Lawrence Technological University

H_2Bot

IGVC 2006 Autonomous Vehicle

Team Members

David Bruder, John Girard, Mark Henke, Marcus Randolph, Bill Schwerin, Brace Stout,
Jacob Paul Bushon, Tim Helsper, MaryGrace-Soleil B. Janas, Danielle E. Johnson, Nathaniel Johnson,
Dave Daraskavich, Bill Gale, and Brett Richardson

Faculty Advisor Statement(*)

I, Dr. CJ Chung of the Department of Math and Computer Science at Lawrence Technological University, certify that the design and development on H_2Bot has been significant and each team member has earned credit hours for their work.

Dr. CJ Chung (chung@ltu.edu, 248-204-3504) Date

(*) Co-advisors: Dr. Lisa Anneberg, Dr. Peter Csaszar, Dr. Robert Fletcher, and Dr. King Man Yee

1. Introduction

For the 2006 Intelligent Ground Vehicle Competition (IGVC), Lawrence Technological University (LTU) presents H$_2$Bot, the product of a collaborative effort among a diverse and multidisciplinary group of LTU engineering and computer science students. This paper describes the various aspects of H$_2$Bot design, design trade-off considerations, and improvements over LTU's past IGVC entries.

2. Design Process

2.1 Project Planning Process

A primary goal for the 2006 IGVC was to involve students from all related disciplines in the design and construction of the entry. From the beginning of this year's project, engineers specializing in electronics, computers, mechanical structures, and software have collaborated to produce what we consider a superior autonomous vehicle.

Notes taken during the 2005 IGVC competition were a key to planning this year's entry. Lessons learned, observations, and ideas for improvement along with the 2006 IGVC rules, helped to form a solid basis of goals and requirements for the development of this year's autonomous vehicle.

2.2 Development Process

While agile methods were used for H$_2$Bot software development as in 2005, a new methodology was used in designing the hardware. Specifically, the methodologies employed were benchmarking and set-based engineering. The team relied heavily on benchmarking from the 2005 competition to evaluate not only the observed best practices from other teams, but also innovative ideas generated through the experience of participating in the competition. Expanding the team to include students from all related disciplines allowed multiple design options for key subsystems to be pursued [Ward 1994]. Using set-based engineering favors, while carrying multiple design options in the early stages of development, increased the chances of constructing a completely successful system. The team used mass, cost, and energy consumption models to evaluate various components. Component mockups were constructed to evaluate packaging options. The team investigated gas electric hybrid, battery, and fuel cell power supplies, and chose a dual power strategy of interchangeable pure battery and fuel cell modules. Since the drive system needed significant improvement more than 30 options were considered. The final decision was based on cost, power consumption, torque, speed limiting, and failsafe braking. Keeping options open allowed for many feedback cycles and iterations to obtain a near-optimal design. **Figure 1** illustrates how initial options gradually narrowed over time, but not necessarily frozen at the same point in time. To allow maximum time for software development, the 2005 Think-Tank robot was used as a development mule while the new robot was being designed and constructed. Agile design methodology was used to create a stream of fully functioning versions of control software. The object-oriented nature of the Java language facilitates reuse and sharing of coded functions among the software developers.

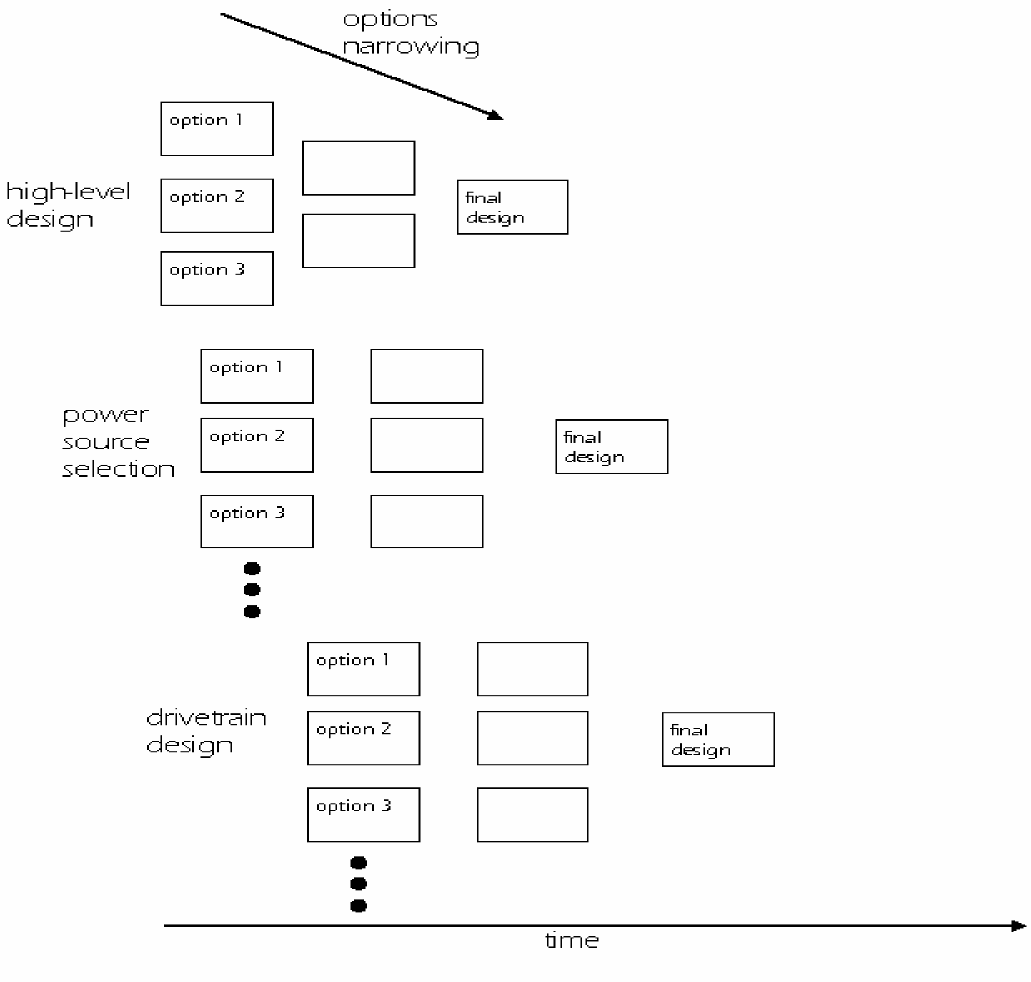


Figure 1. Set Based Concurrent Engineering

2.3 Project tracking and Quality Metrics

A high level project plan was developed in Fall 2005 as follows:

10 / 2005	Concept Integration I
11 / 2005	Concept Integration II
12 / 2005	Proposal and Parts Order
01 / 2006	Integration Build/Test I
02 / 2006	Integration Build/Test II
03 / 2006	Integration Build/Test III & Release
05 / 2006	Submit Design Report
06 / 2006	Competition

2005 mule and 2006 prototype level tests were used to confirm performance of software and hardware against initial targets.

2.4 Collaborative Team Organization

The 2005 H_2Bot team organization can be found below in **Figure 2**. Each functional block represents a self-organizing and managing unit responsible for the three competition events. Thus far, approximately 780 total work hours were contributed to the project by all members of the team.

Electrical Team	Mechanical Team	Software Team
Jacob Paul Bushon, BSEE Tim Helsper, BSEE Mary-Grace-Soleil B. Janas, BSCE Danielle Johnson, BSEE Nathaniel Johnson, BSEE	Dave Darasakavich, BSME Bill Gale, BSME Brett Richardson, BSME	David Bruder, MSCS, Team Captain John Girard, MSCS Mark Henke, MSCS Marcus Randolph, MSCS William Schwerin, MSCS Brace Stout, MSCS

Figure 2. Team Organization

2.5 Vehicle Conceptual Design

The design of the H_2Bot has several features beyond the basic requirements of structural support, rigidity, reliable component mounting, and ample torque. It is constructed mainly from one-inch slotted aluminum extrusion, which makes for easy modifications and adjustments, and provides great rigidity-to-weight characteristics. The vehicle was designed and built in modules; drive train, Laser Measurement Sensor (LMS) mount, equipment box, camera mount, and the two interchangeable power sources, our most notable modules. The rear equipment box panel, which was designed for easy accessibility, can be completely removed, and the power sources can be easily lifted off the vehicle to grant access to the motors and motor controller. Other notable features include the ability of camera mast to break down for transport and the adjustable LMS mount. The mount has the capability to tilt back from horizontal to at least 15 degrees, as well as several different heights for the sensor to give varying fields of view.

3. Hardware Design

3.1 Robot Structure

H_2Bot is a three wheel vehicle with front differential drive steering and a rear caster wheel (as in the 2005 design). It has proven to be an effective configuration that provides zero-turn radius maneuverability and simple motion control. The frame is constructed of extruded aluminum, welded at critical load points (mainly in the drive train), and clamped together at all other connection points. The frame shown in **Figure 3** is enclosed in aluminum sheet metal. A removable body covers the power source and is latched in place (not pictured). This body supports the required payload, as well as a ventilation fans to remove heat produced by the fuel cell and its associated electronics. The fans are shrouded, and the intakes are diverted to prevent water from being ingested. All mating surfaces are either sealed with silicone, or protected by weather stripping. The camera and LMS ports are enclosed as well to round out weather-proofing of the vehicle.

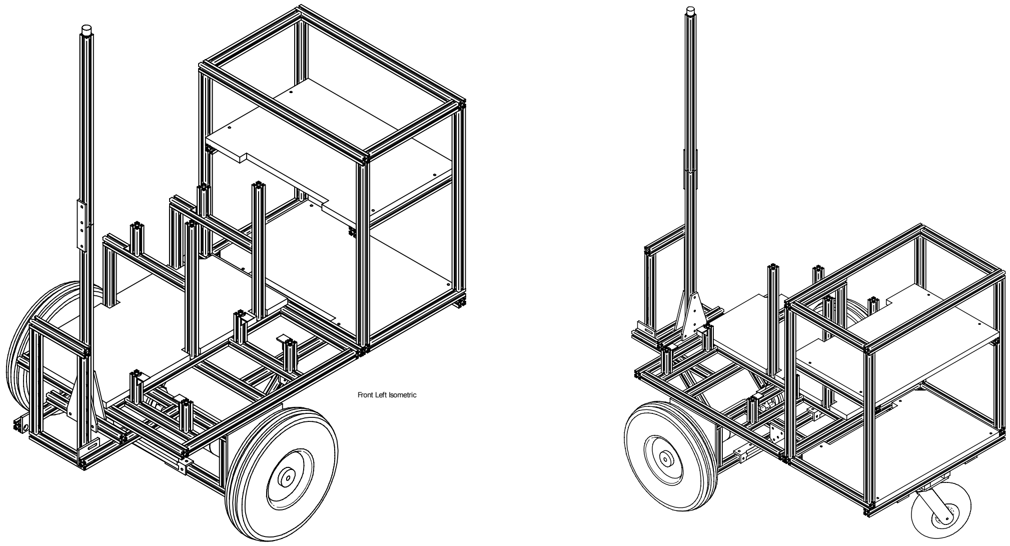


Figure 3. CAD Models of Vehicle Structure

3.2 Drive Train

Significant upgrades in the drivetrain are present in the 2006 design with objectives of lower power consumption, improved speed control, hardware speed limiting, and increased torque. The motor configuration is 3.5 inch diameter, brushed DC servo with windings selected to achieve the required torque, stay within competition speed limits, and achieve the lowest power consumption for a given torque output. Torque capacity is increased by using more powerful motors than found in the 2005 design. Speed resolution is improved by mounting encoders to the motor shaft rather than the gear reduction output. Speed limiting is achieved by selecting motor windings with the appropriate Ke (V/rpm), combined with 16:1 planetary reduction and 24v system drive voltage resulting in a top design speed of 4.2 mph. Planetary gear heads with additional lash allowance were specified to reduce cost over the more costly precision gear heads. A target torque of 389 in-lb per front drive wheel was calculated based on the use-case of climbing over a 2.5” curb. The projected performance is 429 in-lb at a max power consumption of 685 watts / motor. Typical power consumption is about 250 W. Also new in this year’s design are failsafe electromagnetic brakes. When power is removed (such as during e-stop), the brakes engage on both front drive wheels, and quickly stop the vehicle.

3.3 Motor Control

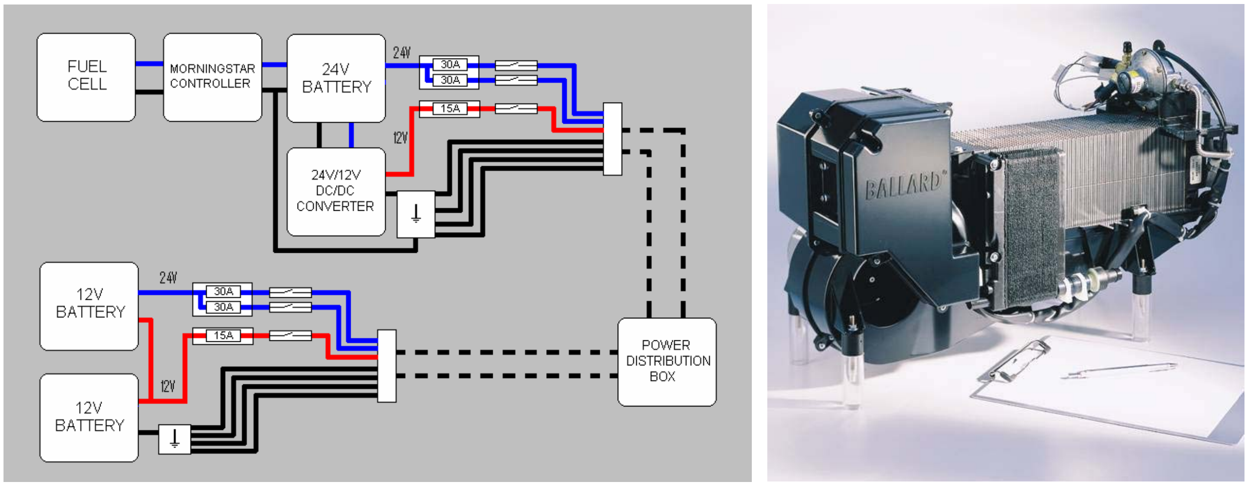
Motor control is facilitated by a Roboteq AX3500 60A/channel controller with serial port interface and included java control class. The AX3500 provides velocity feedback control through optical encoders mounted on the motor shafts. The e-stop is wired to the controller main drive power lines via a mechanical relay which can be triggered by the top-mounted kill switch or wireless control according to IGVC rules.

4. Electrical System

4.1 Power Source

The H_2Bot has interchangeable power systems, each of which is capable of supplying all of the robot's electrical needs.

The primary module uses a Ballard 1.2kW Nexa Fuel Cell Module as the power source (**Figure 4b**). The fuel cell system houses all components required for operation. This includes the Hydrogen Fuel Cell, Hydrogen tank, Morningstar charge/power controller, leak detector, fuel line, 24V battery pack, and 24/12V DC/DC converter for 12V systems. The fuel cell uses 99.999% pure hydrogen gas, and can safely operate both indoors and outdoors as it produces only water vapor and heat as byproducts of electrical generation. A single tank of hydrogen supplies sufficient fuel for the fuel cell to operate at 100% capacity (1200 watts) for approximately 45 minutes, however, H_2Bot normally requires only a fraction of that power. The power output of the fuel cell is actively adjusted with onboard load sensing.



a. Module Scheme **b**. Nexa Fuel Cell

Figure 4. H_2Bot Power Modules

The secondary module consists of two 40Ah AGM Batteries. An onboard charger with 110VAC interface is employed to restore battery power as required.

The installed power module feeds directly to a power distribution box (as shown in **Figure 4a**), which distributes power to the vehicle. Switching power modules is a straightforward process, and can be done in a matter of minutes. The modular design also allows additional power sources for H_2Bot to be incorporated with a minimum of effort.

4.2 Power Distribution

A box-mounted printed-circuit board (PCB) distributes and switches power to each of the electrical components. Molex® and Phoenix Contact® connectors provide a quick-connect interface used when switching between power modules or testing. A separate wiring harness routes power to the motor controller and the power distribution PCB. The PCB provides connections for the E-stop wireless device and a DC-DC converter to supply regulated 24V power required for sensitive electronics. Toggle switches mounted to the box allow the power to each device to be switched individually. The power and communications control system schematic for H_2Bot vehicle is shown in **Figure 5**.

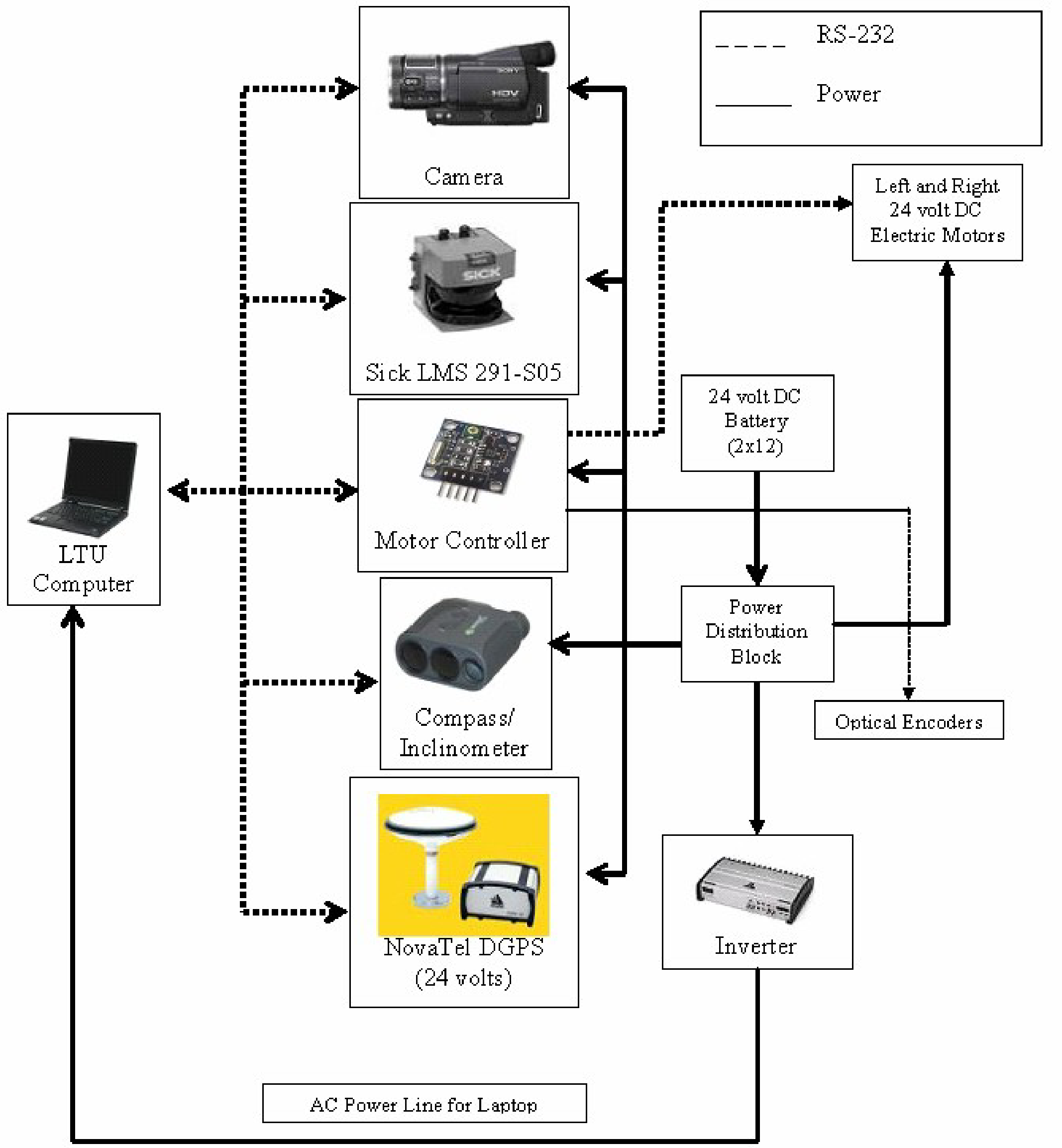


Figure 5. Power and Communications Control System Schematic

4.3 Computer Hardware

The computer hardware incorporated into H_2Bot is a Pentium M Laptop. During development and testing, a "remote desktop" application is utilized, which is accessed using 802.11g WiFi (also used in JAUS checkout). This arrangement is much more convenient for development and was one of the ideas that came out of the 2005 lessons-learned.

4.4 Sensors

In keeping with the theme of modularity, all H_2Bot sensors (and controls) utilize either RS-232, USB 2.0, or IEEE 1394 (FireWire) standard interfaces. **Table 1** summarizes the sensors used in H_2Bot.

Sensor	Function
Optical Encoders	Motor shaft position feedback for servo control
NovaTel ProPack-LB DGPS unit	Global positioning
Digital Compass/Inclinometer	Heading, roll, and pitch information
High Dynamic Range DV Camera	Capture field image stream
Sick Laser Measurement Sensor	Provides a 180 degree polar ranging array

Table 1. Vehicle Sensor Summary

4.5 E-stop

As outlined in the IGVC safety rules, the H_2Bot is equipped with both a manual and a wireless (RF) remote emergency stop (E-Stop) capability. To satisfy the "hardware-only" requirement, a KE100-BFR-CL Remote Keyless Entry System manufactured by Bulldog Security is used as the wireless E-stop component. This wireless system is normally configured for the remote control of systems on a passenger car (remote start, door locks, trunk release). The trunk-release function is integrated into the H_2Bot's E-stop system.

4.6 Vehicle Alert System (JAUS Horn and Lights Hardware)

H_2Bot includes a module that can switch relatively large electrical loads (using relays) controlled with commands sent over USB interface. This module is utilized by the implementation of the vehicle alert system, which is exercised during JAUS checkout. The module is used to switch power to a horn and a light under software control.

5. Software Design

5.1 Software Strategy

The H_2Bot software expands on the hardware interface layer developed in 2005, and continues the use of freely available and open source Sun Java and Eclipse IDE. Java provides a flexible, portable, and scalable approach to multi-threaded software development, all of which are important as the team continues to build on previous work.

5.2 Software Architecture

Figure 6 shows a high-level view of H_2Bot software architecture, reflecting improved modularity and opportunity for reuse over the previous architecture. Of particular interest is that the differences between autonomous and navigation challenge code are contained entirely within the path planning module.

The suite of on-board sensors collects raw data for processing. Driver software, specific to each sensor, supports initialization, calibration, and data reporting functions. The time difference between data acquisition and data sense has been characterized for each device, and 'real' acquisition time is provided with reported data where appropriate. Interpolation is used as necessary to estimate intermediate values.

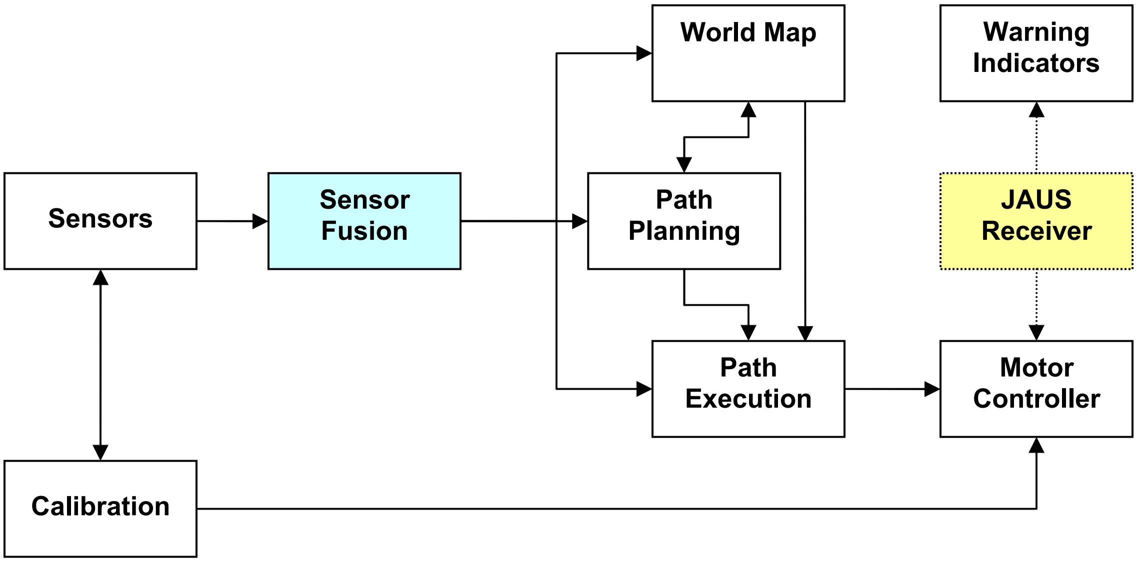


Figure 6. H_2Bot Software Architecture

5.2.1 Calibration

While the H_2Bot can perform well immediately upon setup, the sensor fusion algorithms work best when the sensors have been calibrated to each other. The various calibration algorithms performed are described briefly as follows:

- Camera registration: Cameras are registered against an object of known dimension and placement. Parameters for mapping pixels to corresponding rays are determined.
- Encoder registration: Each wheel of the robot is engaged independently to determine the correspondence of encoder pulses to distance traveled. This is used when determining position by dead reckoning.
- Compass alignment: Fixed local magnetic fields can influence the accuracy of the electronic compass. The H_2Bot (having its encoders properly registered) is commanded to travel in a straight line at maximum speed. The heading reported by the GPS unit (determined using Doppler calculations) is used to calculate compass corrections.
- Tilt sensor registration: The H_2Bot drives in a straight line for a specified distance, makes a 180 degree turn, and drives over the same track for the same distance. The roll and pitch sensor values are averaged to determine the tilt sensor corrections.

- LMS registration: An object of known dimension is passed through the LMS plane of detection. The sensed values are compared against those observed by the cameras to determine the LMS plane of detection. Extension of the plane parameters enables mapping of angular/distance measurements to fixed points in 3D space.

5.2.2 Sensor Fusion

When either the autonomous or navigation functions are enabled, H$_2$Bot marks its initial position as the "origin" of a north-oriented map. During this operation, H$_2$Bot constantly updates its current position and orientation through the monitoring of the encoder pulse counts. The orientation is adjusted as necessary to track the sensed orientation coming from the compass. The H$_2$Bot GPS, even while receiving CDGPS corrections, is not as stable as the dead reckoning algorithm. To support the accuracy required by our sensor fusion algorithms, a differential GPS solution using a fixed local base station would be required. Combining position data with the tilt sensor data provides the necessary information to construct the 3D transformations between the H$_2$Bot local frame of reference and the world frame of reference. From these, the world-coordinate position and orientation of each camera and the LMS is obtained.

Each camera driver is responsible for classifying the pixels in its acquired image as either a background or non-background pixel. Background pixels represent grass, dirt, sand, sky, etc. Non-background pixels represent white, yellow, and orange colors, corresponding to obstacles or things to avoid. A map (similar to an occupancy grid) is maintained. The position on those maps where a background pixel is sensed, at which the corresponding ray intersects, is marked as a negative one value with a heavy weighting. Where a non-background pixel is sensed, the corresponding position is marked as a positive one value with a light weighting. As additional images are processed, either from different cameras or from different viewpoints of the same camera, an accurate map of the area is constructed, with clear areas marked with negative values, and occupied areas marked with positive values. A separate map is maintained for the LMS, which is used to corroborate obstacle information from camera images. Where the LMS senses an obstacle, an area around the corresponding position is checked on the camera images. Where the images show no obstacle, no obstacle is marked on the LMS map. Where the images indicate an obstacle, the LMS map is marked. Outputs from the sensor fusion process are as follows:

- The world map, formed by combining the maps generated from the camera and LMS using a maximum value function
- The current position and orientation

5.3 Video Scene Analysis

The primary purpose of the IGVC vision system is to identify obstacles that threaten the vehicle's progress. Two general types of obstacles exist: those with definite shapes (such as potholes, construction barricades, and other impediments), and those with variable shapes (such as the field lines). This design detects both types of objects reliably. When identified, such objects are placed in the world map.

5.3.1 Preprocessor (Illumination Correction)

The preprocessor's task is to remove as many unimportant factors from the image as possible. One of the most important factors in image analysis is illumination, especially when color is utilized as a feature in later processing steps. A change in illumination, especially a change in illumination intensity, such as darkening or lightening of the scene, could change the contents of the image enough to prevent the system from correctly identifying objects. To reduce the effect of illumination upon the later processing steps, a color card is used to reference the input scene's general illumination to ideal illumination conditions.

A color reference card is placed within the camera's field of view with known colors. **Figure 7** is an example color reference card, optimized for the hue- saturation-intensity color space.

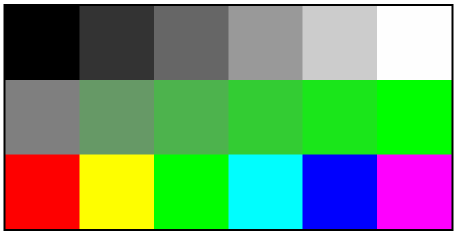


Figure 7. Color reference card

This color reference card is placed on the top surface of the robot so the reference card will receive the same general illumination as the rest of the scene. From the information known about the card and from the data returned about it by the vision sensor, the illumination of the scene as a whole can be calculated, and the image is corrected to match an ideal illumination value.

5.3.2 Complex Feature Profiler and Comparator

After the scene is preprocessed, complex features in the scene are detected, labeled, and tracked. Construction barrels, construction barricades, pails, and potholes are examples of complex features. Note that translation and scale are all permitted to vary freely; the detector must be capable of detecting specific objects at any location in the image and at any scale.

To detect a specific feature, the detection system must be aware of the characteristics of that feature. Several different types of profile-generating algorithm are available. The introduction of color histograms as a search method originated in [Swain 1991], and such a mechanism would be one of the simpler profiling methods. For example, the construction barrel in **Figure 8** consists of about 15% white and 85% orange. This information can be converted into a histogram with spikes at orange and white.

The feature profiler generates a profile for a desired feature. Many of these profiles will exist in a feature database, accessible to the detection system. In addition, segments of a scene are represented in the same way as desired features, and the comparison of a desired profile versus a candidate profile is used to identify objects.

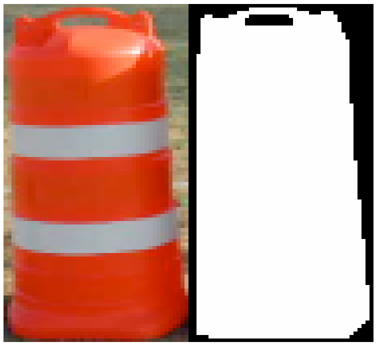


Figure 8. Input image for profiling a construction barrel (image and mask)

5.3.3 Complex Feature Tracking

The next part of the complex feature system is the feature tracking system. This module handles the challenge of tracking a known profile through an entire search space. It will use a limited genetic algorithm to search a range in and around the original feature bounds.

Once a known feature is definitively relocated in the next scene frame, its search rectangle is removed from the scene. Since the system now is relatively sure that the search rectangle contains a known feature, it does not have to waste computational power on analyzing that section further. More than one feature can be tracked at the same time. The more complex features that can be pulled out of the image, the better chance later steps have at picking out more ambiguous features (such as field lines).

5.3.4 Complex Feature Detection

At this point, the detection system has the tools it needs to detect complex features in an input scene. First, any known objects are located and parsed out of the image. If no features are known, then the entire image is searched for complex features. Second, a simple genetic algorithm is executed upon every input scene. Anywhere from dozens to hundreds of search rectangles are inserted into the genetic pool. Crossover and mutation is used to propel the genetic algorithm until the genetic algorithm completes (determined by some end-condition, such as total generations executed). If the best feature match in the scene is over some threshold (such as 80% confidence that the feature in the best search window is the compared object), then that feature is officially recognized, and its details are passed to the complex feature tracker. Once the input scene leaves the complex feature detection system, all detected complex features will have their pixels removed from the image, making the image simpler for analysis of more ambiguous features.

5.3.5 Simple Feature Detection

The final interpretation of the image is simple feature detection. In this case, the simple features are line segments representative of the painted field lines. The ability to detect multiple lines in the image leads to several advantages. In this manner, curves are detected via the linking of two segments, and both boundary field lines can be interpreted in the same search space. Therefore, a multimodal (multi-solution) genetic algorithm is a good choice to detect simple line segments in the simplified image.

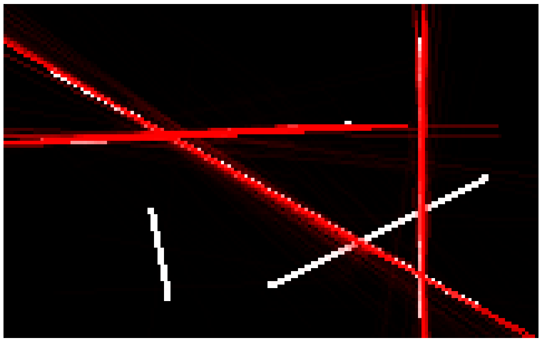


Figure 9. Implementation of the multi-niche crowding algorithm on a test image

The best candidate found is the implementation of a genetic algorithm with multi-niche crowding [Cedeno 1994]. **Figure 9** illustrates an advanced solution from a multi-niche crowding implementation.

5.4 Virtual World Map

The virtual world map is implemented as a 2D array representing the vehicle's environment in terms of traversable space and obstacles. This is the basis for path planning in both the Navigation and Autonomous Challenges. The system populates the map as the vehicle progresses based on sensor inputs.

5.5 Path Planning

Path planning uses information from the world map as well as current position and orientation to determine a path forward. Paths that come too close to perceived obstacles are discounted. As path planning finds "traps", it marks those positions on the world map as having obstacles, so as to avoid the same trap in the future. The objective of path planning is different depending on the challenge type:

- For the autonomous challenge, the objective is to lengthen the path from the starting position while maintaining a safe distance from obstacles. Precedence in consideration is given to paths that continue in the direction H_2Bot is facing.
- For the navigation challenge, a shortest-path is determined from the present position, the unvisited destinations, including the starting position, and what is known of the world. That path

is reevaluated continuously as information is gathered from the sensors. The path provided to path execution is the current best path so determined.

5.6 Path Execution

Given the current position and orientation, and the desired path, the path execution module determines the commands to send to the motor controller to best achieve the desired path. Path execution is aware of H_2Bot dimensions and avoids swinging the caster wheel end into obstacles. Path execution is free to optimize the movements of H_2Bot to minimize elapsed time while following the path. For example, it will back out of a trap rather than taking the time to turn around.

5.7 Radio Control

A wireless joystick was implemented for ease of movement when H_2Bot is not competing. The joystick controls H_2Bot direction and speed. When the joystick is engaged, autonomous control is disengaged.

6. Performance Analysis

Vehicle performance was estimated based on the design parameters. These same attributes were measured after integration as shown in **Table 2**.

Attribute	Design Prediction
Maximum Speed	4.2 mph
Climbing Ability	2.5 inch curb
Nominal Power Consumption	500 watts
Battery Operating Time	4 hours
Hydrogen Operating Time	2 days
Brake holding capacity	>30% grade
Waypoint accuracy (with Omnistar)	.1 to .8 meters

Table 2. Performance Analysis

6.1 Safety

H_2Bot safety features include manual and remote E-stop, failsafe brakes, and hydrogen leak detection shutdown.

6.2 Reliability

The reliability of H_2Bot is improved compared to the 2005 design in drive motor function and power availability. These items were improved by complete drive system redesign and the development dual modular power sources with increased capacity.

6.3 Durability

Durability improvements include the extruded aluminum frame and redesigned drive system. The new frame is significantly stronger and more rigid than last year's fiberboard construction. The drive system has design margins that exceed competition requirements. (Last year's model was not capable of climbing over the front lip of the ramps in the Autonomous Challenge.)

6.4 Testing

Hardware and software tests were conducted under a three phase test plan which included: unit testing of each component by the developer as the components were created, integration testing as each component was integrated into the entire system to check that the components functioned properly as a system, and regression tests to verify that previously tested components did not introduce bugs after integration with other newer components. Practice courses were set up to test the functionality of the robot for both the Autonomous and Navigation Challenges.

6.5 Systems Integration

The systems integration model was a direct benefit from the object-oriented programming model and was aided by the use of Java interfaces and a custom written threading model. Hardware integration was facilitated by the modular design of the chassis and the use of electrical buses to deliver power to each subsystem. Hardware and software systems integration was performed during each increment of the development.

6.6 JAUS Analysis

This portion of the report describes H_2Bot's JAUS-enabled features and addresses the required competition elements.

6.6.1 Process for Learning JAUS

JAUS specifications were first introduced to the group by a faculty advisor. For the unanimous interest in the challenge, the group researched additional information online, primarily JAUS' website.

6.6.2 JAUS Integration into the Design

According to JAUS protocol, the students will be given message commands, transmitted from the operator control unit, via RF (Radio Frequency) data link.

A JAVA class called *JAUSListener* implements the following steps when a message is received.

- The message is decoded: Using the embedded protocol, specified in the Reference Architecture Specification document, JAUSListener checks if a command code (byte 12 and 13) is received.
- Depending on the received command code, JAUSListener invokes the necessary function that implements the specified command. The process of the demonstration will be as follows:
  - ✓ To start the vehicle moving forward in the autonomous mode: resume message <Cmd Code = 0004h>
  - ✓ To stop the vehicle from moving forward in the autonomous mode: standby message <Cmd Code = 0003h>
  - ✓ To activate the warning device such as a horn or a light: discrete devices message <Cmd Code = 0406h>

6.6.3 Challenges Encountered

The technical terminology used in the Reference Architecture Specifications was difficult to understand at first. However, continued research and team collaborations provided the team with a full comprehension of JAUS specifications.

6.7 Vehicle Cost Summary

Table 3 summarizes the total material cost for the H_2Bot vehicle. This is the most expensive and sophisticated vehicle developed by LTU teams so far.

Component	Total Cost	Team Cost
(1) Sick LMS 291-S05 LMS*	$7,000	$0
Nexa 1.2 KW Fuel Cell	$7,000	$7,000
(1) NovaTel ProPack-LB DGPS & GPS-600-LB Antenna*	$2,700	$0
(1) MPC Laptop*	$2,215	$0
(2) Brush DC planetary gearmotors with e-brake	$2,000	$2,000
(1) JVC TK-C1480U Color Video Camera*	$1,000	$0
(2) Hydrogen tanks + fuel	$900	$900
(1) PixelLink PL-A544 Video Converter box*	$500	$0
Electrical Hardware	$500	$500
(1) Digital Compass/inclinometer*	$400	$0
(1) Roboteq AX3500 Dual Channel Motor Controller	$395	$395
(2) Main Battery 12 Volt 40 Ah AGM	$150	$150
Chassis Materials	$300	$0
Misc Hardware (nuts, bolts, etc…)	$200	$200
(2) Hollow Shaft Optical Encoder Kit	$122	$122
(2) 14" Tire & Wheel Assembly	$58	$58
Total	$25,440	$11,325

*reused from 2005 vehicle

Table 3. Vehicle Cost Summary

7. Conclusion

H_2Bot continues the LTU tradition of innovation and continuous improvement. Designing the vehicle from scratch by the incorporation of the pioneering fuel cell power source, modular electrical system, new drive system, and intelligent software, H_2Bot will make an interesting and competitive entry.

8. References

[Ward 1994] Ward, A.C., Sobek, II, D.K., "Toyota and Set-Based Concurrent Engineering," 1994 ASME Design Theory and Methodology Conference Proceedings, Minneapolis, MN

[Ballard 2002] An Introduction to Fuel Cells and Related Technologies, Ballard Power Systems, Inc. 2002

[Swain 1991] M. J. Swain, D. H. Ballard, "*Color indexing*", Int. J. Comput. Vision, 7:11-32. 1991.

[Cedeno 1994] Cedeno, W., and V. Vemuri, "Multi-niche crowding in genetic algorithms and its application to the assembly of DNA restriction fragments, Evolutionary Computation,, Vol. 2, No. 4, pp 321-345, Winter 1994.