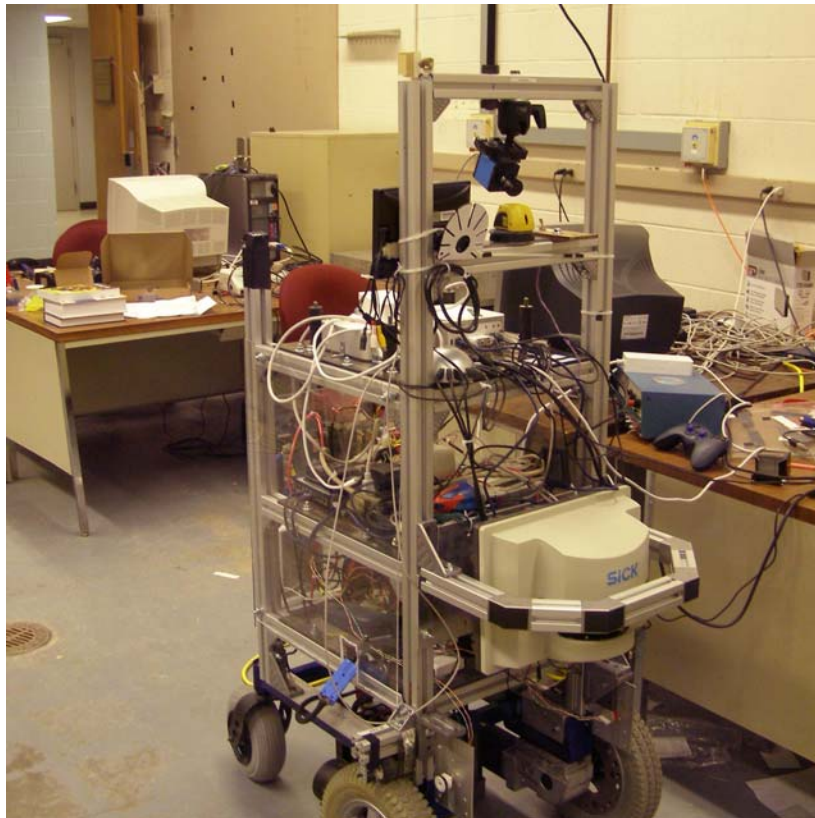




CASE

CASE WESTERN RESERVE UNIVERSITY



Team Case's IGVC Vehicle "Roberto" in Lab

Faculty Statement

"Roberto" was originally designed and assembled by senior-project students for the 2006 IGVC competition. During Spring 2007, four new senior project students focused on redesigning some of the sensing systems, the computing system, and all of the software. These students each received 4 semester-hours of design credit.

Prof. Wyatt Newman, EECS Dept, CWRU

1. Introduction

Case Western Reserve University and Team Case are proud to present Roberto for the 15th annual Intelligent Ground Vehicle Design Competition. Roberto made his debut at the 14th annual IGVC. The vehicle has been redesigned for this year's event with an emphasis on software.

2. Design Methodology

To redesign Roberto, the team took a look at the design specifications from the previous year's robot. The original hardware was sufficient for last year's competition, so much of the original frame and hardware components were used for this year's vehicle. Because of new rules and regulations in this year's competition, the requirements were re-organized into new software components. Each member then took responsibility for different components of the vehicle.

3. Physical Design

The framework of the vehicle consists of a lower frame and drive unit as well as an upper frame with mountings for sensory and control equipment. Specifications for the frame were re-evaluated early in the project and it was decided to only modify the existing frame through the use of additional mountings. The additional mountings resulted in increased height over last year's vehicle.



The Lower Frame is the same frame used in last year's design.

3.1 Lower Frame and Drive Unit

The lower frame, wheels and motors were graciously donated by Invacare. The steel frame was custom built to meet the specifications set forth by last year's team. Two large front drive wheels are independently powered by electric motors. Stability is provided by two castor

wheels in the back. Figure 1 shows the lower frame assembly. For this year's vehicle, two new rotary encoders were installed where the base meets the frame.

To hold the pair of 12V sealed AGM batteries, a battery box was originally drawn and manufactured. The custom design allows it to fit in a relatively small space and provides for easy replacement of the batteries. Mounted in the middle of the frame, the battery box provides stability and traction to the vehicle. No changes were made to the battery box for the 15th IGVC.

3.2 Upper Frame

Bosch 40mm extruded aluminum structural framing components were used to construct the upper frame. This option was chosen because it is both light and strong. The design of the bars provides for easy repositioning of components and simple modification of the frame itself. It was also important that the upper frame be non-magnetic. In the front of the vehicle, two bars extend up to near the maximum allowable height. The boom allows the GPS antenna to avoid interference from other devices. It also places the compass away from magnetic interference from the steel lower frame, motors, and other devices. Finally, the raised section enables the camera to view the course from a high vantage point.

Behind the instrumentation tower, the upper frame provides several polycarbonate shelves. Each shelf hosts different components and provides protection from weather and debris. The two Mac Mini computers are located on the upper polycarbonate platform. They are both securely attached with angle brackets and a shelf above the computers provides protection from any objects that may strike the top of the robot.

4. Sensors and Control Design

The sensor array was designed for accurate detection of markings and obstacles as well as ease of use and modification. The processing requirements of each potential sensor were also considered, as "real-time" response and decision-making requires constant, efficient processing speed. The motor control was designed to be safe, responsive and accurate.

4.1 Vision

Machine Vision is the most intensive computational process within the software system. Because of this, one of two Mac Mini computers is utilized for image processing. The vehicle

has one camera which is centered at a height of 5 feet on Roberto. The camera is linked directly to the vision computer with a (IEEE 1394) Firewire cable, which boasts a significant acquisition speed improvement over last year's Ethernet model. Vision processing is comprised of four major components: color mapping, geometric transformations, template matching, and lane detection. For color mapping, the system is "trained" with manual assistance. Representative snapshots are acquired and displayed for user interaction. The user selects locations in the scene that are judged to be examples of key colors (notably, green, white, yellow and orange), and the corresponding (neighborhood-averaged) RGB values are stored as examples. The algorithms for color-matching were developed for the 2006 competition, but they were re-implemented for the 2007 competition to leverage the LabVIEW programming environment.

The second major processing task is geometric transformation. The camera mounting pose and the wide-angle lens introduce warping and perspective distortions. Images are transformed into equivalent top-down (plan) views, aligned with world coordinates. This transformation is performed on the basis of training images (snapshots with fiducials located at known coordinates with respect to the robot) and 2-D isoperimetric interpolation.

The third image-processing module performs object recognition. Templates for lines, barrels, barriers and potholes are compared against regions within images, and labels are assigned to objects on the basis of template correlation values.

The final aspect of vision processing is concerned with lane detection for the obstacle course. The algorithm for the lane detection divides the image in half vertically, and then processes each half independently, looking for the boundary lines. The lines are detected by making a series of horizontal scans in each half, looking for peaks in color intensity. By only processing a subset of the pixels, we meet our efficiency requirements. Once these peaks are detected, a polynomial is fit to the peaks in each half of the image. To help ensure consistent data, the polynomial fit is then compared to a "long-term" fit line. If the current fit is not within a desired threshold of the long-term fit line, the current fit is ignored, and the "dirty" bit is set for the line, indicating invalid data. Once the two lines are correctly identified and validated against the long-term fit lines, two streams of coordinates are sent through a data socket to the control computer. The lane detection system represents a complete redesign of last year's system, which relied on color matching.

4.2 Obstacle Detection and Ranging

The primary means of obstacle detection and ranging is the SICK LMS 221 laser range finder. The LMS (Laser Measurement System) scans up to a 180 degree sweep in front of the vehicle. Data is taken at 0.5 degree increments, and each point is accurate to within 10cm. Using the data gathered during each sweep, obstacles in range are mapped using polar coordinates. In order to increase throughput, the standard RS-232 interface was replaced by RS-422 which will allow communication at 500kbaud. The LMS is mounted low to the ground and the scanning beam is angled up slightly to avoid detecting inclines or ramps as obstacles. Figure 2 shows the LMS mounted on the front of Roberto.



Figure 2: SICK LMS 221

Data from the LMS will be combined with vision information to provide an enhanced model of the world. The camera will help detect low lying obstacles, as well as simulated pot holes that would not be detected by the LMS. The LMS provides faster updates than the vision system and has very accurate distance measurement, so it works well as the primary obstacle detection sensor.

Very little about the LMS was changed from last year's design specifications. The Added RS422 to USB converter allows the LMS to interface directly with one of the Mac Mini computers used on the vehicle.

4.3 Heading and Position Determination

To navigate a vehicle, its latitude, longitude, and heading must be known. The more accurate these figures are, the more certain the calculated position is. Information for finding these three variables comes from three sources: GPS readings, wheel encoder data, and movement of surrounding objects. By combining data sources, sensor accuracy is increased as well as the state being maintained if one of the sources drops out.

Latitude and longitude are mainly provided by GPS. The GPS signals are provided by Omnistar and are stated to be accurate within ten centimeters. In testing, the latitude and

longitude figures were found to be accurate to several decimal places, verifying Omnistar's accuracy claims. However, in case of GPS dropouts, integration of the wheel encoders is used.

The wheel encoders pulse every time the shaft rotates a certain amount (calculated to be $1/413$ of a wheel revolution). These pulses are counted and provide a total distance moved when multiplied by the distance moved per pulse. At this point, a key assumption is made: the vehicle moves in a straight line and then in a circular path during each individual iteration. This allows the change in latitude and longitude to be calculated with a straight line and then with circles, greatly simplifying the calculation.

Since the radius of curvature of each wheel's path is known, the center of the vehicle has a radius of curvature of the width between the wheels. Given this, the change in the x direction can be calculated based on how much farther one wheel traveled than the other. The change in the y direction can be calculated based on the straight line distance plus the distance in the y direction traveled on the circle. These values are then multiplied by the sine and cosine of the heading (more on this soon) and added to the previous latitude and longitude, updating the value.

The heading is calculated mainly from wheel encoders with help from the GPS receiver. For the encoders, the change in heading is also calculated with the circle based on excess movement by one of the wheels. The circumference of a wheel's circular path is just the distance between the wheels times pi. Heading change is just the ratio of wheel distance divided by the circular path's circumference times 360 degrees.

However, if the wheels slip, the above method will not be accurate. To fix this GPS is used to correct the heading. This is achieved by placing "tent stakes" each time the vehicle travels a meter. After the GPS moves a meter away from the last tent stake, the integral of the path of the GPS points since that tent stake was laid is calculated. If the total distance traveled is close to the radius of the circle then the heading between the tent stake and the point outside of the threshold is close to the true heading. If the distances are not close to equal, the heading is inaccurate and ignored.

By combining these two methods, an accurate heading is maintained at all times. When combined with the latitude and longitude, all things about the vehicle's position are known.

One final feature which is not implemented yet but may be in the future is the use of the surroundings to infer position and heading. This is done by identifying characteristic points in the LIDAR or vision map and tracking their movement over time. Currently, code is written to

track corners and to fit lines to a LIDAR map. However the corners are not always accurately found and the lines seem to jitter too much to make for accurate position inferences. This method is still being investigated and will be until the competition.

4.4 Control

To transform low power computer signals to high-current motor control, two Curtis 1228 drives were chosen. The drives were chosen because they match the power requirements of the motors and they can be controlled with a simple analog signal. The drives use pulse width modulation (PWM) and back-EMF feedback to control the speed of the motors. The control information is sent to the motor drives from the control computer (a Mac Mini) via National Instruments USB-6008 analog and digital I/O devices. Figure 3 shows the two motor controllers, the two NI units, and the wiring in between. The only change that was made to the motor control for the current competition was the replacement of a National Instruments DAC.

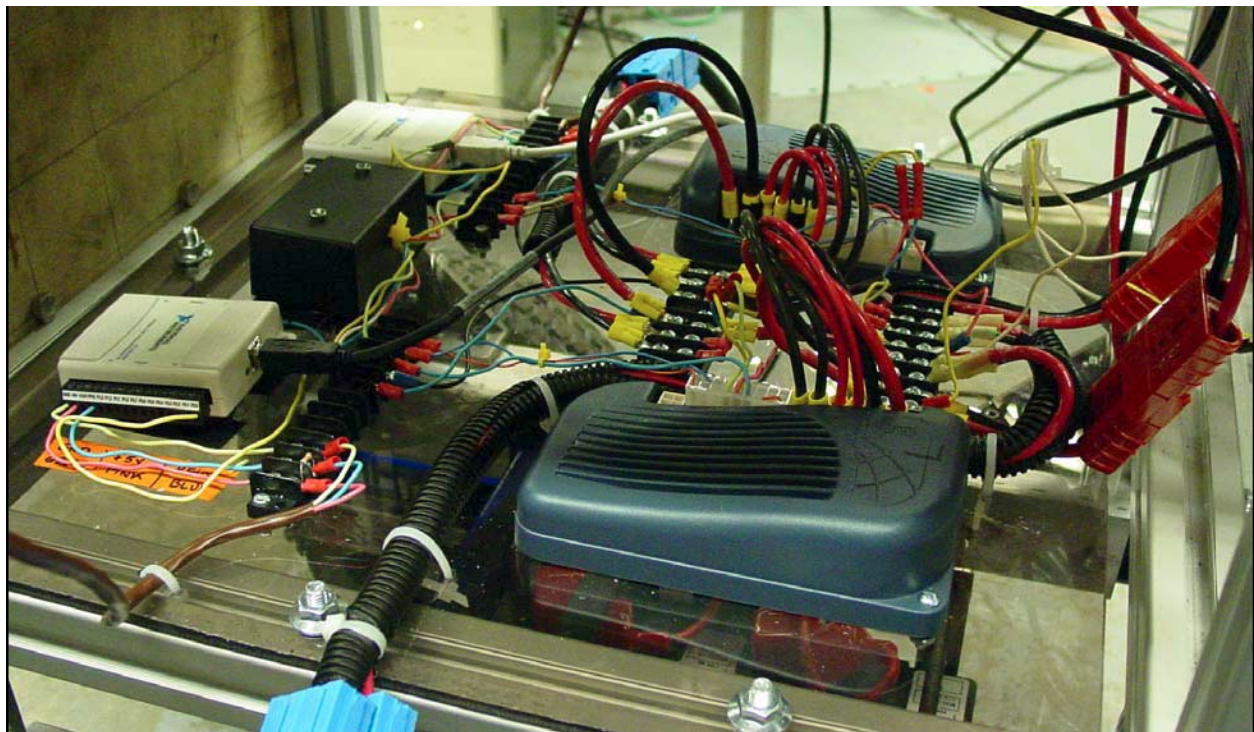


Figure 3: Motor Control Assembly

4.5 Speed and Distance Monitoring

Last year's model used a Hall Effect sensor mounted next to each drive wheel and back-driven LEGO Mindstorms NXT servo motors to gauge wheel motion. Through testing it was found that the Hall Effect encoders and LEGO servo motors were inadequate for accurately measuring wheel motion. This design flaw was overcome by installing a high-precision rotary shaft encoder for each wheel. The rotary encoder is driven by a timing belt originating from the axle connecting the wheel to the motor. These shaft encoders send 256 pulses per encoder revolution and pulse 413 times total for each wheel revolution (due to difference in shaft diameters). This provides a wheel position accurate to .07 cm if no belt slippage occurs. The mounts for the other sensors remain, but the sensors themselves have been removed.

5. Electrical Systems Design

Two 12V sealed AGM batteries are wired in series to provide 24V to the entire vehicle. The power supply from the battery is routed through a 100-Amp circuit breaker/cut-off switch. Past this breaker, the 24V supply is routed to the motor drives and to a breaker/distribution panel. Enabling the motor drives requires not only 24V power (via the main breaker), but also a 24V enable signal. This signal is routed in series through a manual E-stop button and a wireless-remote E-stop.

Several devices require voltages other than 24V. The wheel encoders require 5V, which is provided by the NI USB-6008 directly. The LMS needs 24V, which is isolated from power-supply noise introduced by motor current draw. Simple power conditioning for the LMS is done with a single rectifier diode and a large capacitor. Additional devices (remote E-stop, network switch and GPS) require 12V, which is derived from the 24V supply via a DC-to-DC converter. Finally, 120VAC is provided by two inverters. Two mounted Mac-Mini computers and a touch-screen display require 120VAC, but the computers require separate inverters to provide enough current to each unit. Power for the LMS, the DC-to-DC converters and for the 120VAC inverter are routed through separate 15-A breakers/switches.

6. Software Design

The control software has been completely redesigned from last year's vehicle. The goals for the software design were reliability, portability, and modularity. For the 2007 competition, National Instruments LabVIEW software was chosen, because it provides a naturally modular

architecture, and the code is visual, meaning both new and experienced users can easily read and program the modules. Because Roberto is intended to be an experimental platform for several projects, LabVIEW was a natural choice, and helps to make Roberto much more versatile than last year's version which implemented most of the control in C code..

6.1 Software Organization

The LabVIEW modules reside on separate Mac-Mini computers and communicate with each other via TCP/IP sockets. There exist two versions of some of these modules: one for the Navigation Challenge and one for the Obstacle Challenge. The vehicle will obey a wall-following algorithm to navigate around obstacles.

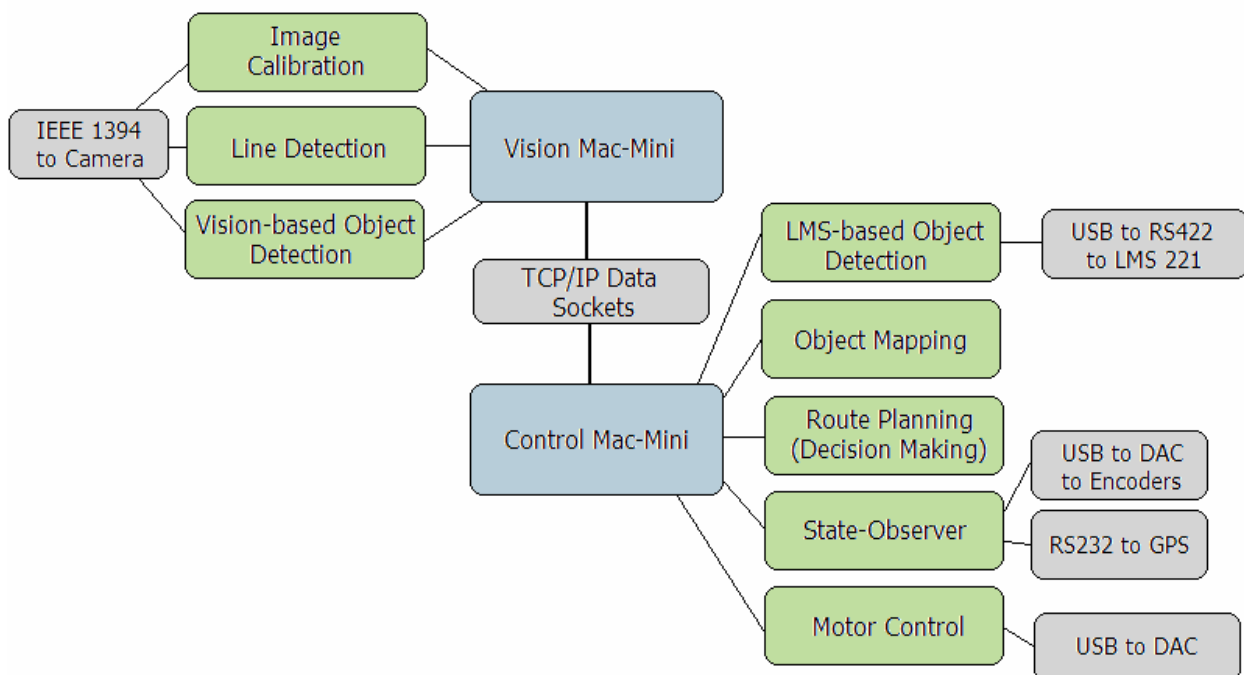


Figure 4: Software Layout Illustration

One Mac-Mini computer, the vision computer, communicates directly with the vehicle's camera via Firewire (IEEE 1394). The modules for Line Detection, Image Calibration, and Vision-based Object Detection reside on the machine. Because vision-based computations are very CPU-intensive, it was decided early in the design phase to have a dedicated machine for vision-based operations.

The second Mac-Mini computer, the control computer, communicates directly with the LMS, the vehicle motors, and the vehicle encoders. The modules for LMS-based Obstacle Detection, State Observation, Obstacle Mapping, Path Finding/Decision Making, and Vehicle

Control reside on the second machine. Because the LMS-based obstacle detection module handles numerical data sent from the LIDAR, it was decided that the Control Mac-Mini would run that module and not the Vision Mac-Mini.

The State-Observer is the brain of the vehicle. It keeps track of distance traveled and speed based on the encoders. It also keeps track of position and heading based on data from the PGS, encoders, and LIDAR data. Finally, it issues commands to the motor control module that steer the robot, accelerate, or stop motion. LIDAR data is obtained by a driver module which outputs the Cartesian coordinates of obstacles to a data socket for general use. The module responsible for creating a map of obstacles takes both LIDAR data and vision data from the data sockets and creates a two-dimensional map of obstacles and safe zones. Lines that are detected with the vision computer are included in this map. This data is then used by the route-planning module to create temporary waypoints for the robot to follow in GPS coordinates. The State-observer then uses the waypoint data to control the robot.

6.2 Navigation Course Decision Making

For the navigation course, Roberto will use a version of its route-planning software that emphasizes lane following. The environment is represented in terms of whether a decimeter squared cell is “safe” or “not safe.” On a dynamically sized map, each cell is initialized to a state of “unknown.” As the vehicle acquires sensory data, such as lines that are detected by the camera or objects detected by the LIDAR, cells are assigned a value. A zero means the space is safe whereas a value of 255 means that the space is not safe to travel. The route-planning software module will create a series of temporary GPS waypoints that are deemed safe in between the given GPS waypoints for the Navigation Challenge.

The vehicle will target the given GPS coordinates at the end of the course. If Roberto sees an obstacle blocking its path – either a line or an object – it will obey a wall-following algorithm. The robot is programmed to follow the obstacle boundary until it has a clear straight-line path to the target GPS point.

Upon reaching the target GPS point, the vehicle will set its starting GPS coordinate as the target coordinate and travel back to its original position. However, it will have produced a reliable map of obstacles along the course to the first coordinate. That means the robot will be

able to travel at a faster speed and route paths around obstacles at a faster rate than when it was first discovering them.

6.3 Obstacle Course Decision Making

For the obstacle course, Roberto will use a version of its route-planning software that emphasizes navigating to a given list GPS waypoints, rather than navigating to a single point and back. The way the environment is handled is exactly as stated in the navigation course section, but with the difference that the vehicle will not follow each given waypoint in order. GPS coordinates that are closest to the vehicle and not already attained will be set as the active destination for the robot. The robot will then navigate to those points and use a wall-following algorithm to avoid obstacles or bad areas such as pits.

7. Performance Predictions

Roberto is still undergoing heavy testing of its abilities. Though the full performance of Roberto will not be known until it is run in competition, there are several parts of its performance that can be predicted and tested.

7.1 Speed

Each motor is geared to drive the standard tires at a maximum speed of 4.5MPH. By doing so, safety is assured as the vehicle can not be driven over this speed by the motors. Due to the gear reduction, the motors produce significant torque and will be able to drive the vehicle at or near its top speed in most conditions. Testing has shown that the vehicle is able to reach and maintain its top speed in a straight line.

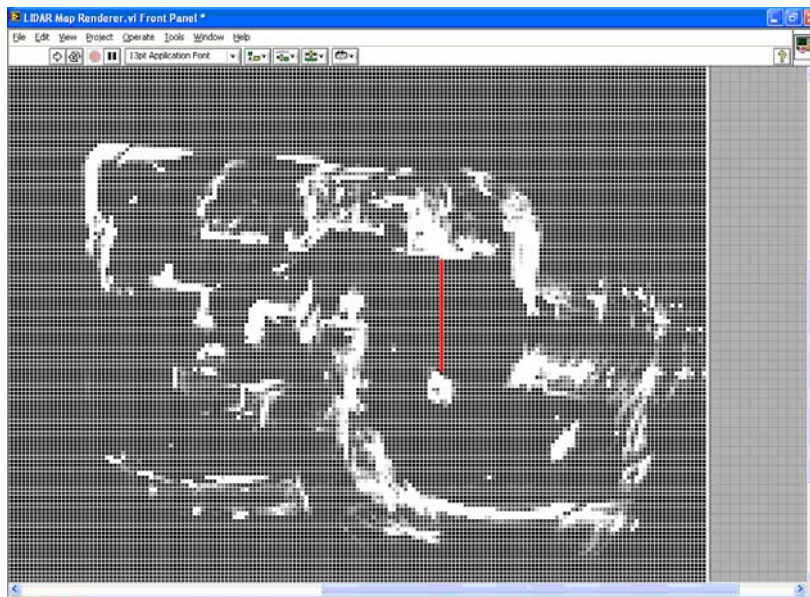
7.2 Battery Life

As components were added, the battery life predictably reduced. While battery life is important to a vehicle in the field, our limitations are much less due to the specific requirements of the test. In our current configuration, our power setup can provide a minimum of forty-five minutes of battery life. This figure was found through high drain tests, with the motors moving at full speed and all components powered and actively working. From this figure, we conclude that our design is sufficiently powered.

7.3 Obstacle Detection Range and Reaction Time

The distance at which obstacles are recognized by the vehicle is set by the software as opposed to limitations of the sensory equipment. For example, the LMS is capable of looking ahead 80 meters, but only objects detected within five meters are recognized in the software. Five meters is our chosen look-ahead for both the vision system and the LMS. After obstacles are detected, their positions are stored so that if that section of course needs to be revisited, those obstacles may be dealt with before they reach the five-meter detection range. Calculations have shown that a five-meter look-ahead will provide sufficient time to react to objects without requiring excessive memory and processor resources.

7.4 Ability to Handle and Negotiate Obstacles



Obstacle Map of Lab Created by Vehicle

Mapping the lab is a good test because distances between walls and objects can be measured and the visual should look much like a blueprint.

As the picture shows, the obstacle mapper does an excellent job of representing the room. Each square represents a 10 cm by 10 cm square, which is represented in a map by a value between 0-255. The important part of the map is the black area, as this is the area marked as navigable. With a tape measure, the highlighted red line was 304.9 cm. In the map, 30 squares are between the barrel and the desk, showing an exceptional amount of accuracy of the map.

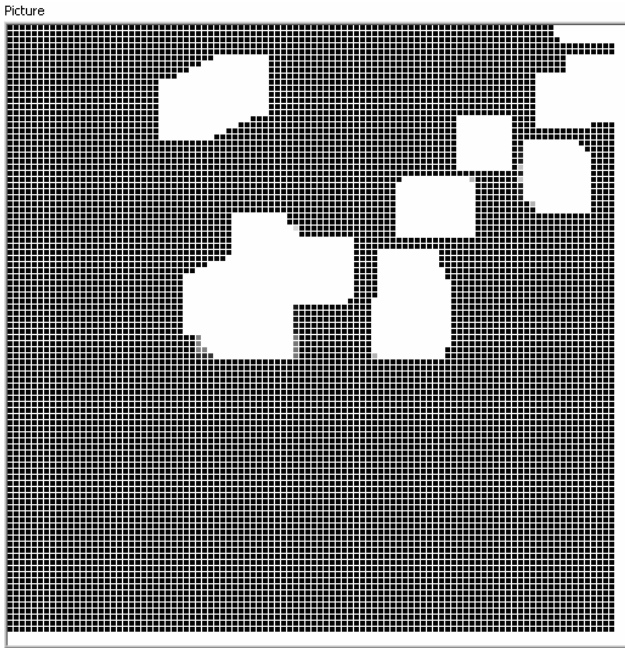
The obstacle mapping code is completely functional and can create an accurate map. While meant to locate obstacles, proving the accuracy of a single object in space would be relatively difficult. To give us more points of interest, a map of the vehicle's lab was created.

Omitted from this picture was the layer that enlarges the obstacle so that a vehicle length is between the black area and the actual point. This was done to make the room more

recognizable. An actual LIDAR map of obstacles would look more like the second picture.

In the second picture, the obstacles have had the buffer area extended around them to ensure that the vehicle cannot contact them. This is what a typical map of obstacles outdoors would look like. The path planner then uses this map to plan a route through the unoccupied part. A sample path planner route segment can be seen in the final picture.

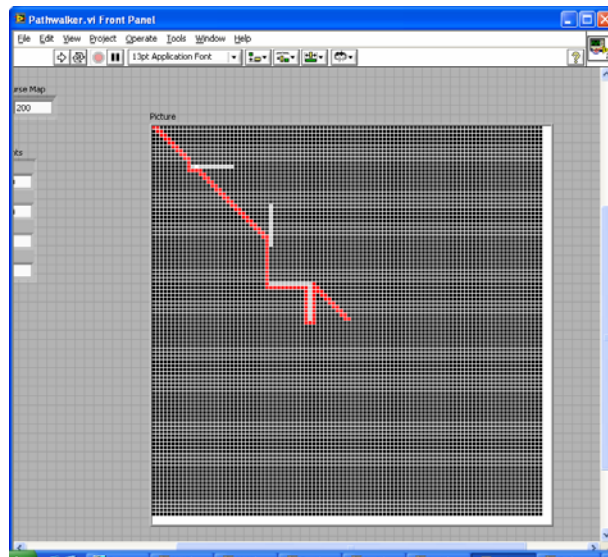
The route is specified in red and takes care to avoid all obstacles. The algorithm is somewhat naïve, but always returns a correct answer. With these walls, the vehicle would collide because of the lack of buffer space. To remedy this, the walls



Map of Obstacles Surrounded by Safety Halo

would need to run through the buffer adding layer seen above. However, the path planning algorithm is the same. Overall, the path planning segment works correctly and provides a collision free path returned as a contiguous set of open segments.

This path connects the vehicle's current location to the destination. Once that destination is reached, the point is removed from the list of waypoints and the vehicle begins moving towards the next point. The queue of points continues to reduce until there are no more points to visit.



Picture of Path Planner with Test Data

While in operation, the vehicle is able to pick up points visually and with the LIDAR unit, place a safety halo around those objects, and plan a path around the objects. Based on the results both of the laboratory mapping test as well as actual testing on obstacles in the field, the vehicle is capable of detecting objects and creating appropriate paths for the vehicle to follow.

7.5 Accuracy of Navigation

For testing and for the competition, Team Case uses a Canadian differential signal with corrections from OmniSTAR. On a clear day, 7 satellites are visible, and the error standard deviation decreases over 40 minutes to approximately 10cm. That means that our GPS positions are extremely accurate over time.

However, GPS can drop out or lose its effectiveness. The use of wheel encoders helps mitigate this risk. The rotary encoder is driven by a timing belt originating from the axle connecting the wheel to the motor. These shaft encoders send 256 pulses per encoder revolution and pulse 413 times total for each wheel revolution (due to difference in shaft diameters). This provides a wheel position accurate to .07 cm if no belt or wheel slippage occurs.

The heading is calculated mainly from wheel encoders with help from the GPS receiver. For the encoders, the change in heading is also calculated with the circle based on excess movement by one of the wheels. The circumference of a wheel's circular path is just the distance between the wheels times pi. Heading change is just the ratio of wheel distance divided by the circular path's circumference times 360 degrees. The heading from the wheel encoders drifts about a degree per meter in optimal conditions, with results varying based on surface conditions.

As mentioned above, GPS readings affect the heading by using "tent posts" to determine heading using the GPS. This is done because the GPS heading is ineffective on the scale of centimeters. Tent posts are accepted as long as the vehicle moves in a relatively straight line for a meter.

The combination of these two systems results in a heading error of about 10 degrees. This is sufficient for the vehicle to navigate with, as the obstacle mapping software is designed to handle erroneous object detection due to heading miscalculations.

8. Team Organization and Time Spent

The senior-project team members listed below invested approximately 10 hrs/wk over the semester. The team leader invested significantly more time. Additionally, two graduate students also contributed their time and expertise.

Team Member	Responsibilities	Total Hours
Jim Piotrowski	Senior project lead, state observation, GPS, vehicle control, mechanical upgrades	400
Neal Heinrich	Laser range-finding, obstacle avoidance, obstacle mapping	200
Luke Wisniewski	Line detection	200
Matt German-Lortie	Camera calibration, obstacle detection with vision	200

9. Budget

Below is a chart that details the cost of the vehicle. It shows retail cost of the component, as well as how much it cost the team. Last year's team spent \$13,250 to construct Roberto. This year's budget is displayed after a listing of the total components used to construct Roberto from the previous year.

Item	<u>Previous Year's Expenses</u>	
	Retail Price (USD)	Cost to Team Case(USD)
Vehicle base donated by Invacare	Est. \$3000	\$0
Bosch aluminum framing and components	\$500	\$0
Curtis 1228 motor driver (x2)	\$182 each	\$0
Sealed 12V AGM batteries (x2)	\$60 each	\$0
Custom machined battery box, encoder disks and sensor mounts, frame mounts	\$1,400	\$0
SICK Laser measurement system LMS 221	\$6,868	\$0
Sick RS-422 to Ethernet converter	\$500	\$0
Laptop: HPT4200	\$2,300	\$0
Matrix DX-7811HRS Camera	\$140	\$0
Camera Lens	\$40	\$0
KWorld DVD Maker USB capture device	\$60	\$0
NovaTel ProPak LB-Plus DGPS	\$5,490	\$0
OmniSTAR differential signal	\$800	\$0
PNI TCM 2.5 electronic compass	\$769	\$0
RCT Remote Switch System 1 wireless E-stop	\$410	\$0
National Instruments USB-6008 (x2)	\$145 each	\$0
RS-232 hub	\$199	\$0
Network router	\$43	\$0
Misc electronic components (connectors, fuses, wire, E-stop, wheel sensors, etc)	Est. \$200	\$0
100-Amp breaker & breaker/distrib panel	\$150	\$0
Power converters (24VDC to 12VDC, 9VDC and 120VAC)	\$60	\$0

Team Case

Roberto

Grand Total		\$20,023	\$0
	<u>This Year's Expenses</u>		
Item		Retail Price (USD)	Cost to Team Case(USD)
Mac Mini Computer (x2)		\$800 each	\$1,600
2 GB SODIUM Memory Modules (x4)		\$70 each	\$280
Wheel Encoder (x2)		\$124 each	\$248
USB Hub		\$40	\$40
RS422 to USB Converter		\$80	\$0 (loaned)
Firewire-based Camera		\$450	\$450
4mm Camera Lens		\$150	\$0 (loaned)
Additional Bosch aluminum framing and other mechanical components		\$100	\$100
USB Wireless Gamepad Controller		\$40	\$0 (loaned)
National Instruments USB-6210		\$499	\$499
Grand Total		\$3,487	\$3,217

10. Acknowledgements

The team would like to thank Drs. Wyatt Newman and Robert Gura for advising them throughout the semester. The continued support of Professor Newman has been invaluable. Graduate students Scott McMichael and Bradley Farnsworth graciously donated their time and expertise throughout the semester to help answer questions. The Case Alumni Association provided much needed funding for the project. Appreciation also goes out to Novatel, OmniSTAR, SICK and Invacare for providing goods and services at greatly reduced or no cost.

11. Conclusion

Roberto has been given many upgrades to its hardware and software over the past year. The changes that have been implemented make the vehicle more responsive to the environment and better able to navigate the challenges in the competition. All mechanical, electrical and software components have been constructed modularly and have been interfaced successfully. The design also incorporates an attractive array of sensing and control components. The robot has proven to be a useful platform for research this year and is ready to perform at the 15th IGVC.