detection					✓	✓			
		Relative Pos. Detection					✓	✓			
Navigation		GPS	✓	✓	✓	✓	✓	✓	✓	✓	
		Generating course	✓	✓	✓	✓	✓	✓	✓	✓	
Communication		Wi-fi	✓	✓	✓	✓	✓	✓	✓	✓	✓
Safety		Emergency stop(man.)	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Emergency stop(remote)	✓	✓	✓	✓	✓	✓	✓	✓	✓

The base vehicle which teams must use is a 16' Wave Adaptive Modular Vehicle (WAM-V) Unmanned Surface Vehicle (USV) built by Marine Advanced Research Inc. Additional hardware additions are permitted as long as the core vehicle structure is left unaltered and all modifications made abide by the competition rules.

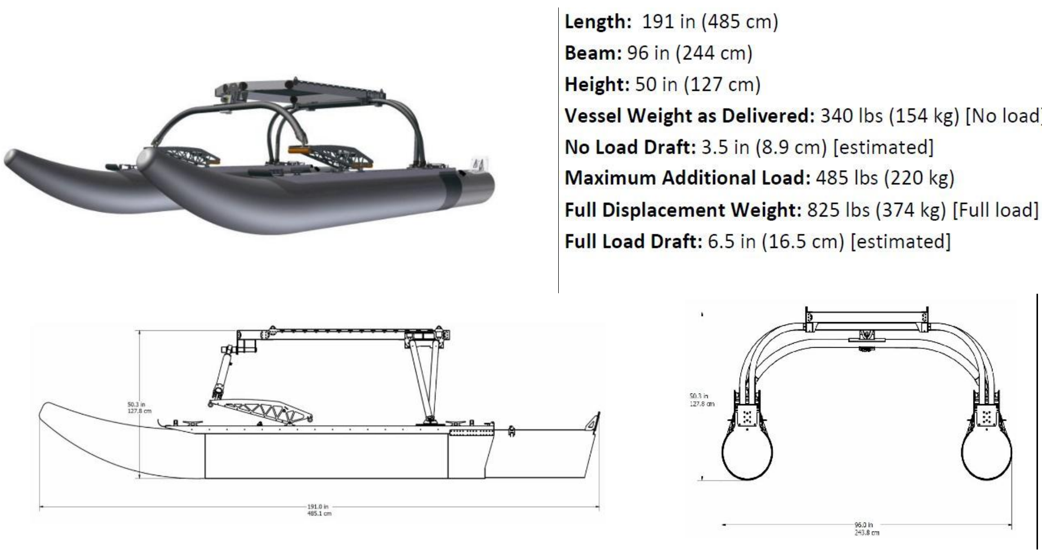


Fig. 1. The base 16' WAM-V USV and its dimensions.

2. Review of Research and Development Works by RobotX Challenge's Teams

2.1 Nanyang Technological University (NTU), Singapore

Hardware. NTU's Team Singaboat took a fairly straightforward approach to their vehicle. Their USV is equipped with an embedded PC in combination with STM32F4xx based microcontrollers. For wireless communication a TP-Link TL-ANT2415 2.4GHz 15 dBi omni-directional antenna and a TP-Link TL-WA7210N Outdoor Wireless N150 150 Mbps access point were selected.

The hydrophone for the acoustic pinger based transit task is a TC4013 reference hydrophone. Five AXIS M2014-E IP66 rated cameras are used for gathering visual data along with an IS16 industrial leddar for obstacle detection. All the hardware is powered by four 24V 70Ah Iron Lithium Phosphate batteries.

Software. The USV relies on ROS (Robot Operating System) Indigo running on Ubuntu 14.04 for both the embedded and ground station PC's. A wireless network established between them enables ROS message transmission. Pre-built ROS packages such as '`navsat-transform`' for parsing GPS data and '`razor-imu-9dof`' to get IMU data are used to drive the sensors. The thrusters are driven by ROS via Arduino coded with '`rosserial_arduino`' while odometry is taken care of by the '`move_base`' package.

Image processing is done using OpenCV. The images taken from the camera are converted into OpenCV images using `Image Transport`. All images are converted from BGR to HSV for better color filtering. Visualization of control algorithms is done using RViz and Gazebo along with ROS.

2.2 National University of Singapore (NUS), Singapore

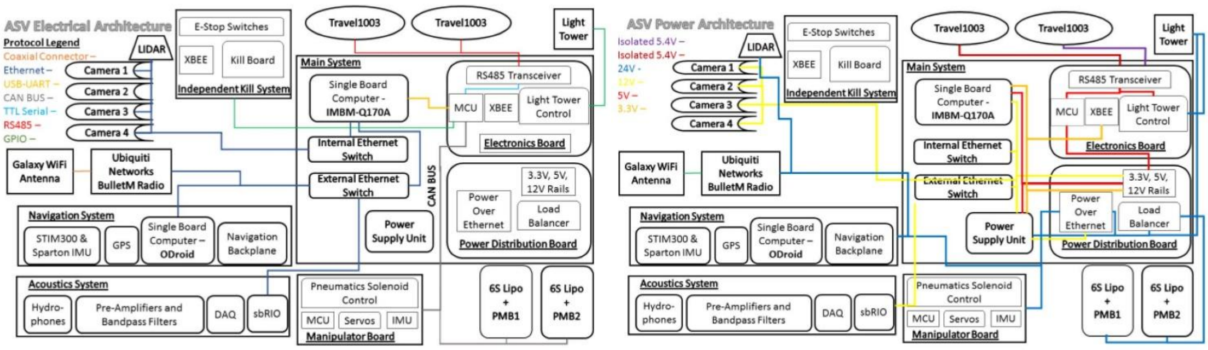
Hardware. The team from NUS chose an Intel Core i7-6700 powered single board computer alongside a NI sbRIO-9606 controller as the primary compute units. An ODROID XU-4 navigation computer was chosen to handle GPS and IMU based navigation. Wireless communication utilizes an XBee Pro 900HP kill link and a Ubiquity BulletM radio data link. Four Mako G-131 POE cameras are used for machine vision.

Two sets of pneumatic manipulators are used. One to deploy the array of 4 TC4013 hydrophones and the other - a set of four pneumatic powered actuators, for the payload delivery system. The launcher also uses servo and stepper motors for pitch and lateral adjustments.

The USV electrical system uses a combination of COTS devices and custom PCBs. Two 22.2V 10Ah battery modules, each equipped with their own custom Power Monitoring Board (PMB), power the electrical system.

Software. NUS's USV also uses ROS as the base for its control system to perform durable inter-process communications and enforce structure on individual task nodes and communication protocols. A high level task node called the mission planner node manages the other nodes and is written in Python for ease of modification at field tests. It makes heavy use of the ROS `smach` and `smachros` libraries.

For the acoustic pinger based transit, a phase difference based algorithm with high resolution and low computational requirement called MUSIC (Multiple Signal Classification) is used. It is ideal for embedded platform deployment.



(a) (b)

Fig. 2. (a) Communication protocol architecture and (b) Power system architecture of the NUS team USV

2.3 Seoul National University (SNU), South Korea

Hardware. Team SNU MACRO chose an ADlink MXE-5501 industrial PC as their mapping computer and a NI compact RIO 9024 computer for control. A SICK LMS511 2D LIDAR and two monocular cameras are used for object identification/shape detection.

This team also chose a pneumatic launcher for the payload delivery task. It uses a total of 3 motors, two for aiming and one for reloading.

Software. The USV uses low level *Reflexive Avoidance* and *Goal Position Tracking* for navigation and obstacle avoidance. For the underwater shape identification task, the image is binarized and then fed through a template matching algorithm (Fig. 3). The same algorithm along with color detection in the HSV color space is used for the find a break task.

For the detect and deliver task, the USV performs a dynamic positioning maneuver to maintain position. The target face is extracted from the LIDARs 2D point array using the RANSAC algorithm.

Fig. 3. Binarization of underwater image using Niblack binarization method.

2.4 Flinders University (FU), Australia

Hardware. The team's TopCat USV uses a dual antenna Trimble BX982 GPS system along with a Velodyne HDL-32 LIDAR and five Microsoft LifeCam Studio web cameras for navigation. A custom ingress proof housing was designed to house all the electronics safely while maximizing sensor coverage.

Software. The software design follows a 3-layer model. The top level for mission planning, the mid-layer for navigation and guidance and the lowest level containing control firmware. The top and mid layers are implemented as ROS nodes and the lowest level uses the Arduino framework.

Navigation is handled by the FastSLAM algorithm implemented using the Point Cloud Library (PCL). Classification of complex 3D shapes for the docking task is performed using Iterative Closest Point (ICP) algorithm.

2.5 Queensland University of Technology (QUT), Australia

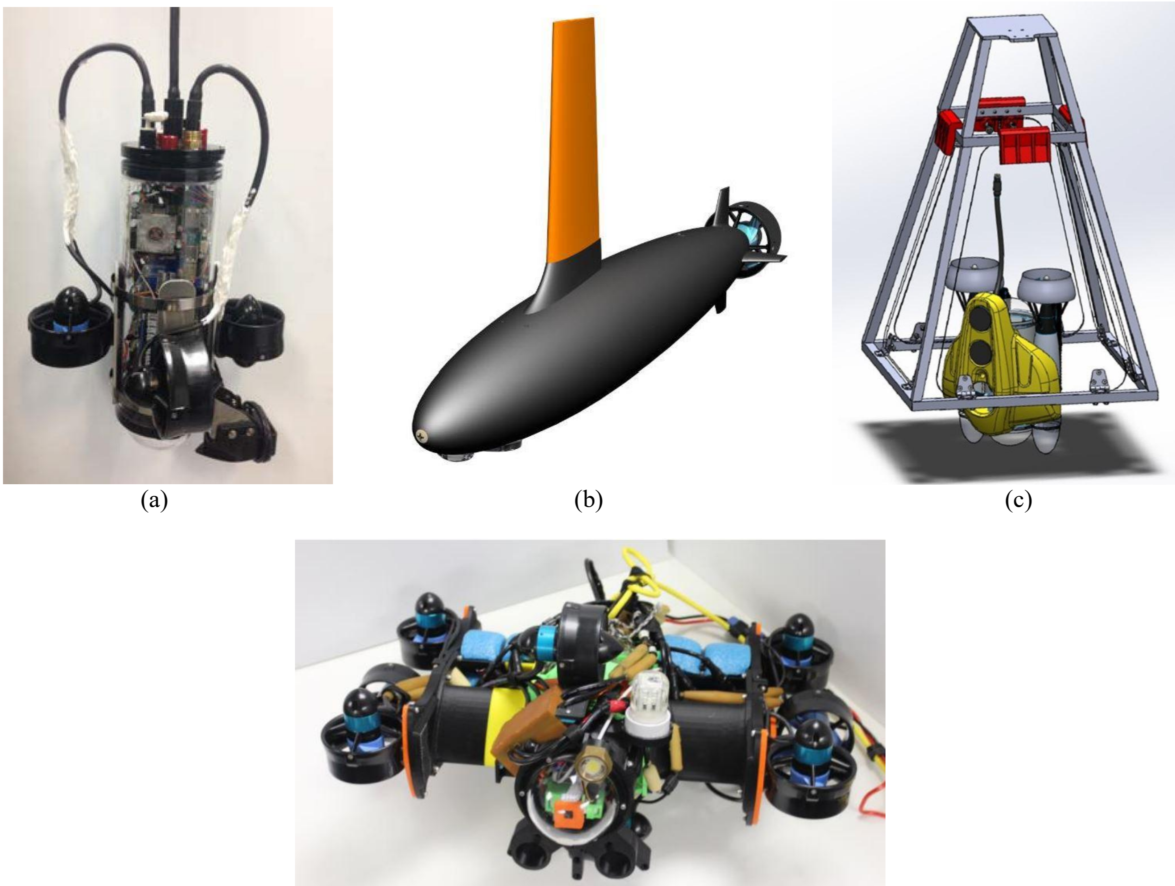
Hardware. The team from QUT, in addition to the standard navigational equipment, uses a Vaisala WTX-520 weather station to provide real-time wind speed and direction data at 1Hz.

For the detect and deliver task, the team took a different approach with the automated ball launcher using two horizontally opposed counter-rotating wheels and friction to jettison the ball.

For the underwater tasks, the team chose to use a separate Autonomous Underwater Vehicle (AUV) tethered to the USV to search nearby underwater areas for task markers. It is a self-contained AUV with all on-board power and is propelled by 4 x T100 Blue Robotics motors (Fig. 3.a). It uses a Microsoft Lifecam Cinema USB camera along with an Odroid XU4 for image processing. The tether to the USV is a simple high strength winch.

Software. The software architecture is built on ROS. It uses LIDAR data to generate a 2D occupancy grid for the obstacle map. The path planning is then done using a version of the A-star algorithm.

Due to some limitations in Gazebo and UWSwim, the team developed their own simulation software – The Autonomous Marine Surface Vessel Simulator. It uses OpenGL 3.0 for rendering and the JBullet physics library for buoyancy simulation.



(a) (b) (c)

(d)

Fig. 3. The subvehicles used by the teams. (a) Queensland University of Technology. (b) Embry Riddle Aeronautical University. (c) Florida Atlantic University-Villanova University. (d) University of Florida.

2.6 Embry Riddle Aeronautical University (ERAU), USA

Hardware. ERAU’s team uses a hanging system tray to hold all the components including batteries which lowered the CG of the vehicle by 120mm. It also uses a subvehicle based on the SWASH (Small Waterplane Area Single Hull) concept to act as an autonomous mobile sensor platform (Fig. 3.b). The subvehicle operates with minimal interaction with the water surface, resulting in a more stable sensing platform.

Software. The 4 Velodyne LIDAR returns are mapped into a fixed NED reference frame. This data is quantized and fills a 3D voxel grid which is flattened to yield a 2D occupancy grid to act as the local navigation map.

For the scan the code task, a deep learning Faster R-CNN algorithm based on the Caffe framework was chosen due to its speed and accuracy. The same algorithm was implemented for the other machine vision tasks. All the systems were tested and tuned on a custom MATLAB simulation environment.

2.7 Florida Atlantic University-Villanova University (FAU-VU), USA

Hardware. The team chose a separate VideoRay high performance Remotely Operated Vehicle (ROV) for the underwater tasks. This is deployed via a custom designed, winch operated launch and recovery system (Fig. 3.c).

A deployable acoustic boom holds the hydrophones.

Software. The USV uses ROS as the main control software with Windows integration modules for the ROV which has only Windows compatible APIs.

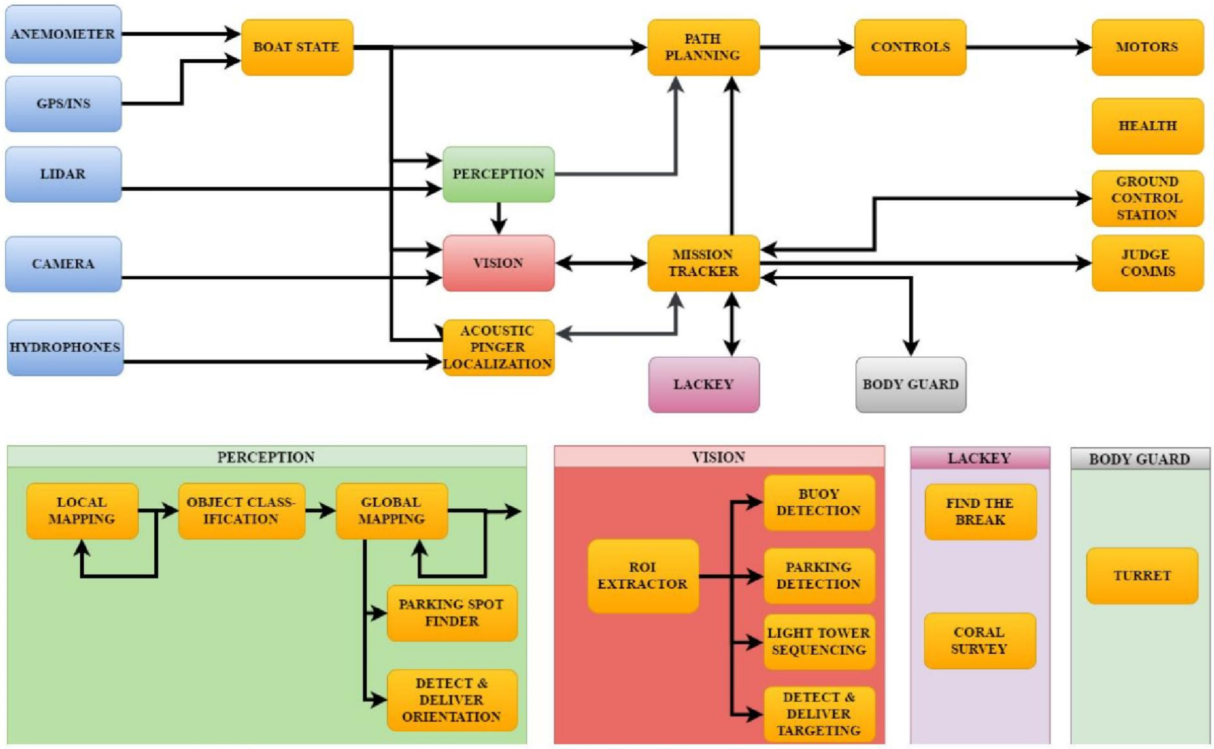


Fig. 4. General system architecture employed by the teams.

2.8 Georgia Institute of Technology (GIT), USA

Hardware. The team selected a Microstrain 3DM-GX3-45 INS navigation system along with a Velodyne PUCK VLP-16 3D LIDAR. The rest were standard components.

Software. For autonomous navigation, the DAMN (Distributed Architecture for Mobile Navigation) Arbiter and potential fields arbiter architectures were implemented.

A custom simulation environment, AAVS, was built from scratch in C#. It utilizes the OpenTK library for graphics and has custom modules for each of the systems sensors.

2.9 University of Florida (UoF), USA

Hardware. The launcher for the payload delivery task uses a spring loaded, closed bolt linear actuator system. The USV also has 2 additional bow thrusters angled outward at 45 degrees for ease of maneuverability.

The UoF team also chose to use a tethered ROV to assist the USV for the underwater shape identification and find the break challenges (Fig. 3.d).

Software. For navigation and path planning, the Rapidly-exploring Random Tree (RRT) algorithm was chosen due to its efficiency. A real-time ROS integrated LQR-RRT (Linear Quadratic Regulator) algorithm was the primary choice.

To compensate for disturbances in the environment, a Model Reference Adaptive Control (MRAC) architecture was chosen. Obstacle detection was implemented using the ROS implementation of Semi-Global Block Matching (SGBM) algorithm which relies on a modified version of the FAST detector.

3. Conclusion

Most of the teams chose to use similar, off the shelf commercial components for the main computing and navigational units. For the underwater tasks, teams such as those from Queensland University of Technology, Florida Atlantic University-Villanova University and University of Florida have chosen to use ROV's for increased visibility and underwater maneuverability. The payload delivery task saw the use of pneumatic, friction and linearly actuated launchers. Each team took a different approach to mounting the hardware on their USV, with some using rails, some using mounting cabinets and the rest using custom built enclosures.

On the software side, the majority of teams used ROS as the main control software due to its high flexibility and easy implementation of different nodes. Navigation and path planning was implemented using various versions of SLAM such as FastSLAM. For image processing and machine vision most teams chose to use OpenCV. Gazebo was also the simulator of choice. Exceptions were Queensland University of Technology and Georgia Institute of Technology, which designed custom simulators, and Embry Riddle Aeronautical University, which used a MATLAB simulation environment.

References

[1] Maritime RobotX Challenge website: https://www.robotx.org/

[2] W. Han, R. Ye, R. Maslim, L. Zhen and H. Chung, "Journal Paper for 2016 RobotX Competition", ***RobotX Forum***, 2016.

[3] G. Wei, A. John, S. Prawira, L. Jie, S. Ong, G. Chia, F. Jiaming, R. Chew, Y. Raaj, G. Bo, N. Zhi, L. Weitao, H. Che, A. Xuan, L. Xiang, S. Han, Y. Quanjie, T. Yeow, T. Jin, M. Tan, K. Ramachandra, "Design and Implementation of Bumblebee Autonomous Surface Vessel 1.0", ***RobotX Forum***, 2016.

[4] W. Suisei, K. Masaya, K. Riku, "Journal Paper About the Development for the 2016 RobotX Competition", ***RobotX Forum***, 2016.

[5] J. Lee, J. Woo, Hwiyong Choi, W. Jung, Hujae Choi, N. Kim, "Team SNU MACRO RobotX Journal Paper", ***RobotX Forum***, 2016.

[6] A. Webb, B. Donnelly, J. Stewart, J. Wheare, M. Kossatz, S. Hutchinson, S. Geyer, T. Crouch, A. Lammas, K. Sammut, "Development and Testing of the TopCat Autonomous Surface Vessel for the Maritime RobotX Challenge 2016", ***RobotX Forum***, 2016.

[7] L. Stanislas, P. Smith, B. Tidd, P. Kujala, S. Nicholson, A. Dawson, R. Lamont, W. Chamberlain, R. Barton, M. Eich, M. Dunbabin, "Bruce: An ASV Solution for the 2016 Maritime RobotX Challenge", ***RobotX Forum***, 2016.

[8] N. Bartlett, D. Byrne, J. Courts, M. Gibson, J. Paye II, T. Truong, G. Wakeham, "Maritime RobotX Challenge", ***RobotX Forum***, 2016.

[9] C. Ash, N. Bloom, J. Brown, A. Butka, S. Cronin, G. Delp, R. Goring, C. Hockley, Z. Joswick, D. Lodato, N. Middlebrooks, B. Mathews, J. Pletz, J. Romney, M. Schoener, N. Schultz, A. Thompson, D. Thompson, J. Wyble, T. Zuercher, C. Reinholtz, E. Coyle, B. Butka, P. Currier, "Design of the Minion Research Platform for the 2016 Maritime RobotX Challenge", ***RobotX Forum***, 2016.

[10] B. Mesa, T. Moscicki, W. Gresov, H. Davis, J. Keonig, L. Ridley, C. Tsai, C. Williams, W. Blummer, T. Cullina, M. Spillane, M. Dhanak, K. Ellenrieder, C. Nataraj, G. Clayton, "Team WORX Journal Paper for the 2016 RobotX Competition", ***RobotX Forum***, 2016.

[11] P. Meyer, S. Seifert, C. Ramee, E. Evans, W. Roberts, K. Cooksey, K. Griendling, "Why Yes, That is Plywood: A Simulation Based Approach to RobotX 2016", ***RobotX Forum***, 2016.

[12] D. Frank, A. Gray, K. Allen, T.Bianchi, K. Cohen, D. Dugger, J. Easterling, M. Griessler, S. Hyman, M. Langford, R. Leyva, L. Murphy, J. Nezvadovitz, A. Olive, B. Peterson, D. Solo, F. Voight, D. Volya, T. Williams, E. Schwartz, C. Crane, I. Hill, S. Ridgeway, "University of Florida: Team NaviGator AMS", ***RobotX Forum***, 2016.

[13] A. Argyris, I. Argyris, A. Bentz, K. Cortez, S. Cory, J. Leon, R. Duenas, R. Eidswick, N. Hortizuela, M. Huang, K. Ige, C. Izumigawa, M. Kang, S. Kim, P. Lemus, M. Loui, A. Nagamine, N. Park, K. Taylor, E. Whelton, A. Wong, D. Wong, "2016 RobotX Journal Paper", ***RobotX Forum***, 2016.

[14] M. Begneaud, E. Davidson, R. Kisor, C. Lagarde, J. Vaughan, "Design of the Vessel to Compete in the 2016 Maritime RobotX Competition", ***RobotX Forum***, 2016.