# University of Alberta
# Autonomous Robotic Vehicle Project
# RoboSub 2019: Auri Bites Back

Moira Blenkinsopp, Jacky Chung, Karen Han, James Hryniw, Noni Hua, David Lenfesty, Nathan Liebrecht, Andreea Muresan, Sophia Ngan, Gabriel Risbud-Vincent, Juan Rojas, Adesh Sangione, Andrew Schroeder, Curtis Stewart, Suzana Trac, Salina Trac, Rumman Waqar

***Abstract* — This paper details the design strategy and implementation of Auri, the Autonomous Robotic Vehicle Project's submarine for the RoboSub 2019 competition. This year the team focused on completing the pre-qualification task, barrel rolls, and the torpedo mission via sonar navigation. This was an interdisciplinary effort - building on years of iteration of our mechanical frame design, custom PCBs, and software architecture. A stable mechanical platform and gradual upgrades to our computer and sensor boards have provided the software team with months of additional testing time to develop and refine its perception, planning, and control systems. With these improvements, we are confident Auri has the ability to flexibly and reliably execute all targeted missions in an uncertain environment.**

## I. Competition Strategy

The Autonomous Robotic Vehicle Project's (ARVP's) goal for 2019 is to reliably execute a limited set of missions, without manipulation, in order to gain sufficient points to qualify for finals.

This year, one of the main decisions made by ARVP was to continue using the existing robot as opposed to manufacturing a new one. This allowed the team to focus on the end-to-end integration of all its mechanical, electrical, and software components. In particular, using Auri as a stable base for this years' competition enabled the team to build upon last years' work and conduct thorough testing, rather than allocating resources towards integrating the existing software with a new robot.

Our goal for RoboSub 2019 is to maximize the points for missions located near the dock, followed by the torpedo and surfacing missions. First, we plan to complete the coin flip task, then proceed to perform a barrel roll through the smaller section of the gate, followed by a spin. Auri will then advance towards the buoys and hit Jiangshi before touching the vampire on the buoy's back-side. At this point, Auri will employ sonar navigation in an attempt to fire a torpedo through the vampire's heart and surface within the octagon. We may also opt to attempt the random pinger task if time permits. Though simple logic is involved with the addition of this final task, Auri's speed may not be sufficient to surface in the octagon prior to returning to the torpedo task within the time constraints. The torpedo task is also typically located further from the dock.

Overall, this plan comprises the following course trajectory as shown in Figure 1:

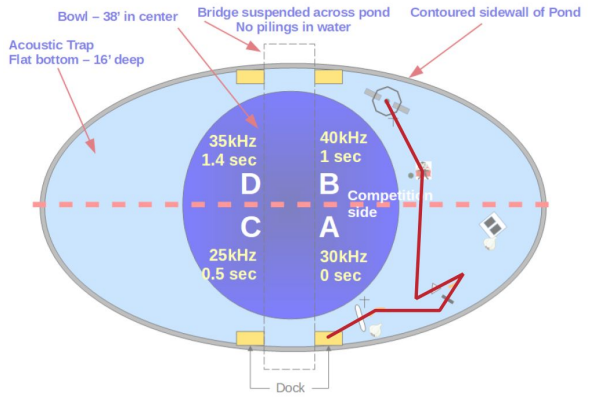


Fig. 1 Anticipated trajectory for the final course configuration

There is an aspect of simplicity associated with these tasks as the team can score very well without the need for manipulation or specialized droppers. However, ARVP was challenged with designing and implementing an effective system to navigate between tasks, which capitalized on the team's improved localization and mapping systems.

## II. Mechanical Design

### A. *Overall Design*

Going into its third year of use, Auri's structural design has proven to be robust, modular, and well-balanced. Auri's design was inspired by the fictional "TIE Fighter" spaceship from *Star Wars*. This aesthetic was kept, along with the acrylic cylindrical double-hull design, which adds ease of access to the electronics. With its aluminum construction and many braces, the frame is strong and lightweight, providing protection for all the interior components. Among others, it houses a DVL, torpedo launchers, a marker dropper system, and a hydrophone assembly. A fully rendered image of Auri can be seen in Figure 2.

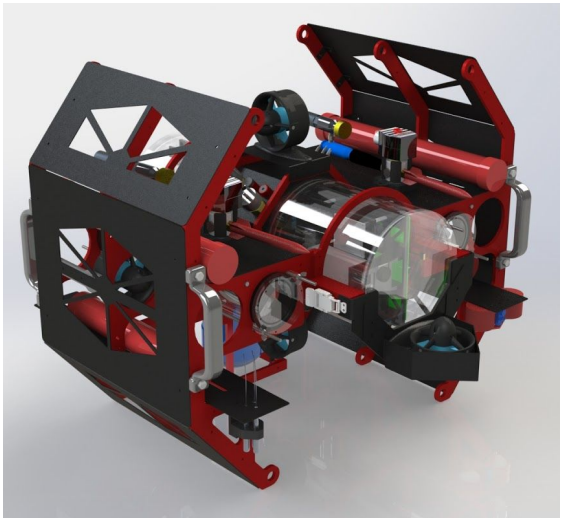

Fig. 2 Fully assembled render of Auri in SOLIDWORKS

### B. *Sealing and Pressure Testing*

The waterproof enclosures on Auri, including its hull, battery pods, and bottom camera enclosure, are all sealed with a proven double o-ring seal on a cylindrical surface. The acrylic tubes allow clear visibility of the o-ring seal and even pressure distribution. Each enclosure has been fitted with a vent plug for ease of opening and closing and to allow for pressure testing. ARVP has modified a bicycle pump to interface with the vent plug, reversed it into a vacuum pump, and completed it with a pressure gauge. All enclosures are vacuum pressure tested before water exposure to ensure sufficient seals and prevent leaks.

### C. *Electronics Trays*

The 2019 front electronics tray was redesigned to support the new NVIDIA Jetson Xavier. The contour of the new tray complements the Xavier base and incorporates a platform on which the processor rests. The front tray was 3D printed with ASA, which has a smaller thermal expansion coefficient compared to ABS and PLA, therefore reducing thermal deformation and enhancing structural support.

The rear electronics tray was also redesigned to improve spatial efficiency. The electronics boards are mounted on 3D printed panels that slide onto brass rods in a triangular configuration. This arrangement allows for increased cable management by routing the wires through the central portion of the assembly. LED strips and leak detectors are mounted on the base ring to communicate information to the team and to detect the presence of water in the hull, respectively.

### C. *Torpedoes*

The torpedo assemblies were designed to be compact and symmetrical while delivering adequate pressure to launch the torpedoes. This is accomplished by using carbon dioxide cartridges in a bucket changer directly mounted to an adjustable ASA regulator. Via several pipe fittings, the regulator is connected to a solenoid and then a steel tube with o-rings to hold the torpedoes. A 12 VDC electrical signal is used to actuate the solenoids, releasing the compressed gas and launching the torpedoes. Each assembly is attached to Auri with a 3D printed mounting bracket bolted into curved slots on the frame for easy adjustment of the firing angle.

## III. Electrical Design

By the end of the 2018 Robosub competition, ARVP's electrical system was better than it had ever been before. The modular design enables rapid repairs and the addition of new features, which transformed it into a highly advanced and capable system. The goal of the 2018-2019 design year was to enhance the already robust system while increasing reliability by replacing aging components. New enhancements were made in the form of critical and non-critical power rails, the ability to hot-swap batteries on the go, a new communications hub for the Xavier processor, and the research and development of a new integrated ADC sonar board.

### A. *Power Distribution*

This year, a unique power distribution system was developed to allow for improved modularity. The new power regulation system consists of a series of converter cards that can be slotted into a single carrier board. At the center of each card is a buck-boost DC/DC switching regulator capable of achieving up to 98% efficiency. The output voltage of each card can be precisely set by using two resistors in series configuration. A major advantage of this system is that the on-system voltage rails allows for increased flexibility. With three card slots on the carrier board, one can configure three cards to output 12 V, 5 V, and 3.3 V. The cards and the slots are standardized such that any card can be plugged into any slot.

In addition to the three edge connectors, the carrier board contains supporting circuitry for each rail to create a critical and non-critical version of each voltage rail. Two of the three connection pairs are critical, meaning that power is supplied to connected devices even when the kill switch is removed. The third pair is a non-critical rail, meaning that any connected device will be denied power when the kill switch is pulled. Having a two-tier power system allows critical computer systems such as the Jetson Xavier to remain on, while shutting down non-critical systems such as the torpedo system, servos, and lights. A render of the card and carrier power distribution system is shown in Figure 3 below.

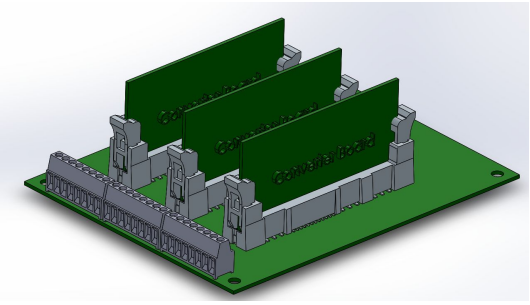

Fig. 3 Card and carrier power distribution system.

### B. *Battery Hot-Swap System*

On the battery monitoring board, hot-swapping capability and reverse polarity protection was incorporated to the electronics battery rail and all battery rails, respectively. The hot-swapping was implemented using an LTC4418 IC. This IC connects one of two valid channels based on voltage and priority. The valid voltage range for each channel is determined using a resistor divider chain based on the hysteresis current of the IC. This prevents the batteries from discharging too far and is within the tolerance of the MOSFET switches.

### C. *Cable System and Battery Upgrades*

As Auri will be attending its third competition, it was in dire need of upgrades to its electrical systems. Many 18 AWG 3-phase power cables for the T200 Blue Robotics thrusters were beginning to rust as insulation was removed by the penetrators through continuous stress and strain. New cabling was installed for these thrusters with heat shrink wrap strategically placed near the penetrators for added strain relief. Additionally, expandable cable sleeving was used to route the 3-phase thruster wires to maintain effective cable organization and management, something with which the team has struggled in the past.

In past years, Auri has relied on custom-made 6400 mAh 4s 35C lithium-ion polymer batteries to supply the electrical systems and thrusters. However, these batteries were difficult to repair and were not readily available. As such, the team has transitioned to using ZIPPY Compact 6200 mAh 4s 40C lithium-ion polymer batteries. Though the battery capacity is slightly less, they are much easier to obtain and have been shown to have a lower internal resistance, thus reducing voltage sag under heavy current draw.

## IV. Software Design

The purchase of a Doppler Velocity Log (DVL) fundamentally changed ARVP's software architecture. In particular, the DVL made it possible to localize underwater, which led the team to convert the entire software system to a map-oriented version of autonomy. As in previous years, the software is based on the open-source ROS framework, which provides a multitude of useful tools for robotics and allows for processes (nodes) to easily communicate via structured messages. This architecture naturally leads to the simple information pipeline shown in Figure 4:

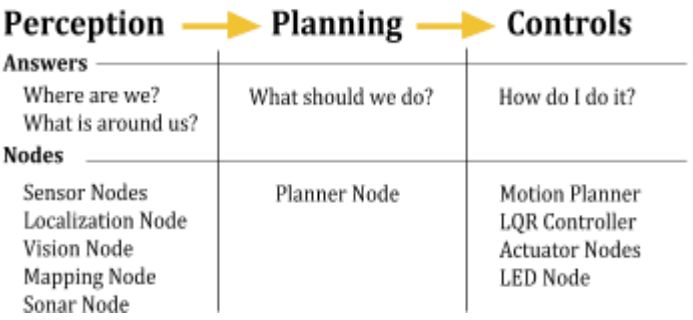


Fig. 4 ARVP's software pipeline

Important components of our software architecture are highlighted below.

### A. *Perception*

*Computer Vision*

As the buoy and torpedo task contain distinctive features, our vision system was transitioned to an entirely deep learning-based system to allow for object detection. We continued to use YOLO [1], but changed the underlying model, reducing the required floating point operations for inference from $3.5 \times 10^{10}$ FLOPs to $5 \times 10^{9}$ FLOPs. Given the similarity of background features in aquatic images, a less generalized and smaller model was able to achieve precision comparable to last year's model while running two and a half times faster.

*Localization*

This year, the team designed and implemented an Unscented Kalman Filter (UKF) [2] to better fuse the data from the depth sensor, IMU and DVL for robot localization. A constant linear acceleration and constant angular velocity 3D kinematics model [3] was used for the state transition model.

In the simulator, we were able to reduce the root mean square error of our positional states by roughly 30% after transitioning to this new system as shown in Figure 5:

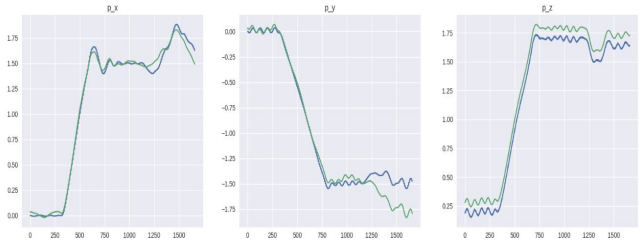

Fig 5. Predicted positional states vs ground truth

*Mapping*

The mapping node is the main tracking interface between perception and planning. It is used to store and update positional estimates of all competition objects.

We estimate the position of competition elements using a 2D to 3D projections of their bounding boxes passed from the vision node. These estimates are fundamentally noisy, especially at a distance, so a simple covariance model in function of distance was used in conjunction with an iterative product of multivariate Gaussians [4] to converge upon the elements' true position. Since this is prone to converge on early false positives, a minimum covariance on all estimates was enforced (process noise).

The mapping system requires initial estimates as priors, but these estimates are rapidly updated and corrected through the many detections from the vision system. Figure 6 below shows the map as viewed by the robot, with the translucent blue spheres representing the 95% confidence intervals of each element.

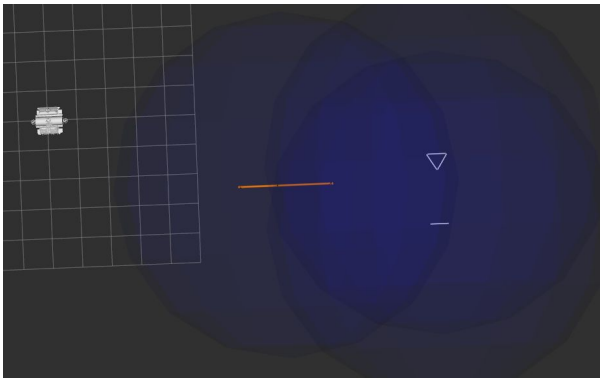

Fig 6. Map Priors - Testing course (RViz)

*Passive Sonar*

This year, we made major algorithmic changes in our sound source localization algorithm. In order to compute the time difference of arrival (TDOA) for a hydrophone pair, we switched over to generalized cross-correlation with phase transform (GCC-PHAT) [5] as given by:

$R_{x1,x2}(t') = \sum_{f=1}^{F} \frac{\Phi_{x1,x2}(f)}{\left|\Phi_{x1,x1}(f)\right|} e^{\alpha(t')}$, where $\Phi_{x1,x2}$ is the cross power spectral density of the two signals. The TDOA estimate is then the maximization of this function over the unknown time delay $t' : \widehat{\tau} = argmax_{t'} R_{x1,x2}(t')$. The position of the sound source *(S)* can then be calculated by solving the equation: $v\widehat{\tau} = \left|S - M_1\right| - \left|S - M_2\right|$, where $M_1$ and $M_2$ are the locations of the hydrophones and $v$ is the velocity of sound underwater. The equation develops into a two-sheet hyperboloid with foci in $M_1$ and $M_2$ and the sound source is on this surface. We can simplify this equation to a hyperbole by only considering the sound on the azimuthal plane (plane containing the microphones). Using two sets of hyperbole from our two pairs of hydrophones, the location of the sound source is the intersection of the two curves. Note that this returns two solutions. By using an initial estimate we are able to pick the correct solution.

To further improve our performance, we added a sound source tracking with a Sequential Importance Resampling (SIR) particle filter [6]. This made our system much more resilient to missing ping data as well as outliers. Figure 7 shows how initial estimates of the pinger location are narrowed down using the passive sonar system.

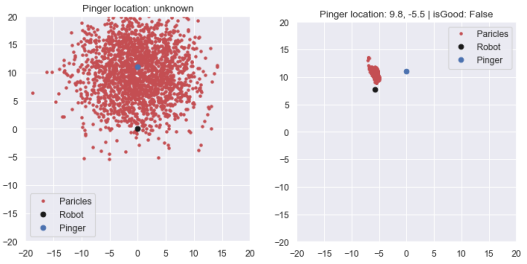


Fig 7. The system starts with an initial estimate and is able to narrow down the location of the pinger.

## B. *Planning*

This year, the team decided to upgrade our planning model from a graphical user interface (GUI) driven finite state machine (FSM) to a completely code-based and event driven solution in C++. This new model allowed us to scale our planning system while still handling asynchronous events including timeouts in a robust way.

By the end of last year, the team struggled with increasing the size and complexity of using the graph-based Rcommander GUI [7]. As more nodes were added, the FSMs became difficult to decipher and would lag the application significantly. Also, a lack of static checks would lead to frequent bugs. For example, it was easy to forget to handle error conditions such as timeouts which would cause a hang and thus fail the entire mission. To remedy this issue, we decided to simplify our mission planner and move towards a purely code-based solution with more static checks and guarantees.

For the new code-based mission planner, we were able to take advantage of the new *boost::outcome* library, which provided a solid framework to enable compile time error handling guarantees. This way, a developer would not forget to handle an error condition such as a timeout.

Beyond static checks, a key innovation in the new code-based planning system was to abstract the concept of time from mission development. In particular, state machines and other loop-based techniques have difficulty handling multiple asynchronous events like ROS messages and timeouts simultaneously. Taking inspiration from two highly asynchronous languages, Golang and node.js, this problem was solved by building an event handling system around Linux's poll(2) system call. Simply put, any

asynchronous operation such as a remote procedure call or a timeout is wrapped in a Linux file descriptor such as eventfd(2) or timerfd_create(2) and waited-on with a "select" function having similar semantics to go's select keyword. Furthermore, the entire system is single threaded as a result of Linux's poll(2) system call. This single threaded execution model eliminates the need to incorporate memory synchronization, greatly reducing the cognitive load on the developer.

### *C. Control*

*LQR Control*

For low level controls, the team focused on improving the functionality of Auri's Linear Quadratic Regulator (LQR) control system. For instance, the control inputs were normalized to better hold target velocities. This allowed the integration of the motion planning system, which uses velocity control. Also, the controller was updated to support arbitrary thruster configurations. This made it possible to evaluate different thruster configurations in the simulator, with the goal of determining which thruster configurations could be used to optimize the controllability of the vehicle.

*Motion Planning*

A setback with using LQR control for positional control is that there was no way to control the path the robot would take to move to its goal. This often resulted in very unpredictable and inefficient movement from the robot. In order to remedy this problem, we integrated the ROS based motion planning library move-base into our stack. We used the dynamic window approach (DWA) algorithm [8] for local robot navigation. This generates velocity commands to send to the LQR controller. The addition of the motion planner made it possible to have much more control over the robot's overall movement and has allowed the team to obtain more consistent results when performing missions.

## V. Experimental Results

Given ARVP's focus on conducting more thorough and frequent software testing, improving Auri's mechanical and electrical reliability were top priorities this year. In order to maximize in-pool testing time, Auri had to be functionable and operable to maintain a weekly testing cadence. This reliability was achieved by scheduling mechanical and electrical system "cut-over" weeks in which the two teams would radically change the robot frame and electronics within one week so as to minimize the robot's downtime.

Since last year, new thruster guards, additional cross bracing and better buoyancy adjustment mechanisms have all helped increase the reliability of the mechanical frame. Notably, the team now relies on checklists and streamlined procedures such as vacuum pressure tests and motor tests to ensure that all systems are functioning properly before all pool tests.

As usual, unit tests, simulation and continuous integration are the backbone of ARVP's testing infrastructure. Overall, the team created dozens of unit tests, ran over 2000 continuous integration (CI) jobs, and tested our robot in the simulator for upwards of 1500 hours. However, real-world tests are still critical to the team's software development. The club has accumulated about 60 hours of total pool testing in an olympic-size dive tank. Since early May, these tests have been run twice a week with weekday tests focusing on motion planning and control, while weekend tests focused on testing continuous missions with a full course setup.

## VI. Acknowledgements

It has taken many years for ARVP to flourish into the streamlined, self-sufficient organization that it is today. Beyond the dedication and contribution of the team members, the group could not have done any of its work without the support of many partners, advisors and mentors.

In particular, ARVP would like the thank the Faculty of Engineering at the University of Alberta and its staff for providing generous funding, space, and tools so the team can do its best work. In particular, Raymond Matthias, Erin Lee, Rebecca Blanchette and Don Villacencio have been key allies throughout the year. We would also like to thank our advisor Dr. Bob Koch for all his help with approvals and advice regarding the team's control system and Robert Donovan for his ongoing support both as a mentor and as a diver during our pool tests.

Finally, we would also like to recognize all our corporate donors, without whom our robot would be made of sticks:

- Gold Sponsors: APEGA Foundation
- Silver Sponsors: Shell Canada, Honda Canada, Nortek, Rail Services Co., Travis CI

## REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[2] E. Wan and R. Merwe, “The Unscented Kalman Filter for Nonlinear Estimation,” Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373). IEEE, 2000.

[3] T. Fossen, “Handbook of marine craft hydrodynamics and motion control” 1st ed. Chichester, West Sussex: Wiley, 2011, Section 2.2.1.

[4] C. E. Rasmussen and C. K. I. Williams, “Gaussian Processes for Machine Learning”. The MIT Press, 2006.

[5] C. Knapp and G. Carter, “The generalized correlation method for estimation of time delay”, IEEE Transactions on Acoustics 1976, Speech and Signal Processing ASSP-24(4), 320-327.

[6] N. Gordon, D. Salmond, A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation”. IEEE Proceedings F on Radar and Signal Processing 1993 140(2), 107–113.

[7] H. Nguyen, “RCommander: a Photoshop for Robots” [Online]. Available: http://wiki.ros.org/rcommander _core. [Accessed: July 8, 2019].

[8] D. Fox, W. Burgard, and S. Thrun. "The dynamic window approach to collision avoidance". Technical Report IAI-TR-95-13, University of Bonn 1995.

# Appendix A

## COMPONENT TABLE

| Component | Vendor | Model/Type | Specs | Cost (if new) |
|---|---|---|---|---|
| Buoyancy Control | Home Depot | PVC | 2” PVC pipes + end caps | $40 |
| Frame | ARVP | Custom | Custom aluminum waterjet | n/a |
| Waterproof Housing | ARVP | Custom | Custom CNC enclosure | n/a |
| Waterproof Connectors | Subconn | Ethernet and power | 25A / contact (power) | $700 |
| Thrusters | Blue Robotics | T200 | Brushless thruster | $1480 |
| Motor Control | Zubax Robotics | MYXA-B ESC | Closed-loop controllers | Sponsored |
| High Level Control | ARVP | 18-state LQR | LQR controller | n/a |
| Actuators | Blue Robotics | HS-646WP | Waterproof servo | $35 |
| Propellers | n/a | n/a | n/a | n/a |
| Battery | HobbyKing | ZIPPY Compact | 6200mAh 4s 40c LiPo | $720 |
| Converter | ARVP | Custom | 100Wx3, (12V & 5V used) | $450 |
| Regulator | Anddo | - | Outputs 0-200psi | Sponsored |
| CPU | Nvidia | Jetson Xavier | 8-core ARM processor, 512-Core Volta GPU | $799 |
| Internal Comm Network | CAN, I2C | n/a | n/a | n/a |
| External Comm Interface | Ethernet | n/a | n/a | n/a |
| Programming Language 1 | C++ | n/a | n/a | n/a |
| Programming Language 2 | Python | n/a | n/a | n/a |
| Compass | LORD Microstrain | 3DM-GX5-25 | AHRS | $2500 |
| Inertial Measurement Unit (IMU) | LORD Microstrain | 3DM-GX5-25 | AHRS | $2500 |
| Doppler Velocity Log (DVL) | Teledyne Marine | DVL 1000 | DVL | $18000 |
| Camera(s) | GoPro | Hero 3+ | Monocular, HDR | $350 |
| Hydrophones | Teledyne Marine | TC4013-1 | 1Hz-170kHz, Omni | $4500 |
| Manipulator | n/a | n/a | n/a | n/a |
| Algorithms: vision | pjreddie | Darknet / YOLO | Fast generic SSD | Free |
| Algorithms: acoustics | ARVP | au_sonar | DSP + Particle Filter | n/a |
| Algorithms: localization and mapping | ARVP | au_localization / au_mapping | UKF + Gaussian updates | n/a |
| Algorithms: autonomy | ARVP | au_planner | No FSMs, preemptable functions | n/a |
| Open Source Software | ROS | Various | Multiple ROS packages | Free |
| Team Size (number of people) | 58 | | | |
| HW/SW expertise ratio | 12:5 | | | |
| Testing time: simulation | 1500+ | | | |
| Testing time: in-water | 60 | | | |