University of Alberta
Autonomous Robotic Vehicle Project
RoboSub 2018: Auri Strikes Back

Rumman Waqar - Project Lead, Jonathon Machinski - Admin Team Lead, Curtis Stewart - Mechanical Team Co- Lead, Ryan van Drecht - Mechanical Team Co-Load, Alain Letourneau - Electrical Team Lead, James Hryniw - Software Team Lead, Juan Rojas, Nicholas Wengel, Nathan Liebrecht, Jacky Chung, Noni Hua, Willard Farmer, Moira Blenkinsopp, Andrew Schroeder, Mike Bardwell, Koltin Kosik-Harvey, David Lenfesty

***Abstract*—The paper presented details the design choices and experimental trials of the Autonomous Robotic Vehicle Project leading up to the RoboSub 2018 competition. Building on last year's successes, the team returns to San Diego with an improved hull design, sonar, revamped sensor communication and expanded software capability. Overall, the robot is more performant and reliable than ever.**

I. INTRODUCTION

This year, the Autonomous Robotic Vehicle Project's goal was to address the failings of the previous robot, Auri, while expanding on its strengths.

A new software design has optimized the control and vision system while reducing technical debt.

The electrical system uses standardized variations of our well-tested designs to ensure the robot performs reliably and also adds sonar capabilities to Auri.

The mechanical design improved the frame's strength and expandability, allowed the robot to be balanced with any component configuration but kept the ease of access from the previous frame iteration.

II. MECHANICAL DESIGN OVERVIEW

Going into its second year of use, Auri's frame was redesigned to improve rigidity and balance, as well as to provide mounting locations for new components required for the 2018 Robosub competition. Auri's original design was inspired by the fictional "TIE Fighter" spaceship from *Star Wars*. This aesthetic was kept in the re-design, along with the aluminum and acrylic double-hull. To reduce effort from the control system, the horizontal thrusters and battery enclosures are now positioned on the planes about the center of mass. Further, this frame is easier to assemble than its predecessor because it is self-standing without the central hull in place. Slots were integrated into the trays to allow for easy adjustment of weights to balance the robot, even with different component configurations. Additionally, the new frame has been designed to mount a DVL. Mounting locations for the new dropper system and hydrophone assembly are also included. With six degrees of freedom and a weight of 74 lbs, Auri is ARVP's most ambitious robot yet.

A. Hull

Auri's hull was reused this year, with small changes made to accommodate the new subconns. A central aluminum hull is used to hold two electronics trays vertically or horizontally. A clear acrylic cap is used on each side with a double o-ring seal to complete the double-hull design. This design allows for clear visibility and quick access of the electronics trays. The cylindrical shape of the acrylic tubes allows for even pressure distribution to reduce stress, while the octagonal shape of the aluminum section provides multiple flat surfaces for easy installation of penetrators and subconn connections.

6061 alloy aluminum was chosen for the central section because of it is not magnetic, and thus does not interfere with the electrical components. This alloy is also highly machinable, allowing for the hull's custom octagonal shape.

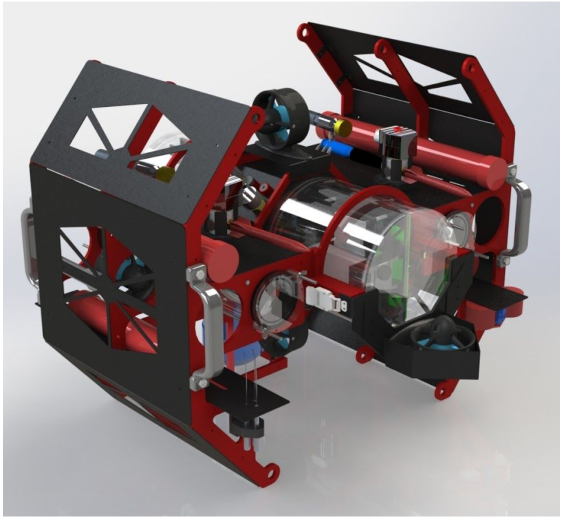

Fig 1. Fully Assembled Render of Auri in SOLIDWORKS

B. Frame

The 2018 redesign increased the size of the frame to allow for the mounting of a DVL and future mechanisms such as a cooling system or mechanical arm. The new frame is also more structurally stable and balanced. As with the previous frame iteration, all components are mounted within the wings of the frame for protection. Moreover, a more open design has been cut into the side panels to improves hydrodynamics while strafing. Finally, by integrating the battery pods into the ribs of the frame at the midline, the center of mass and buoyancy are now closer together, meaning the robot is able to hold different orientations with less effort from the control system.

C. Battery Enclosures

Auri has two separate battery enclosures, which are now slightly longer to accommodate the new batteries. These enclosures are now embedded into the frame at the midline. The battery enclosures are made out of 3.5-inch acrylic cylinders, which are sealed at both ends with aluminum double o-ring flanges. The caps of the assembly are designed with 1/2-inch acrylic ends and have subconn electrical connections.

D. Marker Release Mechanism

Auri has been equipped with a new dropper system remade to drop gold balls. The dropper was 3D printed out of ABS plastic to reduce weight. A waterproof servo is used to turn a 3D printed arm, releasing the balls. The arm was design to allow for one ball to be dropped at a time.

E. Torpedoes

The torpedo assemblies were designed to be compact while delivering adequate pressure to launch the torpedoes. This was accomplished by using CO2 cartridges instead of an air tank and compressor. The torpedo launching module consists of a CO2 bucket changer directly mounted to an on/off ASA regulator. This regulator is connected through a 90-degree fitting to a variable pressure regulator which adjusts the CO2 pressure to a desired value. The variable pressure regulator is then connected to a two-way solenoid valve. After the solenoid is a steel tube with o-rings to hold the torpedoes. An electrical signal is used to actuate the solenoids, releasing the CO2 and launching the torpedoes. The torpedoes were 3D printed from PLA filament. Each assembly is attached to Auri with a 3D printed mounting bracket. This mounting bracket is bolted onto curved slots in the upper tray of the frame to allow for the angle of the torpedoes to be easily adjusted.

III. ELECTRICAL DESIGN OVERVIEW

The electrical system in this year's robot is comprised of modular and standardized printed circuit board (PCB) designs, upgraded wiring solutions, and circuit protection improvements. This is all mounted in the same compact, easily removable, 3D printed panels used in last year's design. This allows the robot to reliably and efficiently accomplish the tasks it sets out to do.

A. Electrical Power Delivery

Four 14.8V, 94.72Wh lithium-polymer batteries are used to power the robot; three batteries power the thrusters and one powers the remaining electronics. Each thruster battery is connected to two Zubax Myxa B electronic speed controllers (ESC) and all batteries operate independently from each other. The batteries are housed in two sealable acrylic cylinders, separate from the main hull, and feed the robot through Subconn Power Series waterproof cables, capable of carrying up to 25A per contact.

B. Power Regulation

The computers and embedded systems require either 12V or 5V from a stable source, thus a voltage regulation system is used to convert 14.8V from the battery to the acceptable levels required by the electronics. The regulator board uses two Murata UWE series eighth-brick isolated DC-DC converters which use a step-up/down topology. A small expansion board is mounted to the main regulator board to monitor power output of each rail and sends data through an i^2c interface.

C. PCB Design Standardization

Each embedded system in the robot uses the Teensy 3.2 microcontroller and all have identical communication interfaces if they are required to communicate with the main computer: an Nvidia TX2. This allows for many different backup communication options. During normal operation, UAVCAN is used for reading and commanding the Teensys.

D. Battery Monitoring Board and Kill-switch

This board monitors power delivered by the batteries using Texas Instrument's INA3221 power monitoring integrated circuit (IC). It can also interrupt current flow from the three thruster batteries using low-side NMOS transistor switches, which are turned on only when the kill-switch is activated. The kill-switch uses multiple redundant hall effect sensors to detect the presence of an external neodymium magnet. Only when a magnetic field is detected, the switch allows the transistors to turn on.

E. Multi-purpose Actuation Board

The primary purpose of this board is to actuate the 12V solenoid valves that launch the 3D printed torpedoes. Low-side NMOS transistors with carefully designed snubber circuits, to minimize high frequency ringing from the solenoid coils during turn-off, are used to actuate the valves. Its secondary functions include RC servo motor and RGB LED strip control, using NXP Semiconductors' PCA9685 PWM generator IC.

F. *Internal Environment Sensor Board*

To monitor the hull's internal conditions, this board is used to monitor air temperature, pressure, and humidity using NXP's MPL3115A2 and Honeywell's HIH7120 ICs. Using these sensors, air leaks can be detected by looking for fast, abnormal changes in the hull's internal environment.

G. *Communication Hub*

This board allows the TX2 to connect to a CAN bus for interfacing with the ESCs and embedded systems, using the MCP2562 transceiver. It also provides i^2c busses for connecting to sensors, such as Blue Robotics' 300m depth sensor.

H. *Hydrophone Signal Conditioning Board*

Before the signals provided by the Teledyne TC4013 hydrophones can be processed by the signal signal processor, they need to be amplified, filtered, and DC biased. The board uses Analog Devices' AD8336 variable gain amplifier, the LTC1264 switched-capacitor filter, and the LTC6242 and LT1804 op-amps for voltage buffering, clipping, and peak detection. A custom designed +/-5V, on-board power supply is used to allow for pure AC amplification and filtering.

IV. SOFTWARE

Auri's software system can be subdivided into several distinct modules; *Control System, Computer Vision*, *Simulation, and Mission Planning.* All of these components are connected through the Robot Operating System (ROS), an open source communications library. ROS was chosen as a software framework because it provides a multitude of useful tools for robotics and allows for processes to easily exchange data asynchronously. By abstracting the data transfer layer, each node is completely modular.

A. *Control System*

This year, Auri's control system was completely redesigned. In previous years, the team had used a PID control system which presented numerous challenges. Most notably,

- There was significant amount of time being dedicated to calibrating the controllers; and
- The control system could only effectively control one axis of movement at a time.

Upon researching what types of control systems had historically performed well at RoboSub, ARVP decided to develop a Linear Quadratic Regulator (LQR) control system. The main focus of an LQR control system is to achieve certain criteria for a given system as efficiently as possible using an optimal control law. The optimal control law is a set of differential equations that minimizes a cost function which typically depends on the state and control variables. An LQR control system generates the control law using on four matrices. These matrices are the A, B, Q, and R matrices which model the physical dynamics, control dynamics, state cost, and control cost, respectively [1].

There were two main objectives for the development of the control system:

1) Develop a dynamics model (which describes the physical behaviour of the robot) that could be used as a basis for the A and B matrices.
2) Integrate the controller with the rest of the software architecture.

I. Dynamics Model

ARVP used Thor I. Fossen's robot-like vectorial model for marine craft as a basis for the dynamics model [2]. The model consists of various matrices that describe different aspects of the robot's physical behaviour. Each matrix is a function of at least one of the eighteen states, which consist of the robot's pose, orientation, linear & angular velocities, as well as the integrals of the pose & orientation error on the X, Y, and Z axes.

Each matrix is also a function of constant robot-specific physical parameters. While some of these parameters were easily obtained, some (such as the added mass coefficients) had to be approximated because they can only be obtained via complicated and costly experimental setups. Once all parameters were defined, the model had to be linearized so that its nonlinear elements would not affect the LQR controller. This was done by taking the Jacobian of the non-linear model with respect to the state and the control forces which generated the A and B matrices, respectively.

Lastly, the Q and R matrices were given arbitrary values to begin with, and were later refined based on experimental results and testing.

II. Software Integration

With the linearized model in place, a preliminary LQR controller was developed using MATLAB. MATLAB's symbolic and numerical capabilities allowed the team to generate a symbolic A matrix, which made it possible to apply the linearization at each iteration (i.e. in real-time) about the current state.

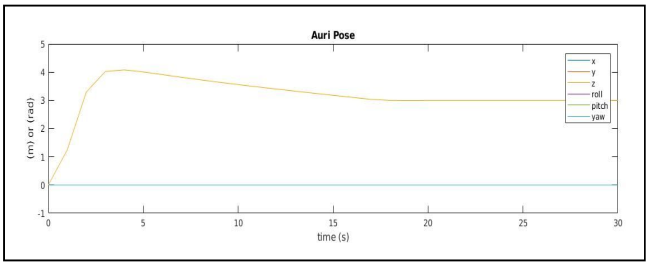

Fig. 2 Early Depth Test Using MATLAB LQR Controller Code

Once the MATLAB LQR controller's performance was deemed adequate, the controller code was re-written in Python so that it could be used in competition. This involved a significant change to the existing control system architecture, as the team went from a single-input single-output system to a multi-input multi-output architecture.

III. LQR Controller Performance

From initial LQR testing, Auri's performance in the pool demonstrated a noticeable improvement. The robot could now hold more complex orientations and was more stable compared to the PID controllers. Figure 3 shows Auri in the simulator steadily holding a state with 90 degree roll and yaw angles, while maintaining its depth.

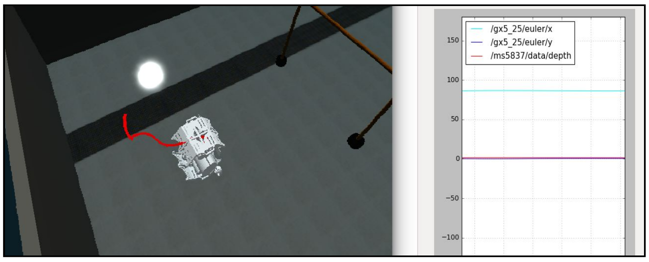

Fig. 3 Auri steadily holding a Complex Orientation

B. Computer Vision

Underwater, there is poor color contrast and lighting, and for this reason Auri's vision algorithms are designed with color invariance in mind. Auri relies on two techniques to isolate objects without explicitly declaring tight color ranges:

- A 2D histogram to isolate colors with high contrast to an overall image, and
- Image segmentation using superpixels.

These techniques are used to extract contours from images, which are then used for object identification through a shape matching process.

I. 2D Histogram Technique

To isolate contours of interest in an image, we first perform background subtraction on our images using a 2D hue and value histogram. From this histogram, we extract a list of disconnected clusters, representing the different color regions in the image. By removing the largest clusters, we can create a binary mask on pixels in the image that most stand out. Contours are then from the binary mask using OpenCV.

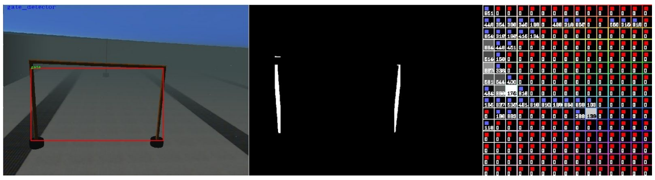

Fig. 4 Auri detecting the gate using the 2D histogram technique

This technique works under the assumption that the received image's background will be roughly one color, and that the foreground will have some minimum amount of contrast. It's primary advantage is its simplicity and speed.

II. Image Segmentation with Superpixels

Whereas the histogram technique looks at an image as a whole, image segmentation was used to account for color locality. The basic idea of this technique is to divide an into a grid of roughly equal sized chunks called "superpixels" which each consist of a single average color. This is accomplished using a third party library called gSLICr. Once an image has been segmented, good contours can be obtained by thresholding.

III. Shape Matching

Shape matching is the concept of classifying contours by matching them to pre-generated contours of a particular object. This technique is generic, and allows new detection models to be trained efficiently. For each target object, we render its 3D model at incremental orientations using OpenSceneGraph. Using our image segmentation strategy, we then extract the contours and store their translation and scale invariant format in a database. The process is left rotation variant in order to reduce match collisions and allow for 3D pose estimation. For speed purposes, CUDA is used to GPU accelerate comparisons between perceived contours.

Fig. 5 Auri detecting the path using shape matching

III. Deep Learning

In 2018, ARVP continued to use the YOLO [3] as a primary means of object classification. However, the introduction of dice this year presented many new challenges. For example, the similarities between dice, as opposed to the distinct colors of the buoys, were much more difficult for the neural network to differentiate. Another issue was the lack of representative training data. Whereas the team already has plenty of footage of buoys, we had to start from scratch for the dice. To tackle this problem, we generated synthetic images to increase the amount of data available for training. This was done by creating a Transdec Pool scene in Unity, allowing us to capture images of the object and their respective labels. This resulted in a slightly better average IoU, 73.99 vs 71.53, when using both synthetic and real images as compared to training only on real images. Labelling time was also reduced because the synthetic data set is pre-labelled.

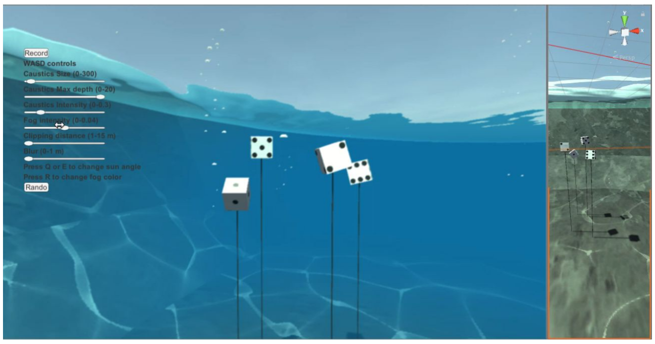

Fig. 6 A Unity simulator is used to generate synthetic training data

C. Simulator

Developed late last year, ARVP's simulator continues to play a critical role in testing the software system end-to-end. Built off the UWSim [4] project for marine research, testing in the simulator significantly sped up the development process for many projects, including the control system, various detection algorithms and the mission planning stack.

This year, the simulator dynamics were significantly improved in parallel to the development of the LQR dynamics model. With the addition of realistic drag and thrust coefficients, the robot's behaviour is much closer to reality than in previous versions.

D. Sonar

Three Teledyne TC4013 hydrophones are mounted in a triangular array where each hydrophones is less than 15 mm apart. Such a setup ensures that the time difference of arrival (TDOA) between each pair of hydrophones is less than one wave period of the signal. This simplifies the positioning algorithm and computational manipulations.

The PRUDAQ cape on the Beaglebone Black samples each hydrophone signal at 500 Ksps. This digital signal is captured by the programmable realtime units (PRUs) on Beaglebone Black. This data is sent to an AM335x ARM Cortex-A8 processor using shared memory, where all the processing is done.

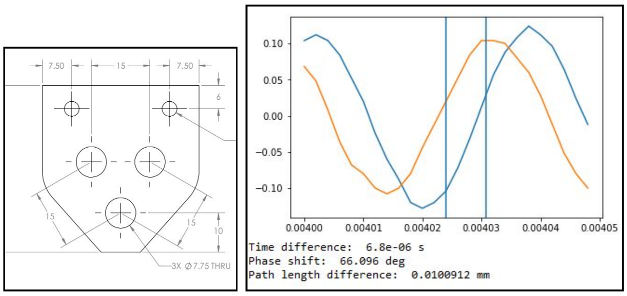

Fig. 7 Left: hydrophone mounting array Right: TODA plot

In order to determine the position of the sonar pinger relative to Auri, phase shift analysis uses an FFT between the three hydrophones. The phase shift is used to extrapolate the time difference. Given the times of arrival, an analytical solution of the multilateration equations is calculated using a simplified 2D version of the method developed by R. Bucher and D. Misra [5].

Once the relative position of the pinger has been calculated, the mission planner updates the yaw controller with a new heading, causing Auri to turn in the direction of the acoustic pinger.

E. Infrastructure

As the software team at ARVP grows, we strive to keep our software development process stable and efficient. Advancements in platform include containerized and continuously integrated builds, a pull request based workflow, and other small improvements such as build time reductions. These changes allow us to keep technical debt low and continue to increase code and feature throughput.

Central to our development cycle is our build and continuous integration system. Our first major change in this area was moving from over a dozen repositories to a single central repository. This now allows us to make cross-cutting changes and large code refactoring in a simple and atomic manner. Next, in order to ensure quick and correct turnaround on code changes, we moved our build and development environment to a containerized system. Moving our build to Docker greatly reduced the "works on my machine" phenomena and allowed easier integration with Travis CI, our cloud continuous integration provider. This new workflow has greatly reduced interference between teams and ensures a baseline of quality in order to contribute to the codebase.

Finally, we made many small "quality of life" improvements such as global automatically enforced code formatting, integration with ccache to speed up the development cycle, and seamless integration with CLion, an integrated development environment which provides intelligent source code navigation.

REFERENCES

[1] T. Fossen, Handbook of marine craft hydrodynamics and motion control =, 1st ed. Chichester, West Sussex: Wiley, 2011, p. 418.
[2] T. Fossen, Handbook of marine craft hydrodynamics and motion control =, 1st ed. Chichester, West Sussex: Wiley, 2011, p. 12.
[3] 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.
[4] S. K. Dhurandher, S. Misra, M. S. Obaidat, and S. Khairwal, “UWSim: A Simulator for Underwater Sensor Networks,” Simulation, vol. 84, no. 7, pp. 327–338, 2008.
[5] R. Buchner and D. Misra. A Synthesizable VHDL Model of the Exact Solution for Three-dimensional Hyperbolic Positioning System. VLSI Design, 2002, Vol. 15 (2), pp. 507-520.

APPENDIX A. COMPONENT SPECIFICATION

Component	Vendor	Model/Type	Specs	Cost (if new)
Buoyancy Control	-			
Frame	Custom			
Waterproof Housing	Custom			
Waterproof Connectors	Subconn	Power	25A/contact	500
Thrusters	Blue Robotics	T200		1250
Motor Control	Zubax	Myxa B		190
High Level Control	Custom LQR Control			
Actuators	Blue Robotics	HS-646WP		
Propellers	-			
Battery	Custom	Custom	14.8V, 94.72Wh	1400
Converter	Custom			
Regulator	Murata	UWE	100W	
CPU	Nvidia	Jetson TX2		
Internal Comm Network	-	CAN, I2C		
External Comm Interface	-	Ethernet		
Programming Language 1	C++			
Programming Language 2	Python			
Compass	-			
Inertial Measurement Unit (IMU)	Lord	Microstrain GX5_25		3500
Doppler Velocity Log (DVL)	Nortek	DVL 1000 - 300m		18000
Camera(s)	Go Pro	Hero 3+		350
Hydrophones	Teledyne	TC4013		4500
Manipulator	-			
Algorithms: vision	YOLO, OpenCV, Custom			0
Algorithms: acoustics	Custom			
Algorithms: localization and mapping	-			
Algorithms: autonomy	Custom			
Open source software	ROS, OpenCV, Eigen, YOLO			
Team size	47			
HW/SW expertise ratio	63/37			
Testing time: simulation	1000+ hrs			
Testing time: in-water	70 hrs			

APPENDIX B. COMMUNITY OUTREACH

The Autonomous Robotic Vehicle Project's Community Outreach Program has a very large and extensive output that is a result of years of effort. We currently undertake so many projects, it is hard to accurately summarize into this document. We will begin with ROS Edmonton, this is a monthly talk hosted by the club at Edmonton's startup incubator, Startup Edmonton. The event was acquired by ARVP to help create a hub for Edmonton's robotic industry, academia, and hobbyist communities, while demoing and giving tutorials on how to use the open source Robotic Operating System (ROS). This year, topics included: ROS 2 Today, Turtlebot 3 Burger demo and tutorial, a three part series on the ROS Navigation stack including building a simulator model and live demo model, and an AR Tag Tutorial. ROS Edmonton was also lucky enough to have ACAMP (acamp.ca) host a talk on their Alberta-based autonomous car project, in collaboration with the Government of Alberta. ARVP member and O'Reilly Media writer, Justin Francis, presented his article and tutorial on an Intro to using TensorFlow for machine learning and visualization. Last year, guests included Dr. Russ Greiner of the University of Alberta for a talk on Bayesian Machine Learning, and Dr. Giovanni Beltrame for Swarm Robotics from École Polytechnique de Montreal. ROS Edmonton updates can be found at rosedmonton.org.

ARVP is visible on Facebook, Instagram, Twitter, our website (arvp.org) and YouTube. The proudest of which is our Youtube channel filled with charismatic and entertaining videos designed to excite and dazzle viewers of the club. A fun recruitment video was made this year that was very successful, drawing excitement to the club, seeing larger than normal recruitment numbers in the fall. The team, outputs a monthly newsletter output to our public board in the UofA and to our Facebook page. The newsletter contains updates of club activities and recognizes individual members for their outstanding contributions to the club.

In addition to these systematic community outreach methods, the Autonomous Robotic Vehicle Project tries to present at as many individual community events as possible. Last year, the Edmonton community was invited to our unveiling for our new robot. The audience of over 50 guests got to see our new design for the first time, a live demonstration, our new competition video, and had a free BBQ. We were featured on Global News TV and CBC Radio to advertise the event and had 3 TV media stations in attendance for the event. Our next unveiling event is July 7th that will show our new robot. ARVP hosted a GitHub Tutorial session in November for University of Alberta students. ARVP members coached a high school robotics FTC Robotics team weekly. ARVP members then judged the FTC Robotics competition in January. ARVP hosted a team bonding Session in November to build props for the new competition obstacles with food served after. During the Christmas break, ARVP hosted a team bonding Potluck and movie night to watch a movie in the competition casino theme, the movie was 21. ARVP had booths at the following events, APEGA Science Olympics, APEGA Summit Awards, ECE First Year Night, Shell SELF Symposium 2018. ARVP hosted the First Annual ARVP Formal back in March, where members were encouraged to invite guests and a formal dinner and awards were given to members and an entertaining night of engineering challenges and games were had. ARVP also presented at two schools this year, Kildare Elementary on May 3rd, where a presentation and judging of Scratch projects was had, the second was to the Fort Vermillion School District on April 26th. Ft Vermillion is a remote agricultural community, an 8 hours drive North of Edmonton, nearing the Arctic Circle and Northwest Territories. ARVP was invited to give a presentation on the robotics and the clubs operations to the audience of grade 4 to 9 students, before their district wide robotics competition, including both VEX and EV3 robots. The presentation was intended to encourage kids in the remote community that an education at the University of Alberta is obtainable if they continue down this STEM focused path. ARVP was encouraged to engage with students and help out where possible during the robotics tournament. The local media was in attendance and put our appearance in the local paper, the Echo Pioneer. The school district hosted two ARVP members in a hotel, and paid for the expenses of attending. In addition, the superintendent of the district was so impressed, he made a $1,000 donation to the club.