Lawrence Technological University

H_2.2Bot

IGVC 2010 Autonomous Vehicle

Team Members

Adrian Snyder, Alex Campbell, Matt Lanting, Stephanie Shevenock, Ben Bargman, Jacob Kallie, and Mitchell Dumond

Contributors

Paul Downen, Ze Cheng, Tasha Watkins, Amy Tsang, Samantha Burgess, Phil Lucas, Matt Greer, and Sam Sinnamon

Faculty Advisor Statement[1]

I, Dr. Robert Fletcher of the Department of Mechanical Engineering at Lawrence Technological University, certify that the design and development on H_2.2Bot has been significant and qualifies for course credit in senior design.

Dr. Robert Fletcher (fletcher@ltu.edu, 248-204-2525)

Date

---

[1] Co-advisor: Dr. C.J. Chung

1) Introduction

Lawrence Technological University is proud to present the H_2.2Bot for the 2010 Intelligent Ground Vehicle Competition (IGVC). This is the product of a multidisciplinary group of undergraduate and graduate students from the math and computer science, electrical engineering, and mechanical engineering departments. The vehicle is named for the Ballard 1.2 kW Proton Exchange Membrane (PEM) fuel cell that serves as its primary energy source. This is the third fuel cell robot from LTU, but this platform introduces two new types of energy storage used in parallel with the fuel cell. Not only was the H_2.2Bot designed to compete in the IGVC, but it also serves as a research platform for autonomous energy management.

2) Design Process

2.1 Team Structure

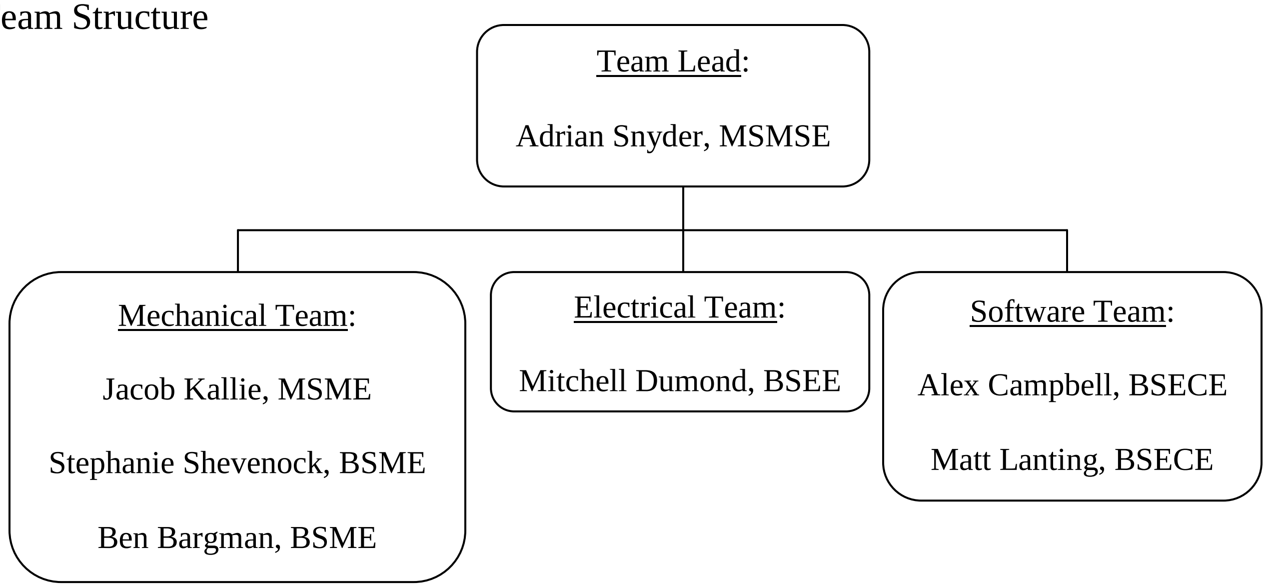


Figure 1: Team Structure

Effective team communication was a primary focus. Weekly team meetings were conducted with all team members and our team advisor to ensure that everyone had a clear idea of their tasks and how they fit into the project as a whole. Subgroup meetings were held at least once a week and were often conducted as work/lab sessions.

2.2 Project Planning

The H_2.2Bot project began in the spring of 2009 as a research platform for autonomous energy management and as an applied project for the extensive amount of PEM fuel cell testing Lawrence Tech has performed since 2006. It was decided at the beginning of the project to use the IGVC rules and regulations as a design benchmark. Lawrence Tech's long history of being involved in the IGVC competition provided numerous resources and contacts in designing the vehicle.

Along with complying with the rules and regulation of IGVC, we wanted to make sure that we were fulfilling our obligations to those funding the energy management research. This meant incorporating lithium ion batteries and ultracapacitors as a means of energy storage and buffering to work in conjunction with the fuel cell. Also, an array of current, voltage, and temperature sensors have been implemented as a means to collect data from the power generation, storage and distribution devices.

As well as having some innovative means of powering the robot, many of the design innovations focused on ease of access, serviceability, and tuning. Components that require regular change outs were kept in easy to access locations with quick connects. Interfacing with devices was made as simple as possible by locating connectors in areas easy to reach. Mounting points for sensors were designed with flexibility in mind so that they might be adjusted quickly and easily to improve performance.

3) Innovations

With the H_2.2Bot, we developed several innovative methods for increasing the robot's functionality, ease of use, and ability to maintenance. We fabricated an adjustable LIDAR mount (figure 2), that enables us to raise and lower or increase and decrease the angel of the plane that the LIDAR reads. We also greatly reduced maintenance time by incorporating quick connect/disconnects for charging the ultracapacitor bank and replacing the hydrogen storage tanks (figure 3). In addition, we built a mounting frame around the batteries that enabled us to easily disconnect and replace a set of four spent batteries in about a minute (figure 3). For maintenance, we built mounting points onto the chassis for a set of modified jack stands that allow us to securely lift the robot even on uneven terrain.

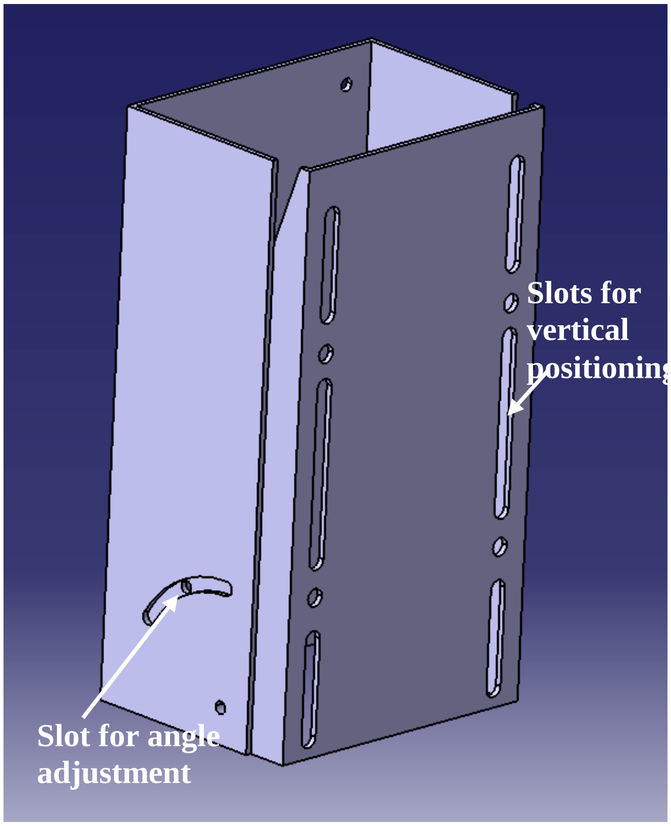


Figure 2: Adjustable LIDAR mount

4) Hardware Design

4.1 Mechanical Structure and Components

4.1.1 Chassis and Packaging

Our main goal in competing is to showcase the robot's ability to navigate; therefore, much care was taken to ensure that the chassis would not detract from the H_2.2Bot's ability to perform. To do this, we needed to reduce overall vehicle weight, ensure vehicle stability, and provide adequate and functional mounting solutions for each component.

To reduce weight, our chassis was constructed from 1.5" hollow aluminum square stock and aluminum sheet metal. The H_2.2Bot is quite large (59.5" long by 30.125" wide by 70" tall) and incorporates many, very heavy components, meaning that weight could potentially be a concern. The frame weighs about 48lbs, but a typical steel alloy with the same volume of material would have weighed close to 140lbs. Aluminum is also advantageous because it is non-ferrous and would not interfere with the compass.

To increase stability, the H_2.2Bot was designed to be low and wide, situating a majority of the weight at the base of the chassis and spreading it across a large area to avoid tipping. The H_2.2Bot has several large and heavy components, such as the fuel cell, ultracapacitor bank, and hydrogen tanks, which could not be arranged vertically without greatly affecting the center of gravity.

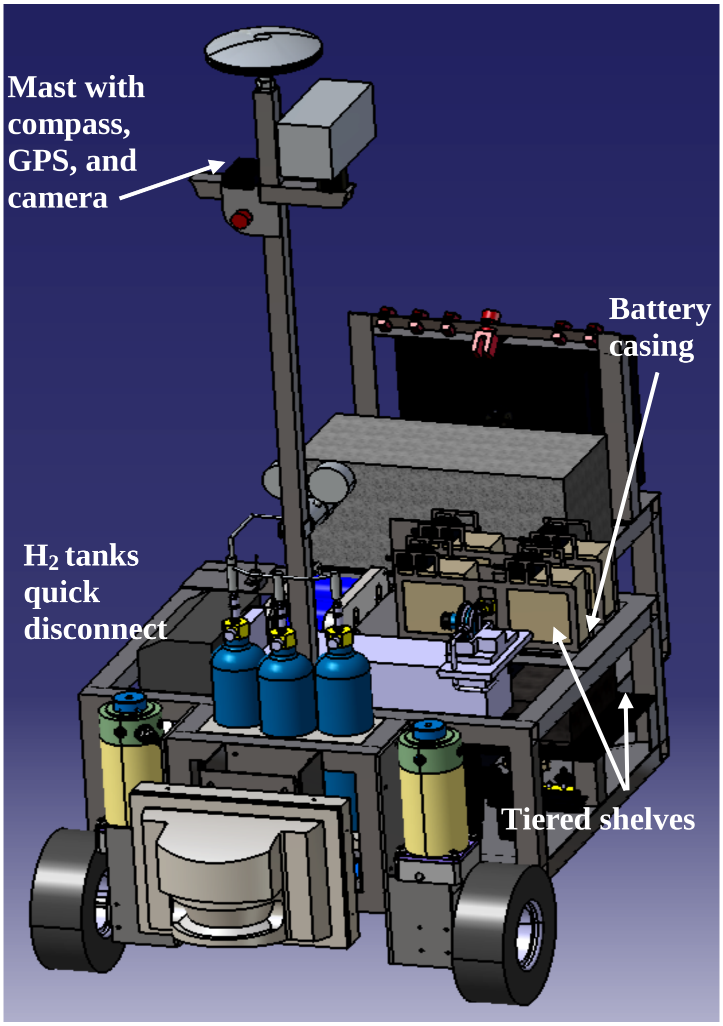


Figure 3: The H_2.2Bot right side view

To help with packaging efficiency and equipment functionality, we developed a tiered shelving system for some of the lighter components and built a mast for the navigation equipment (figure 3). With the shelves, we left the batteries on top for ease of access and placed components that would not need regular replacement on the lower shelves. This greatly increased our packaging efficiency, leaving room to incorporate additional data acquisition equipment. The mast is a 57" tall hollow aluminum square tube that slides into the chassis frame (figure 3), providing a good field of view for the camera and mounting for the GPS receiver. It also keeps the compass further from EMF interference from the electrical systems on the main chassis.

4.1.2 Drivetrain

The H_2.2Bot was designed with two 10” differentially driven wheels in the front to help with maneuverability, while the two casters in the back aid in stability. Most of the H_2.2Bot’s weight is over the front wheels, providing more traction. The H_2.2Bot was designed with gearboxes positioned perpendicular to the ground, increasing the amount of clearance for the bottom of the chassis. The motor and gearbox assemblies are custom units from Magmotor and provide ample torque and speed.

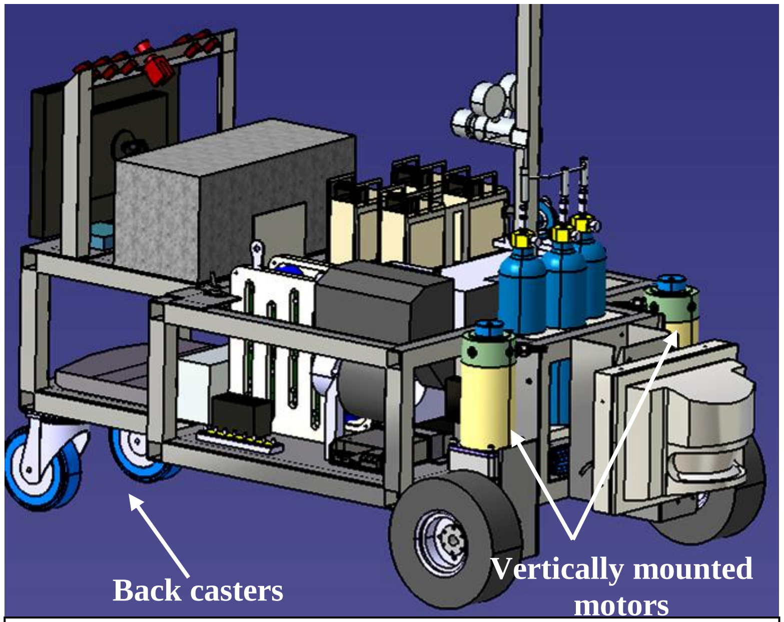


Figure 4: The H_2.2Bot left side view

4.1.3 Body

The body is constructed of fiberglass and is designed to protect the H_2.2Bot’s components from the elements while still allowing easy accessibility. Fans were added to make sure that internal components are kept cool during operation. The weather proof touch screen that provides the user interface with the H_2.2Bot was kept accessible even with the body on.

4.2 Electronics

4.2.1 Navigation Sensors

The H2.2Bot uses 4 primary navigation sensors: a Novatel ProPak-V3 GPS, a PNI Prime 3-axis digital compass, a Sick LMS221 LIDAR, and a Panasonic PV-GS500 Palmcorder. The GPS and compass are used for waypoint navigation, and the camera and LIDAR are used for obstacle avoidance and lane following.

4.2.2 Computer and Data Acquisition

The H2.2Bot control software runs on a rugged Panasonic Toughbook 52 computer mounted to a 12V supply bracket commonly used in the field.. By using a single laptop and mounting bracket, we can easily remove the unit to update the control software while maintaining a common development target.

For data acquisition, we have a National Instruments cRio chassis and controller that provides access to roughly 30 voltage, current, and other sensors. This allows us to preprocess the energy consumption of the robot with higher accuracy and make that data available to the decision-making processes in the main control algorithms.

4.2.3 Motor Control

Motor control is handled by a Roboteq AX2860 80A/channel controller. The AX2860 provides velocity feedback measurements through quadrature optical encoders. Additionally, the E-Stop is wired to the controller's power control. By pressing the E-stop we ground the power to the controller bringing the motor to a stop.

4.2.4 Communication Structure

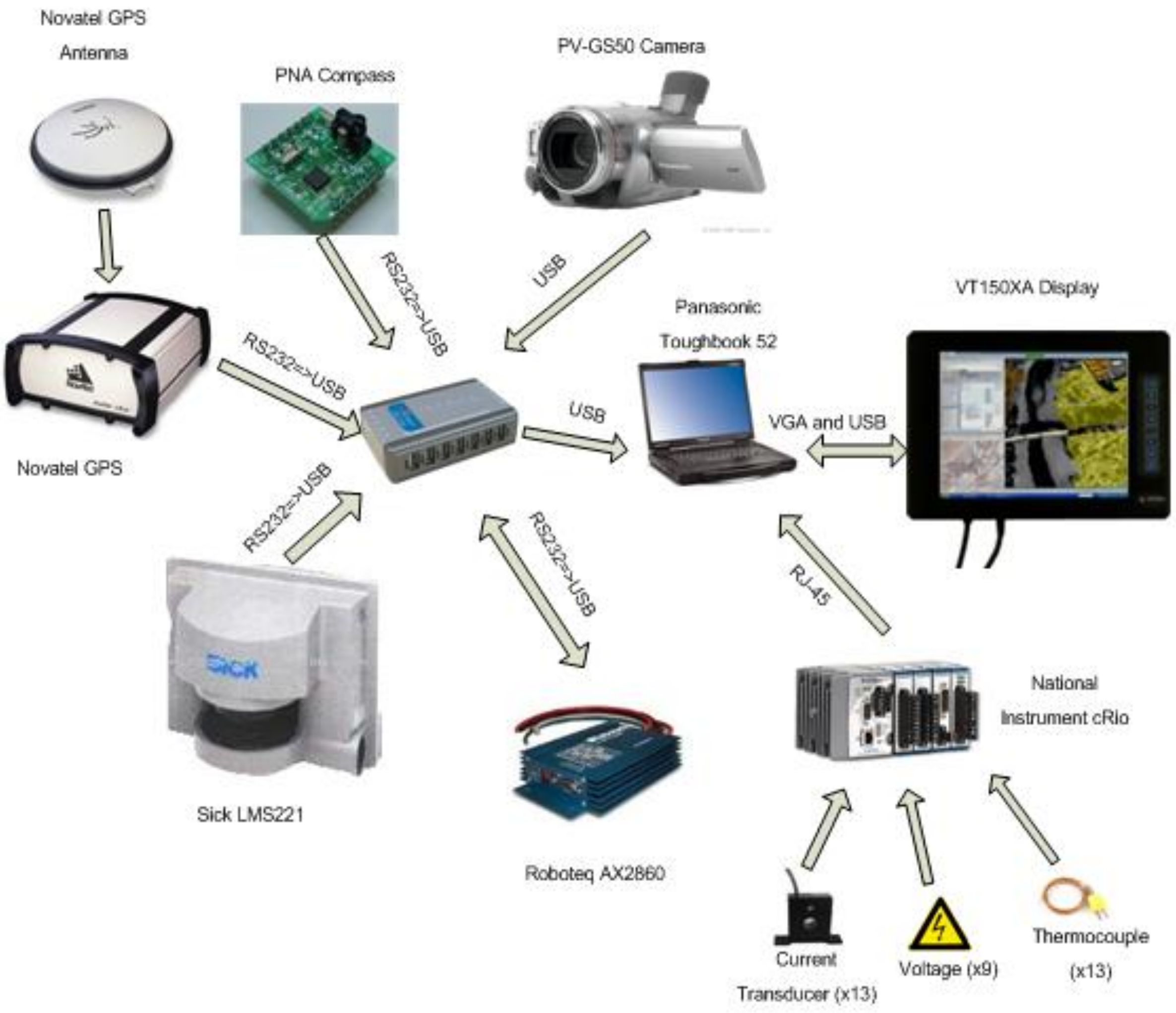


Figure 5: Communication Architecture

The image above shows how the electronics are connected to the computer. The LIDAR, motor controller, and Compass are connected to the USB hub via RS232 to USB adapters. The GPS and Camera are connected directly via USB. The cRIO is connected to the laptop via a RJ-45 cable. The display is connected to the laptop via VGA and USB cable. USB is used for the display's touch screen to let you input commands though the monitor which is bright enough to be seen in direct sunlight.

4.3 Power Generation, Storage, and Distribution

4.3.1 Fuel Cell

The Ballard Nexa is a 1.2kW, air-cooled, proton exchange membrane (PEM), hydrogen and oxygen fueled fuel cell. The byproducts of the hydrogen and oxygen reaction are electricity, heat, and pure water. The fuel cell is load following with a peak Current of 40amps. Peak voltage at open circuit is 41V and 29V at peak current. The unit has an on board roots blower for oxygen. The max inlet pressure for hydrogen is self regulated at 5psi. Nexa has a self humidifying function that will humidify the hydrogen before entering the 47 cell stack. The safety systems onboard Nexa ensure operator safty and prevent equipment damage. Fuel cell operating parameters are continuously monitored to ensure they stay in the desired range. A cell voltage checker monitors each cell and shuts down Nexa if there is a failure. A hydrogen leak detector is implemented detecting excessive hydrogen purge of an external fuel leak in the stack shutting down the system.

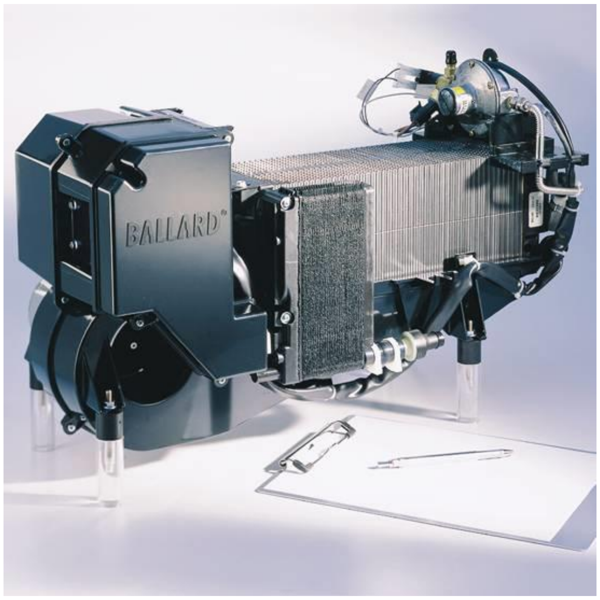


Figure 6: Nexa PEM Fuel Cell

1,200 watt Nexa Fuel Cell Power Module from Ballard Power Systems.

Physical size (L x W x H): 22" x 10" x 13"
Weight: 27 lbs.

4.3.2 Batteries

Originally with the H2.2Bot, we were using lead acid batteries. However, as we realized that most military UGV’s used Li-ion batteries, we decided to replace the lead acids with Li-ions in order to keep with the main focus of our project (military power management). We also decided to switch to Li-ions because they are lighter and smaller than lead acids and have virtually no memory. We chose Bren-Tronic’s milspec BB-2590/U, part number BT-70791A, NSN 6140-01-490-4316 Li-ions since these are typically used for military applications. We use four batteries in 14.4amp, 12V mode with the fuel cell and the ultracapacitors to provide a constant onboard energy source and help with initial system start up. When the fuel cell is able to handle the load from the robot after the batteries have been partially discharged, it charges the batteries for later use.

4.3.3 Ultracapacitors

The ultracapacitors are part of Maxwell Technologies BOOSTCAP line. Each cap is rated at 2.7 volts and 2000 Farad. 16 of these units in series hold just over 116 kJ at 43.2v and 125 Farad. The capacitors handle power transients from the H_2.2Bot’s electrical components, particularly the drive motors, and increase reliability and lifetime of the fuel cell and batteries.

4.3.4 Power Distribution

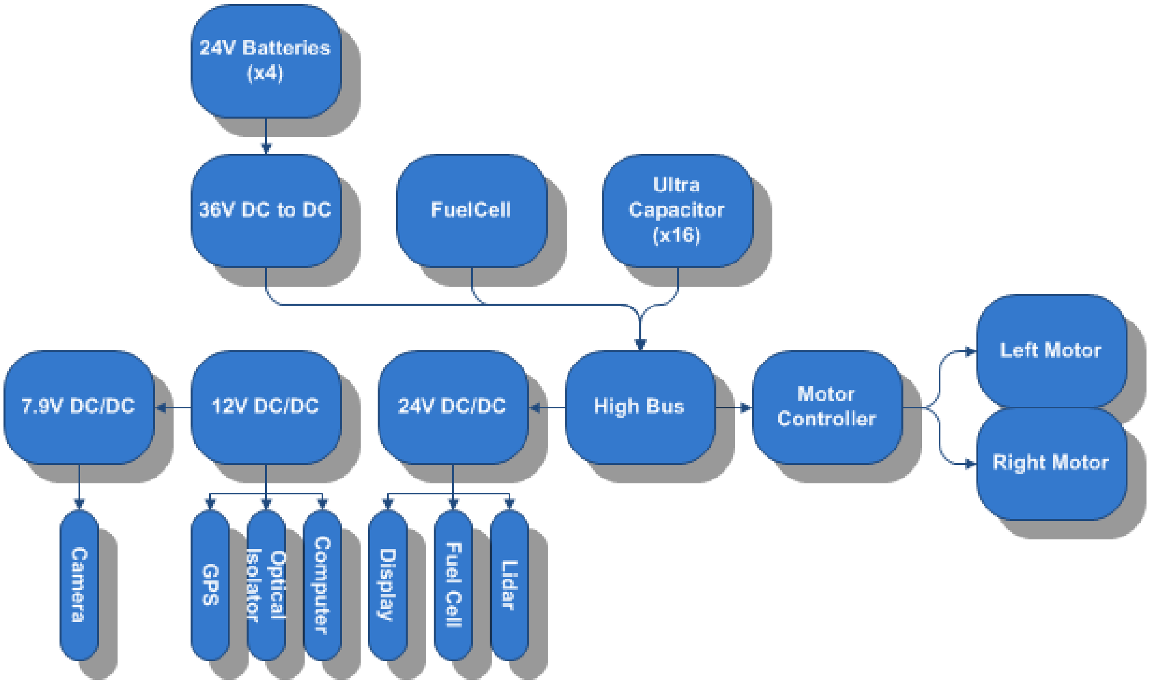


Figure 7: Power Distribution Diagram

The fuel cell, batteries, and ultracapacitors are wired together in parallel. A variable output DC/DC has been placed between the batteries and high bus to dictate the minimum voltage of the high bus, and the high bus voltage at which the batteries will deliver current. The fuel cell and ultracapacitors are load following devices and will naturally match one another's voltage level as they deliver power to the system. This allows us to design the system so that these devices do not need any power conditioning. The motor controller has a wide input range and feeds directly off of the variable high bus voltage. Auxiliary electronics require a 24v, 12v, or 7.9v source. These converters are placed in a waterfall type sequence and provide power to low power components.

5) Software Design

5.1 Software Strategy

The MCS department at Lawrence Tech has decided to begin working toward a common set of reusable code and algorithms that will help future teams in pushing the boundaries of autonomous navigation. The H_2.2Bot team has developed several libraries and interfaces that will minimize the effort required to get future platforms up and running, and allow the team to focus on control logic rather than device integration.

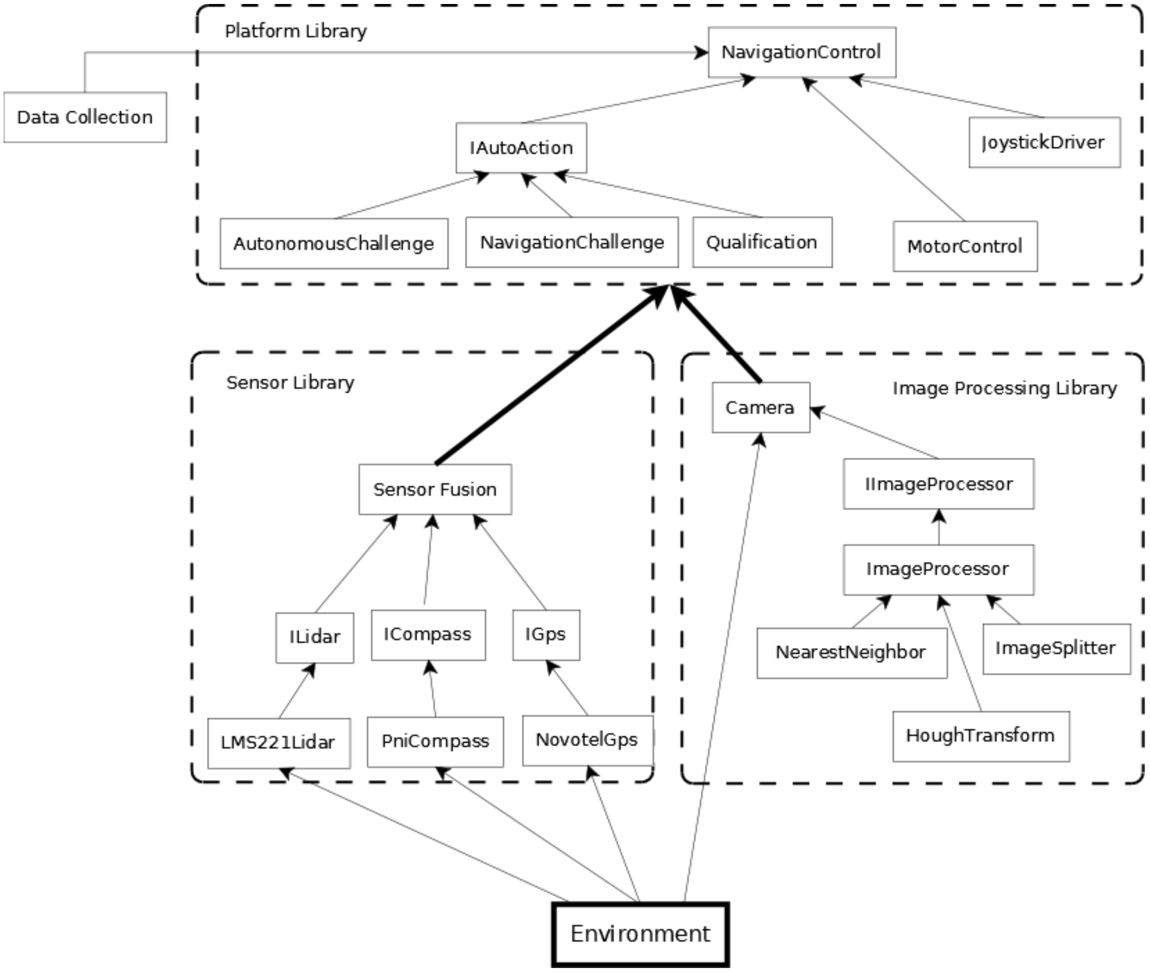


Figure 8: Software Architecture Diagram

The H2.2Bot control software is written in C# in order to take advantage of native driver code for data acquisition solutions.

The sensors are used to determine the unit’s current position, orientation, and environment. All of the sensors (as well as the Roboteq AX3550 motor controller) communicate over RS232. In addition, each of the sensors implements a serial port location algorithm, which automatically retrieves the serial port object based on known responses to inputs. This ensures full plug and play compatibility between bench-top and full system testing.

The Image Processing library consists of a collection of filters and tools for processing images. These tools were created using a combination of the AForge.NET Framework, and libraries created in-house by the software team. The camera communicates with the system via a USB connection and is used to provide accurate representations of lanes and boundaries drawn on the ground. The image processing library is separated from the sensors because computer vision algorithms are much more varied and constantly being improved. The Platform library is the most specific to the hardware, encompassing the motors, data acquisition, and serial communications with power devices (such as the Nexa fuel cell).

By decoupling the Sensor and Image Processing libraries from the Platform libraries we are able to change the hardware considerably without having to rewrite more than the specific device code and comport auto-location. This allows future teams to add additional capabilities to the sensor library, such as Kalman filtering, without altering the sensor interfaces themselves.

5.2 Autonomous Challenge Details

The H_2.2Bot uses one of two algorithms for obstacle avoidance depending on whether it is in the driving autonomously between lanes, or following a series of waypoints. If lane detection is necessary, the robot combines images obtained from the camera with the obstacle data obtained from the LIDAR and applies a series of image filters to find a path to the most distant point it can “see”.

Lanes are detected with the camera by first applying an algorithm that searches for the brightest pixels in an image. The program then creates a binary image where the bright pixels are white and all other pixels are black. Noisy pixels are filtered out with a combination of blob detection and erosion techniques so that only the lines remain. The image is then split down the middle and a Hough transform is applied to each half to find the most prominent lines through the image. The Hough lines are then drawn on the image and the two halves are merged back into a single image. An example image from the robot is shown below. The white lines are the result of the brightest pixel algorithm, and the red lines are the Hough lines drawn by the robot. The image shows how the Hough transform is able to fill in the gap between dashed lines.

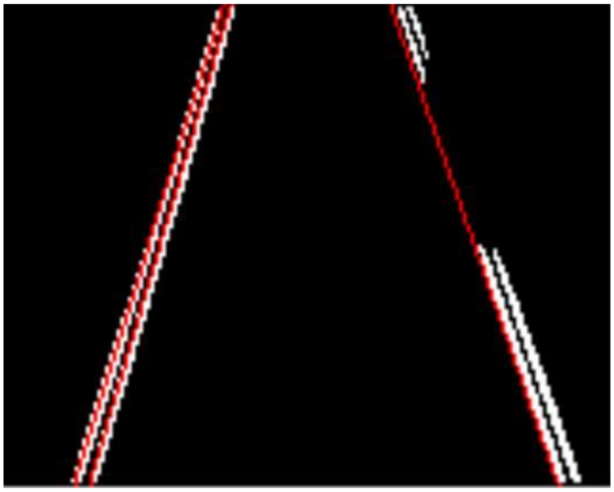


Figure 9: Example of brightest pixel and Hough transform

After the lanes have been found, another image is created from LIDAR data which shows a continuous contour of the obstacles on a black background. An example contour is shown below:

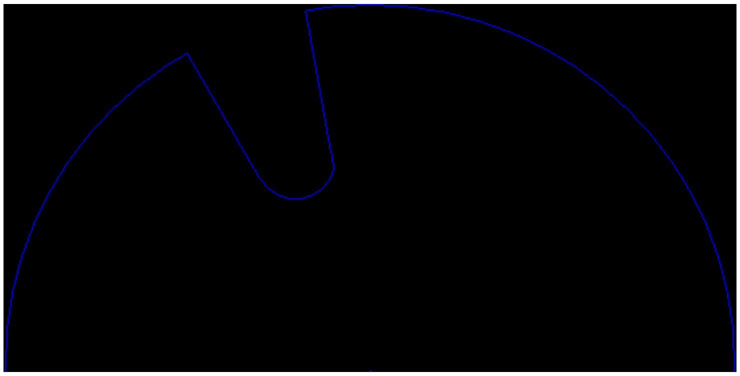


Figure 10: Example LIDAR contour

The arc is the result of LIDAR data being ignored beyond a pre-set threshold. Once the LIDAR image has been obtained, it is combined with the lane image from the camera to form a single composite image. To accurately overlay the image, we created a camera calibration tool that allows manual calibration of distances in the camera image. A grid is drawn over the camera image, and a person marks the floor at each intersection of gridlines in the image, and measures the distances from each point center of the image, and from each point to the bottom of the image. These measurements are entered in the computer and we get a mapping of real-world distances to pixel locations which can be used to accurately overlay the LIDAR contour.

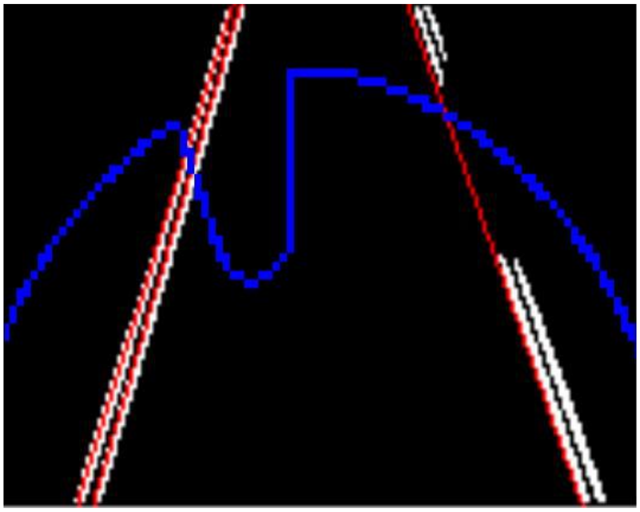


Figure 11: Example of camera and LIDAR Composite

Once we obtain a single image that contains both lane and obstacle data, a blob of a single color is created by filling the area around the center pixel on the bottom edge of the screen. This blob represents the area that is traversable by the robot. The image is then eroded slightly to get rid of any extra lines extending beyond the blob. Once the blob of traversable area has been isolated, a skeletonization algorithm is run to find a path through the blob that is equidistant from the horizontal sides.

Blob Before Erosion Blob after erosion Results of Skeletonization

Figure 11: Image Processing

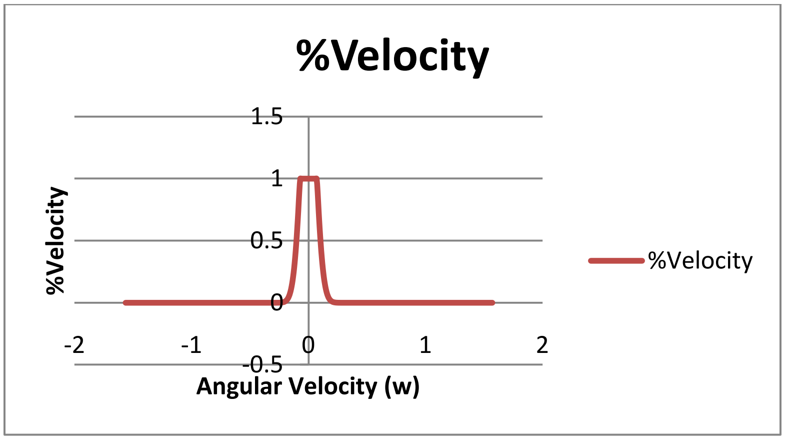
Since this path may branch, the point farthest from the center of the bottom edge of the image along the path is selected as the final destination point. The extra branches are eliminated by using a version of A* to find a path through the skeleton to the final destination point.

5.3 Navigation Challenge Details

The H_2.2Bot's Waypoint Navigation plots the difference between the bearing of the robot and the forward azimuth, or the "Turning Angle" to angular velocity.

$$\left(\frac{\pm\theta}{180}\right) * \ a = \omega$$

Where a is the maximum angular velocity, ω is the resultant angular velocity, and θ is the turning angle. We then plot the angular velocity to forward velocity using a modified Gaussian function.



$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}$$

Let a = 2, b = 0, c = 5, result clipped to [0, 1]

This gives us a smooth transition between turning and driving forward, and avoids 'jerky' motion, which can cause current surges from the motors and mechanically demanding sudden movements. This also allows for continuous self-correction along the length of the path. The parameters were fine-tuned for this system based on the expected error ranges of both the GPS and the compass; the sum of these does not exceed 5% as a matter of worst-case (the value of c, our variance). Lastly, since we are using continuous functions rather than discrete states for Waypoint Navigation, we can encapsulate the algorithm and provide access to the target waypoint and change it in real time, without stopping the process and restarting. This is important in the Navigation Challenge, since we can grab successive points in the A* path finding algorithm and change the target on the fly.

5.4 Plan for Control Decisions

The Platform library contains a Navigation module, which for safety reasons, is tied directly to the Joystick which controls whether the robot is operating autonomously or manually. It also contains an instance of IAutoAction, an interface common to all autonomous actions, so that we can decouple what algorithms we are using from the core navigation code. In general, the IAutoAction is allowed to run until it sets a complete flag or manual control is initiated.

This is beneficial because the algorithms we can write to manipulate the robot are not hard-coded on a per-project basis, and funneling all control through the NavigationDriver prevents clashes and potentially unsafe conditions arising from many objects attempting to use the motors at any given time.

For the Navigation Challenge, we create a standard A* grid and add obstacles as they are verified by the LIDAR. The Waypoint Navigation algorithm is invoked (see **Waypoint Navigation**), repeatedly for points along the path generated by the A* algorithm, which is adjusted as new information becomes available.

6) Performance Specifications

6.1 Safety

6.1.1 Mechanical Safety

We wanted to ensure mechanical safety during both maintenance and operation. For ease of maintenance, we built mounting points on the robot and modified a set of jack stands to allow us to securely mount the robot on even or uneven terrain. The mounting points on the robot are spherical holes that are drilled into solid aluminum blocks which fit around the spherical bearing that we welded to the jack stands. This allows the robot to swivel about the bearing, adjusting for any uneven terrain while still providing a safe mounting. We also included two e-stops. One e-stop is waist high in the center of the back and at a 45 degree angle to maximize accessibility in. The second e-stop is on the front, mounted on the mast, and arm height to allow easy frontal access.

6.1.2 Electrical Safety

H2.2Bot is equipped with hardware emergency stop buttons. There are two onboard red E-Stop buttons, and a wireless E-Stop button. One is located on the center of the switch panel, facing the back of the robot. Another is located on the mast, facing forward. The wireless E-Stop can be activated from 100’ away. When any E-Stop is pressed the motor controller’s enable is turned off bringing the robot to a stop immediately.

The H2.2Bot has high and lower power fuses to protect all equipment if a short circuit occurs. The 36V, 24V, and 12V DC/DC converters are all equipped with short circuit, overload, overvoltage, and over temperature protection.

The RoboteQ AX2860 is also equipped with safety features. There is an automatic stop in case of a command loss from the computer. There is a built in overcurrent protection, along with a temperature sensor that adjusts power limits in response to overheating.

6.2 Robot Performance

Attribute	Design Prediction
Maximum Speed	4.9 mph
Climbing Ability	Up to 30^{o}
Nominal Power Consumption	510 watts
Battery Operating Time	7 hours
Waypoint accuracy (with Omnistar)	150 centimeters

Table 1: Performance Analysis

6.3 Project Cost

		Total - Retail	Our Cost	Unit Price
1	Logitech Wireless Gamepad & Reciever	39.99	39.99	39.99
1	Novatel GPS & Antenna	3496.50	2700.00	2700.00
1	SICK LIDAR	2730.00	2730.00	2730.00
1	PNI Compass	325.00	325.00	325.00
1	RoboteQ Motor Controller	720.00	720.00	720.00
1	Panasonic Camcorder	350.00	0.00	350.00
3	Ovonic Hydrogen Systems Metal Hydride Tanks	2715.00	2715.00	905.00
1	Swagelok Pressure Regulator	406.26	406.26	406.26
1	American Sensor Technologies Pressure Transducer	166.00	166.00	166.00
1	Meanwell 36V to 24V DC-DC Converter	376.92	376.92	376.92
1	D-Link USB Hub	28.99	28.99	28.99
2	Prolific USB to Serial Converter	40.00	40.00	20.00
1	Ballard Fuel Cell	6700.00	0.00	6700.00
16	Maxwell Capacitors	1436.80	1436.80	89.80
2	Magmotor Motors	6060.00	6060.00	3030.00
2	Kendal 10.5 Inch Wheels	174.00	174.00	87.00
2	Casters	64.00	64.00	32.00
4	Bren-Tronics Inc. Lithium Ion Batteries	1200.00	1200.00	300.00
1	Panasonic Laptop & Havis Mount	4003.00	4003.00	4003.00
1	cRIO	5462.00	5462.00	5462.00
8	Flex-core 25A Toroid Rings	1888.00	1888.00	236.00
5	Flex-core 50A Toroid Rings	1180.50	1180.50	236.10
2	Flex-core Large Toroid Rings with Conditioner Boxes	1494.00	1494.00	747.00
1	Analytic Systems DC-DC Converter	1582.00	1582.00	1582.00
1	Mean Well 24v to 12v DC-DC Converter	175.00	175.00	175.00
1	Metal Stock and Fasteners	600.00	600.00	600.00
1	Wire, small electronics, and connectors	1035.00	1035.00	1035.00
		$45,513.96	**$36,602.46**	

Table 2: Budget

7) Conclusion

We are proud to compete in the 2010 IGVC competition with the H_2.2Bot. This project not only serves as a test bed for autonomous navigation, but also as a means for researching energy management aboard a wheel-driven autonomous robot. This robot will serve for a number of years as a data collection and research tool. We hope to apply the knowledge and experience gained in this project to alternative drivetrains such as tracked vehicles and legged robots. These platforms will also be designed to compete in the IGVC as a means to demonstrate our robots abilities.