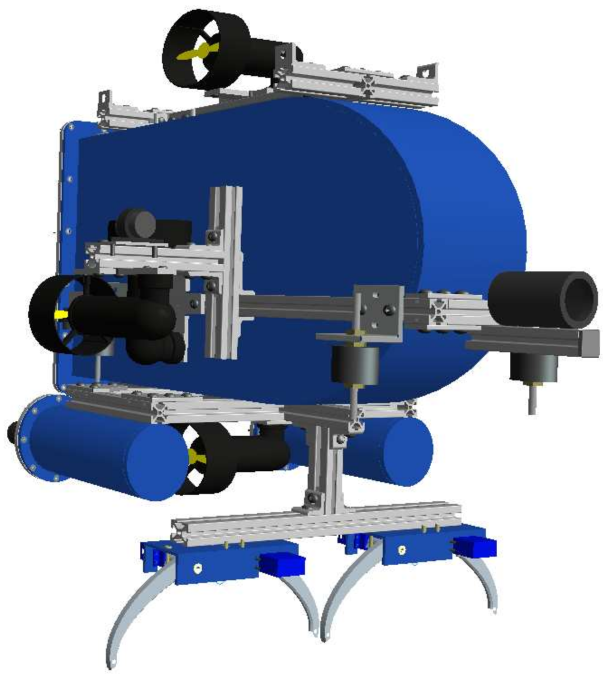
autonomous robotic vehicle project

UNIVERSITY OF ALBERTA



OVERVIEW OF BEARACUDA AUTONOMOUS UNDERWATER VEHICLE

Authors
Aassem Askari, Edmond Chen, Veselin Ganev
Mike Sumka, Chris Woloschuk

Abstract

The Autonomous Robot Vehicle Project (ARVP) started at the University of Alberta as a way to get students involved with the design, construction and promotion of autonomous robots and robotic technology. This will be the third year of attending the competition with their vehicle named *Bearacuda* and the final year of competition for *Bearacuda*. In preparation of this year's competition, the team has upgraded the old systems and built new ones to expand the capabilities of *Bearacuda*. A brand new electronics rack will allow for a more efficient use of space within the hull and to decrease the likelihood of wiring failure when the rack is placed inside of the hull. The passive sonar system features four hydrophones and a custom built PCB to amplify and filter the signal and then send time-difference-of-arrival data to the embedded computer which then calculates the location of an acoustic pinger. A new main board was designed to accommodate the brand new ColdFire Microcontroller that the team will be using this year. The power system was also improved by allowing each circuit board to regulate its own power down to an appropriate level in one step rather than multiple steps as in previous years. The software for *Bearacuda* was designed to be as modular as possible, using a custom built framework called "DisCo". This allows for easy communication between different, independent components such as the microcontroller and embedded computer which allows for more versatility within the system.

1 Introduction

The Autonomous Vehicle Project (ARVP) was founded in 1997 as a means for students at the University of Alberta to develop, apply and promote robotic technology. Members of the ARVP are interdisciplinary students from the Faculties of Engineering and Computing Science. Through the ARVP, these students gain the necessary skills and understanding of the different processes involved in the design, construction and testing of autonomous robots to prepare them for a possible career in the field of robotics.

From 1997 to 2006, the ARVP was involved exclusively with ground based autonomous vehicles to great success. Since then, it has focused its effort on underwater-based autonomous vehicles. This year, ARVP accepted the challenge put forth by the 13th International Autonomous Underwater Vehicle Competition, hosted by Association for Unmanned Vehicle Systems International (AUVSI) and Office of Naval Research (ONR). The challenge is to build a fully autonomous underwater vehicle (AUV) that can perform the following tasks:

- Following a predetermined path.
- Bumping into a series of coloured buoys in a predetermined pattern.
- Travelling over two horizontal pipes.
- Dropping a marker into a bin
- Launching a torpedo towards a stationary target.
- Locating an acoustic pinger.
- Grabbing an object and surfacing in a predetermined area.

This is the third year of competition for ARVPs autonomous robotic vehicle, named *Bearacuda*. This year, *Bearacuda* will be attempting to hit the buoys in the correct order, travelling over the two submerged PVC pipes, locating the acoustic underwater pinger and surfacing within the correct octagon.

2 Mechanical

In the first year of mechanical development a modular platform was developed that allowed the team to meet its primary objective of passing through the gate while still providing room for expansion to tackle future tasks. The 2008 and 2009 mechanical setup of the vehicle worked so well that the AUV was able to pass through the gate in qualifying with all control gains set to zero. However, at the 2009 competition, the platform ran into serious reliability issues due to the design of the internal electronics rack. Therefore, the focus of the platform this year was to redesign the rack and endplate so that more reliable connections could be made with minimum shifting of the internal wiring. A summary of the specifications of the AUV can be found in Table 1.1.

Table 1.1: **Summary of Specifications**

Size	0.75m x 0.6m x 0.70m
Weight	34 kg
Top speed	1.1 m/s
Running Time	29 min

2.1 Mechanical Design

The 6061-T6 aluminum hulls (Figure 1) were designed to allow for component positioning that increases the distance between the center of gravity (CG) and buoyancy (CB), increasing the restoring moment and static stability of the vehicle. Two separate battery hulls were designed as a safety feature to isolate the electronics in the main hull from a potential battery failure. The hulls were welded using GTAW welds and are sealed with o-rings squeezed between the flanges and endcap surfaces. During testing and competition the main hull seal is tested for integrity by creating a slight vacuum and monitoring internal pressure via an external vacuum gauge. To decrease assembly time and improve the assembly ergonomics the screws used in the main endcap were changed to a larger size and the nuts were switched from jam nuts to wing nuts reducing the number of tools required for assembly to 1. The hulls and endcaps were also given a blue rubber coating to improve corrosion resistance and vehicle aesthetics.

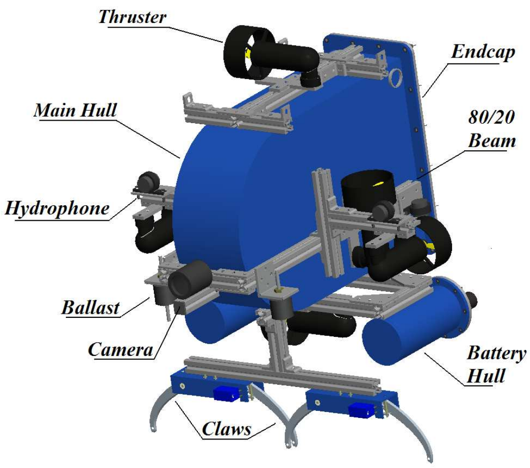


Figure 1: Bearacuda

80/20 beams were chosen for the frame and provide a means for the mounting of external components. The benefit of using 80/20 beams is that they allow for components to be easily moved around to statically trim the vehicle as well as allowing for the addition of future modular components. The 80/20 beams, which are connected to the hull via welded aluminum L-brackets, run along the top, sides, and bottom of the vehicle (Figure 1).

To statically trim the vehicle a variable ballast mechanism (Figure 1) was created. Incremental masses are placed on either side of the CG. Sliding the variable ballasts forwards or backwards allows for fine adjustment of pitch and heel. Adding small increments of mass to all four ballasts allows the vehicle to be adjusted very close to the 0.5% buoyancy mark reducing the amount of load on the vertical thrusters. The variable ballast mechanism can be adjusted on the fly in the test pool to quickly balance the vehicle before running the thrusters.

Thrusters were chosen for the method of propulsion because of their high availability and ease of control compared to other methods such as fish-like fins, or servo-actuated control surfaces. Six Seabotix BTD150 thrusters are arranged around the major axes of the vehicle to provide 5 degrees of freedom including all three translations (heave, sway and surge) as well as two rotations (yaw and heel). Sway was chosen over pitch control to aide in fine positioning over the marker bins and PVC counselor.

An array of 4 SQ26-11 hydrophones were mounted to the 80/20 beam frame using custom brackets. Two hydrophones were mounted on the thruster arms and face forwards and other two were mounted on the rear and face downwards. The goal of this hydrophone placement was to maximize the distance between the hydrophones. The angle of the brackets can be adjusted to change the hydrophone angle if required.

The cameras are mounted externally to the 80/20 beams to allow *Bearacuda* to have vision both forward and down for the tasks.

A set of two modular claw units were designed to grab the PVC Counselor and restrain it in the required 3 degrees of freedom (Figure 2). The task was analyzed and several key design features were implemented in order to maximize the probability of successfully grabbing the briefcase. The claws are actively controlled by the field commander via 4 Traxxas 2075 waterproofed servos mounted to each claw. By arranging the servos on opposite sides of the claw mounts, only 1 pulse width modulation channel was required to control the 4 servos cutting down on cables and controller complexity. To trigger the claw, a water-proofed momentary bump switch was placed under the main housing of the claw mechanism. A trigger plate was then placed above this switch to give it a wider area of sensitivity. The bump switches on each claw unit relays information back to the onboard embedded computer to help assess whether or not the treasure has been grabbed correctly and to prevent the claws from misfiring in other stages of the competition. The modularity of the claws will allow them to be easily adapted to future treasure geometries. The claws also serve the dual purpose of holding and releasing markers into the bins.

The electrical components are housed within the main hull by means of an electronics rack). The modular design of the electronics rack allows easy removal of the electronics when they need to be tested on the workbench and provides a substantial amount of room for future components. This year, the rack was completely redesigned from a single plate to triple plate design. The plate is also firmly attached to the backplate, which allows the entire rack to be easily slid into and out of the hull with minimum disturbance to the cables. This was a problem in past years as the wires could easily snag when the rack was being moved, leading to serious dependability issues.

A solid model of the vehicle and all of its submodules was created in PRO/Engineer to ensure proper design space was allocated for all current and future internal components. A complete engineering drawings package including assembly guide was created from the solid model and provided a means of communication between the ARVP and the University of Alberta machine shop during the manufacturing phase of the project.

2.2 Mechanical Validations

Several calculations were performed to validate the mechanical design before manufacturing was started and are summarized in Table 1.2 below. Validation ensured that all the critical requirements were met and prevented costly iteration at the manufacturing stage.

The 2010 mechanical AUV improvements extend upon past achievements and provide a platform that meets all the requirements of the ARVP’s primary objective. The modularity of the design will allow for the easy addition of components to complete the team's additional objectives in future years. This year, the main objective was to increase the reliability of the platform as that had been a struggle with previous years.

Table 1.2: Summary of Mechanical Validation

Calculation	Objective	Result	Conclusion
Buoyancy	% buoyancy >= 0.5%	0.5 – 5% buoyancy with variable ballast mechanism	Meets requirement
Max Velocity	Between 0.5 – 1.5m/s	Approximately 0.8m/s	Meets requirement
Depth	Reach depth of 16ft without yielding	Factor of safety = 10	Exceeds requirement
O-Ring Seal	Maintain seal up to 16ft	Factor of safety = 190	Exceeds requirement

3 Electrical

For the third generation of *Bearacuda*, the electrical team focused on the implementation of a sonar system, implementation of a new microcontroller and redesigned main board. As well, voltage regulation was broken up from one large central voltage regulator to multiple smaller regulators.

3.1 Main Board

For the initial electrical design, a Printed Circuit Board (PCB) was custom designed and fabricated to integrate all of the various electrical subsystems (Figure 2). The Main Board can be divided into several different areas: the micro-controller, communication conversion, voltage regulation, and connection ports. For the new board, only one micro-controller was used for the sensors.

The Main Board offers voltage regulation and Bucking down from 12V to 5V for the devices, which use lower voltages. It has TTL to RS232 and TTL to USB conversion. LEDs are used to indicate USB activity and motor controller power status. A system of two LEDs and a physical switch prevents the input power from being applied backwards.

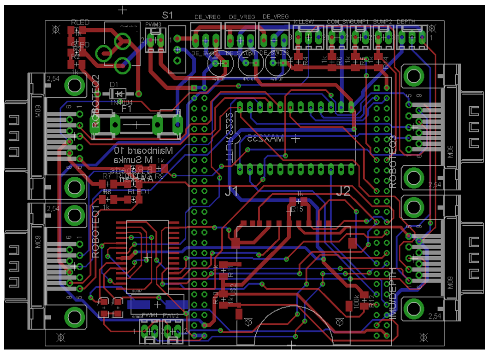


Figure 2: The Main Board

3.2 Actuators

The actuators used for this year's competitive design were the Roboteq AX500s (Figure 3). These motor-controllers drive the thrusters with a 16 kHz pulse width modulation (PWM) and are capable of sustaining a current of up to 15A. They also include features such as battery voltage detection, set current limiting, and battery regeneration.

Figure 3: Roboteq AX500

3.3 Sensors

In order to achieve a proper bearing, the electrical team chose Ocean Server's OS5500 (Figure 4), a combination inertial measurement unit (IMU) and digital compass. This device also allows the incorporation of a pressure transducer as a means of obtain the depth. The OS5500 has connectors for both USB and RS232 communication and offers an ASCII interface to allow programming with the micro-controllers.

Bearacuda is capable of taking internal temperature values from multiple sources. The temperature can be measured from the motor controllers and the IMU.

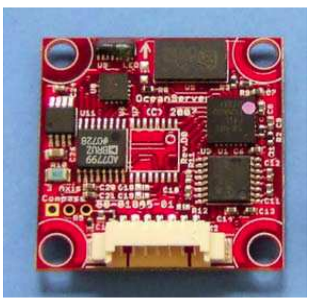


Figure 4: OS5500

3.4 Sonar System

The passive sonar system (Figure 5) uses four SQ26R1 hydrophones with built in pre-amplifier to 'hear' the pinger. The captured signal from the pinger first passes through a 10 V/V pre-amplifier to boost the signal as it was determined that the built in pre-amplifier did not supply a large enough gain to process the signal. After the pre-amplifier, the signal is sent through a programmable second-order band pass filter. The filters isolate the desired frequency and dramatically remove any noise that has entered the signal. Once the signal has been filtered, it is sent through a variable amplifier to prepare the signal for sampling by a dsPIC33FJ16GS502 microcontroller from Microchip Technology. The microcontroller samples each of the four signals from the hydrophones and when the signal exceeds a pre-set threshold, an interrupt is triggered and time-difference-of-arrival (TDOA) data is captured. The TDOA data is then sent via a UART-to-USB converter to the embedded computer.

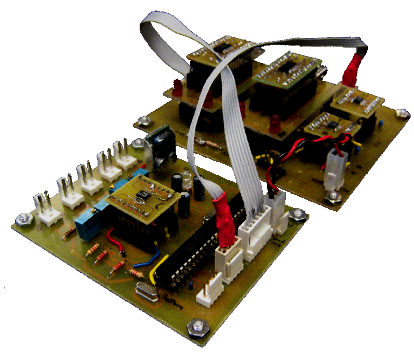


Figure 5: Passive Sonar System

3.5 Power System

The electrical power system is comprised of two identical lithium polymer (LiPo) batteries, each encased in a separate battery hull. The LiPos have five cells and are rated for 5000 mAh at 18.5V with a maximum continuous current rating of 125A. One LiPo is used to power the motor controllers and power the thrusters, while the other is used to supply the rest of the electronics. The team decided to break up the onboard voltage regulation to allow for each system to regulate its own power. This was done because in the past, all power for the electronics would be first regulated to 12 V, and then forwarded to each system. Each system often had to then regulate the voltage further down to either 3.3 V or 5 V. Switch mode voltage regulators from Dimension Engineering were used to regulate the voltage for each individual system down to the desired voltages.

3.6 Connectors

Subconn wet mateable electrical connectors were used and the cable splices were waterproofed using Scotchcast insulating resin. All internal cabling was made in our lab.

4 Software

The software for *Bearacuda* was architected to be as modular as possible. It utilizes a framework called "DisCo" which was developed in the department of Computing Science at the University of Alberta, with the goal of providing a unified communication framework between components in robotics systems. The software consists of a number of independent components that communicate with each other using small packets of information. These components are divided between micro-controller (MC), embedded computer (EC) and a laptop/development computer. An important benefit of the design is that two components can easily communicate with each other regardless of whether they are running on the same device or on different devices that are networked together. Figure 6 shows a general overview of the more important components and how they interact with each other.

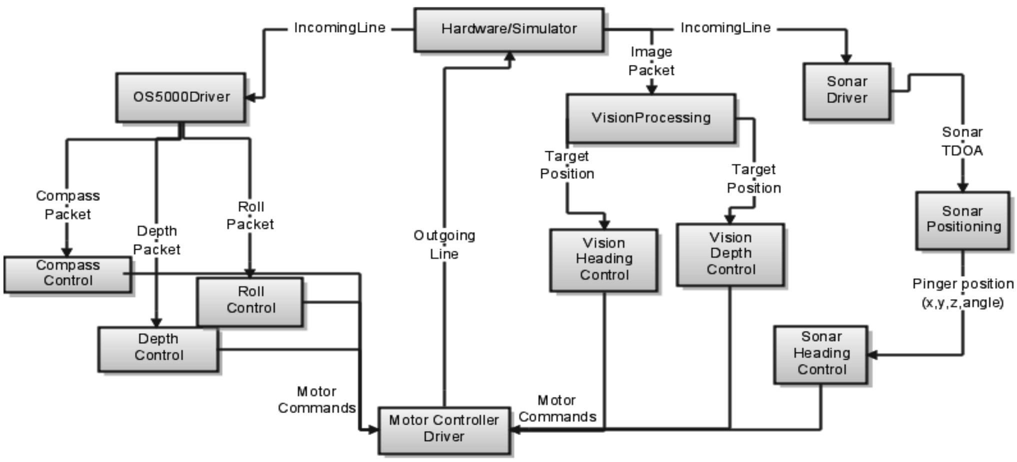


Figure 6: High level drawing of relationship between components

4.1 Drivers

The software system contains a collection of independent pluggable driver components that are responsible for interfacing with our hardware devices. The drivers are designed to be abstract so that the rest of the system can be independent of their implementation. Each of the drivers is a component, responsible for initializing a device, communicating with it using the appropriate protocol and interfacing that device with the higher level components.

The low level drivers consist of several components that help the higher level drivers communicate with the actual hardware such as an RS-232 serial driver component, a GPIO driver component and a PWM driver component. Additionally, there are two other low level drivers used during development/testing: simulated drivers which can be used for interfacing the higher level drivers with our simulator and a "log driver" which allows for "replaying" a previous run of *Bearacuda* for debugging purposes.

The high level drivers are responsible for decoding data received from the devices as well as encoding data that needs to be sent to them. An important feature is that these drivers do not care whether they are talking to the low level drivers for the real devices, or to the drivers for simulated hardware.

4.2 Navigation

The software team developed a generic PID controller in order to be able make decisions about how to control the thrusters based on current mission and sensor readings. The controller component itself is independent of the item being actually controlled, allowing it to control different items and making it easier to modify. The higher level components can use the controllers by requesting that the target of controller is changed as well as starting/stopping individual controllers and changing gains and other parameters.

Heading and depth control are done by monitoring the current values (from the compass and the depth sensor) and feeding them to the respective controller instances which will produce appropriate efforts that in turn will determine what power should be applied to each of the thrusters.

When performing a vision oriented task the vision component will produce a location for the object being tracked and that will be sent to the appropriate controller (which knows the desired value for the location of the object). Similarly, for the sonar mission, the Sonar component, after estimating the relative heading of the pinger, will update the target of the heading controller.

4.3 Mission Planning

The Mission Planner component is responsible for running and controlling the various missions that *Bearacuda* will perform. A mission has a completion condition and/or a timeout. Whenever a completion condition has been reached then the current mission will terminate and the Field Commander will move onto the next mission. If however a mission is taking too long and reaches its timeout then the Mission Planner will terminate it and move on.

This is done so that *Bearacuda* does not spend too much time on a single mission without getting a chance to perform the other missions.

Each mission has various parameters which are all stored in XML files that are automatically reloaded when changed. This allows us to quickly update the parameters during development/testing.

4.4 Sonar

To locate the pinger, multilateration is applied to the four TDOAs obtained from the sonar board to estimate the position of the pinger relative to the center of *Bearacuda*. By searching for the location of the pinger in three dimensions and using four hydrophones an analytical solution of the multilateration equations can be found [1].

Since the sonar board processes the interrupts serially, it is possible that one signal may arrive before the previous signal has been processed, resulting in an incorrect distance being calculated. The sonar component is able to detect this problem by checking if any of the TDOAs are too close to each other and it will adjust times that are too close in order to reduce the average error in these cases. A simulation program was developed to show how various amounts of measurement error can affect the estimated position of the pinger.

4.5 Vision

The vision system is implemented using the OpenCV library. A frame grabber component captures frames from the camera continuously and sends them to the image processing component which applies the appropriate filters for the current mission and then makes the results available to the vision controllers as well as the mission planner.

For locating the buoy, a red filter was applied so that only the "red enough" pixels are left and then using blob detection, the AUV locates the blob that most likely corresponds to the buoy and aims for the center of that blob.

Similarly, for the hedge mission, a green filter is applied first in order to isolate the hedge from the rest of the frame. Then the biggest blob is found and the AUV aims for its center.

For the bins, the system attempts to recognize the white border around the bins by looking for pixels that are close to white and then again finding the biggest blob. Currently we are unable to perform shape recognition.

A major difficulty with the color-based image processing that we use is that the colors as seen by Bearacuda can differ significantly at different times of the day. Currently, we address this by making it easy to quickly change our thresholds and immediately see the effectiveness of the current thresholds through a graphical interface.

4.6 Graphical Display

When testing we visualize data coming from the sensors, including live video, annotated with image processing information. (Figure 7) This allows us to easily see what *Bearacuda* is doing at the moment. It is also possible to replay a recording of previously collected data.

4.7 Remote Control

A remote control interface was implemented to allow us to control *Bearacuda* manually when testing/troubleshooting. It is able to issue a target value for the depth controller and it can also control the turning of *Bearacuda* by either supplying a direct turning effort command or by adjusting the target of the heading controller which in turn will make *Bearacuda* turn in the required direction.

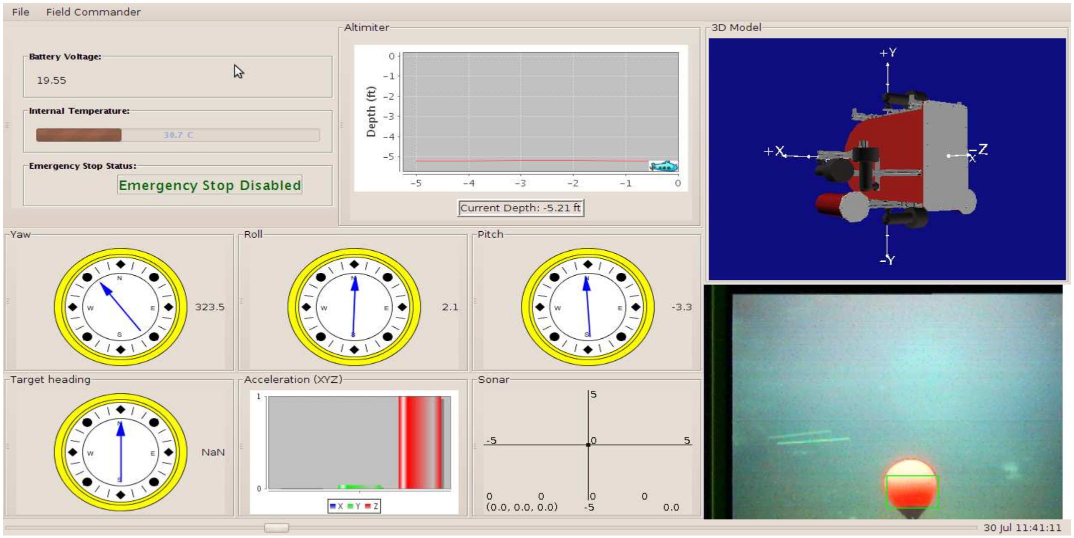


Figure 7: Graphical User Interface

4.8 Simulator

In order to help us with the development process, we have developed a simulator that allows us to test parts of our software. The simulator makes use of the Physics Abstraction Layer (PAL) library [2] in order to perform simple physics simulations. For visualization it uses Java3D to render the AUV and the other simulated objects. Note that currently the simulator does not attempt to accurately model the real AUV or the environment, so it cannot be used for tuning the controllers for the real AUV. However, it can be used for testing the correctness of most of our software and for experimenting with different mission plans.

5 Acknowledgments

We would like to thank all of your sponsors, past and present, for both their generosity and support: Northern Underwater Systems, Shell, Seabotix, Sun Microcontrollers, Subconn, Microchip, Alberta Printed Circuits, and Active Electronics. Special thanks also goes out to W. A. Moussa P.Eng, Ph.D., Curt Stout, P.Eng, Loren Wyard-Scott, P.Eng, Martin Jagersand, Ph.D., Connor Harper, Linda Kelly, Jamie Reid, Michael Blinzer and the RLAI Laboratory, Department of Computing Science, University of Alberta.

6 2009-2010 Team Members

These are the team members of ARVP for the 2009-2010 year.

Team Leader: Brodi Roberts
Platform Team Leader: Rory Dawson
Platform Team: Aassem Askari, Michael Sumka, Eugene Hsung
Sonar Team Leader: Chris Woloschuk
Sonar Team: Uri Chin, Adam Crookshanks, James Currie, Allen Lee
Software Team Leader: Veselin Ganev
Software Team: Dale McConachie
COPPER Team Leader: Edmond Chen

7 References

[1] R. Bucher, D. Misra, "A Synthesizable VHDL Model of the Exact Solution for Three-dimensional Hyperbolic Positioning System," *VLSI Design*, Vol. 15 (2), pp. 507–520, 2002.

[2] http://www.adrianboeing.com/pal/index.html