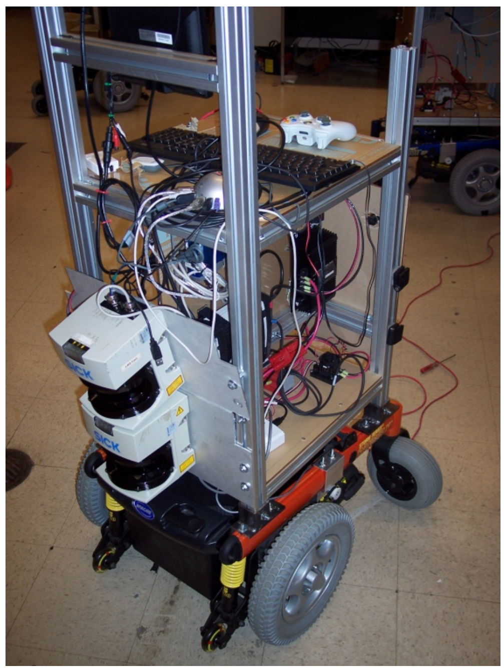
CASE

CASE WESTERN RESERVE UNIVERSITY



# Harlie

Faculty Statement: "Harlie" was designed and built by our student team in an undergraduate class in mobile robotics during Spring 2008.  For this substantial design effort, the students received 300-level credit.

# 1 Introduction

Case Western Reserve University is proud to present for the first time, Harlie, our entry into the 2008 Intelligent Ground Vehicle Competition. Harlie is a redesigned version of Team Case's previous entry, Roberto, which was used in the $14^{th}$ and $15^{th}$ annual competitions.

The new hardware features a better frame design and a safer electrical system. New sensors were also added to address all of the obstacles in the competition. Finally, the software architecture and components were redesigned to be robust and modular.

# 2 Design Process

The decision to completely redesign our last entry was made only at the beginning of the year. Since then, our team has had a large amount of planning and design to complete in a short period of time in order to have a platform capable of performing in this year's competition. Structural, electrical, and software architecture had to be redesigned due to age and original design flaws.

The method we chose for all aspects of the design was a modified waterfall method that included backwards looping if necessary. The waterfall method started with requirements analysis in order to decide what features are necessary and what features would be beneficial to incorporate if possible. The next stage of the waterfall method was to create a design for the system based on the requirements found in the previous stage. After an appropriate design was agreed upon, we then moved to the implementation stage in which the design is carried out into a product. After the implementation stage, we entered the validation stage to rigorously test our design in order to find any flaws. After the validation stage we moved into the maintenance stage in which the design is kept relevant and working properly. The modification to the waterfall method allowed us to move from one stage to any previous stage at any time if the team found it necessary.

The reason for choosing this method was due to the accelerated schedule with which we had to work. A major criticism of this approach is the inability to foresee every requirement in the initial planning. However, since we had a previous entry off which to base our design, we found that this method suited our needs.

## 2.1 Structural Design

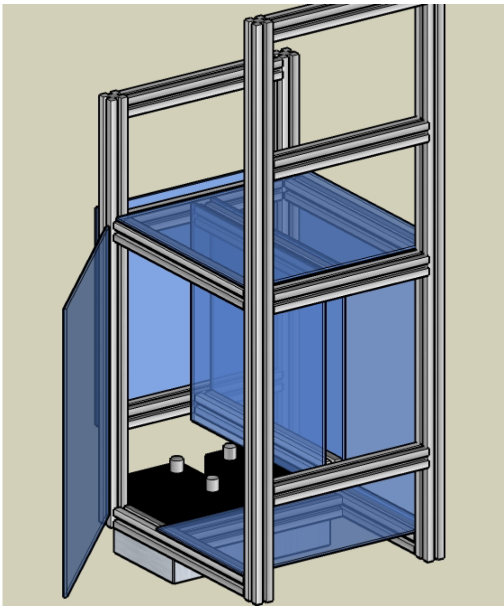


Figure 1: 3D model of frame design.

The basic shape and structure of Harlie is very similar Roberto. Both platforms were constructed using Bosch framing to allow for a flexible layout and for rapid prototyping. Both platforms also used 1/4$^{th}$ in. acrylic sheets for mounting components and for water proofing. The major difference in the design is how the panels to which the components are mounted are laid out on the frame. Roberto's panels are stacked on top of each other to make the design compact. However, this design also made maintenance very difficult due to the lack of space needed for tools between each panel. In the new design, our team opted for one horizontally mounted panel and two vertically mounted panels that can be accessed from either side of the robot. Also new to the design was the addition of doors. The doors allow for the components to be safe from weather while still allowing easy access to the hardware components.

## 2.2 Electrical Design

The electrical design mainly focused on safety. One of the problems addressed was the power distribution. The previous design used open metal bus bars to provide power to components. Electrical shock, short circuiting and accidental overpowering of delicate electronics can result if these components are left unprotected. The new design now includes bus bars that are protected from accidental contact and have more fuses that will aid in surge protection.

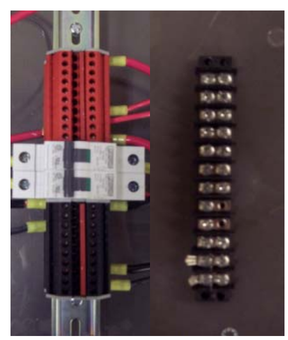


Figure 2: New terminal blocks (left) and old bus-bar (right).

## 2.3 Software Design

The reason for a software design was due to unfamiliarity with the old design and a lack of maintenance of the old software. The final design that our team has settled on can be seen in Figure 3: Software architecture.

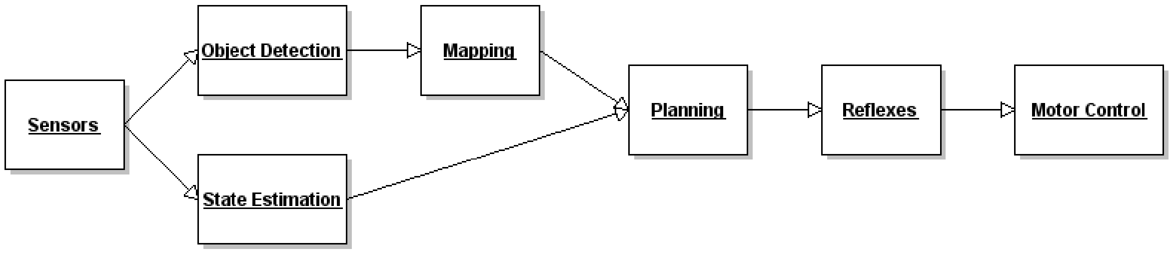



Figure 3: Software architecture

The communication between each component of the design is handled through Datasocket server, which is built into LabView programming language provided by National Instruments. Taking advantage of this prebuilt communication helped our team quickly develop our software and removed a possible weak point in our software.

# 3 Mechanics

The base of our robot was built on a pre-built wheel chair base. The upper frame is made from Bosch aluminum framings that were machined to our specifications. Custom mounts were manufactured in order to attach the aluminum frame to the base.

## 3.1 Base

Figure 4: Wheel chair base provided by Invacare.

The base is from the Ranger X model wheel chair from Invacare's Storm series wheel chairs. The wheelchair base provided our team with a rugged base equipped with suspension and motors. The suspension system will help to stabilize the upper frame unit to reduce noise in sensor values, which is a desirable quality for an outdoor robot. Also, since it is a wheelchair base designed for patient care, the base and motors are manufactured so that there is little if any room for failure. The motors provided by the manufacturer are also limited to a 5 mph max speed, which fits our needs perfectly.

## 3.2 Frame

The frame was designed and built using 4 cm square aluminum framing. The frame was built to support one horizontally mounted panel and two panels vertically mounted in opposite directions. The design was chosen to maximize accessibility and surface area. Designing a method for mounting the frame to our base was one challenge our team had to overcome. As manufactured, the base was sloped down towards the casters. Using a simple mount would cause our frame to also lean at this angle. To overcome this problem our team developed custom mounts that would allow us to level the frame on top of the sloped base. Having a level frame made mounting the LIDARs easier since they were required to be mounted parallel with the ground.

# 4 Electrical System

Harlie's components have many power needs. The LIDARs require a reliable and clean 24V with a low current draw to operate quickly, but the motors require a high current at 24V. The Mac Mini computers require a special 18.5V and other sensors, such as the GPS, require only 12V. To provide this power, the wheelchair base also came with two easily replaceable batteries that are connected in series to give 24V. The base also provided an external port for charging the batteries, making recharging easy.

To generate the other power required for the robot, we used a Carnetix CNX-2140 DC-DC converter to generate 18.5V for the Mac Minis, a Samlex 24-12 DC-DC converter to generate 12V, and a custom-made 24V-24V regulator that produced regulated clean power for the LIDARs. To distribute the power to multiple components, each voltage was fed to a color coded bus bar on separate panels. In our convention, red signifies 24V, blue indicates 12V, and black is used for all grounds. By keeping wiring and bus bars to this coding scheme, our team was able to avoid the costly error of providing sensors with too much power.

For surge protection, our team used a number of different thermal circuit breakers. We used one 120A main breaker that is also used as a main power switch. We also used two 63A breakers, each supplying power to different wheels. For individual electronics, we used small 10A breakers that integrate with the style of bus bar that we chose. A diagram of the power system is shown in Figure 5: Circuit diagram for the electrical system.

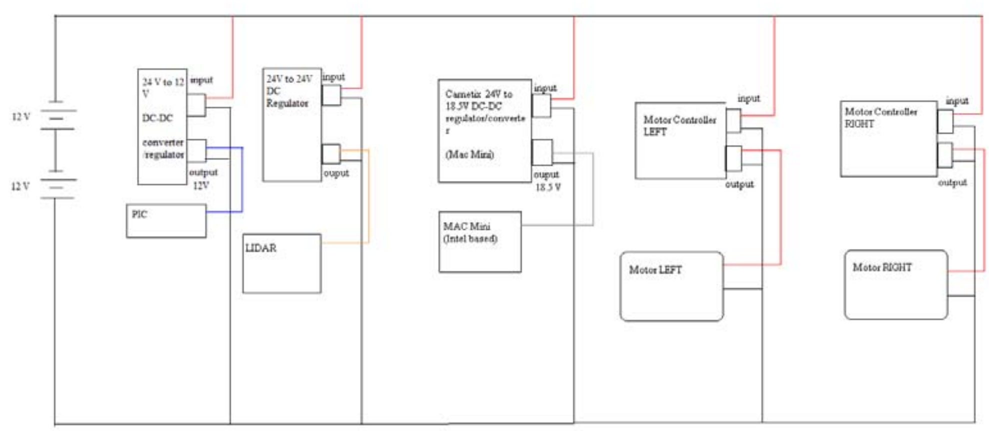



**Figure 5: Circuit diagram for the electrical system**

The Carnetix and Samlex power supplies were both purchased items that would give us the desired voltage and had reasonable power capabilities. The 24V-24V regulator, however, was custom-assembled by our team. At the heart of the regulator is a store bought regulator that has a range of outputs.  Also, since the regulator used is a switching regulator, we needed to filter the output to make sure it was clean power.

# 5 Sensors and Processing

For this challenge, the robot needs to be able to understand where within the GPS coordinates it is, what its heading is, and where obstacles and lines are relative to it.  All three things can be very difficult and require good sensors and sensor processing.  Based on our requirements analysis, we decided that the sensors we need are three Firefly MV cameras, two SICK LIDARs, two wheel encoders, one GPS, and one MEMS style IMU.  Each of these sensors can, by themselves or by being coupled with each other, provide all of the information we need to complete the tasks provided.

## 5.1 Cameras

To best detect obstacles hidden from the LIDARs and to detect the painted lines for the navigation challenge, our team decided to use three cameras.  The chosen camera is the Firefly MV camera made by Point Grey Research.  The cameras communicate to the computer via a standard IEEE-1394 fire wire cable.  We chose to have two of the cameras pointed at the ground to the sides of the robot to be used only for detecting lines.  The third camera is a front-facing camera that will be used to detect lines in front of the robot and objects that are hidden from the LIDARs such as rails and potholes.

The camera processing is done by color image segmentation.  To find lines, the image is segmented based on white pixels.  The results are then processed by a series of erosion and dilation filters to remove unwanted objects.  The results from the filtering are then passed to a particle analysis tool provided by National Instruments vision processing toolbox.  From the particle analysis the objects are classified and their information is passed to the mapper, where the information is put into an occupancy grid to be used for planning.

## 5.2 LIDAR


Obstacle detection is primarily done through the use of two SICK LMS range finders.  The LMS (Laser Measurement System) scans over a range of 180 degrees with ½ degree resolution and returns distance accurate to 10cm with a range of 80m.  The scans from both LIDARs are obtained via a RS-422 to USB converter.  Use of the RS-422 to USB adapter allows it to communicate at a rate of 500kbaud.

One LMS is mounted so the scan will be parallel to the ground. This range finder is used for object detection. To convert scan points to obstacles we iterate through each scan and use a modified RANSAC method to group the points into lines and discarding outliers. The obstacles are then sent via Datasockets to the mapper for use by the planner. This LIDAR is also used for position localization. By looking at successive scans, we can deduce the position offset by using the RANSAC method and a least-squares fitting technique to generate a best guess of the position offset. This information is then relayed to the physical state observer to be included into the Kalman Filter.

For the second LMS we used a hybrid LIDAR which is angled down towards the ground at approximately 20˚ in order to intersects the ground 10 feet in front of the robot. A hybrid LIDAR returns the reflectance at each scan point along with the distance to that point. We can use this LIDAR to not only detect low-lying obstacles but also lines painted on the grass. The usefulness of the hybrid LIDAR is still under investigation.

## 5.3 Wheel Encoders

To aid in position localization, we use one quadrature wheel encoder for each wheel. Quadrature encoders have two output pulse trains that are out of phase with one leading or lagging the other depending on the direction of motion. The signals produced are then integrated on an NI-DAQ 6211 board in real time without the intervention of the main computer's CPU. The NI-DAQ boards were generously donated by National Instruments. The encoder count can then be read from the device through a LabView vi. This information is then sent to the physical state observer to be used as error feedback in the closed loop control and in the Kalman Filter.

## 5.4 GPS

For determining our global position we used a NovaTel ProPak LB-Plus Differential GPS. We used Canadian differential signals with corrections from Omnistart. When we have a clear view of the satellites, the GPS is reported to be accurate within 10cm. Actual test data shows that the estimate is only true in open fields on a perfectly clear day. The GPS signal can be thrown off by nearby buildings, trees, puddles, and clouds. Some of these effects can actually bias the signal in a single direction which violates many assumptions made by the Kalman filter used and can cause serious problems when trying to accurately determine position.

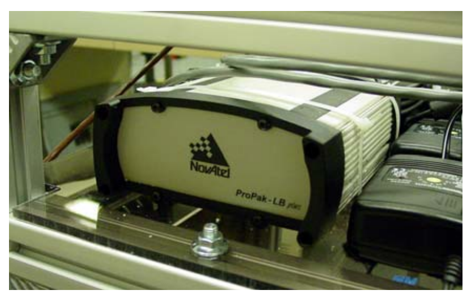



Figure 6: NovaTel DGPS

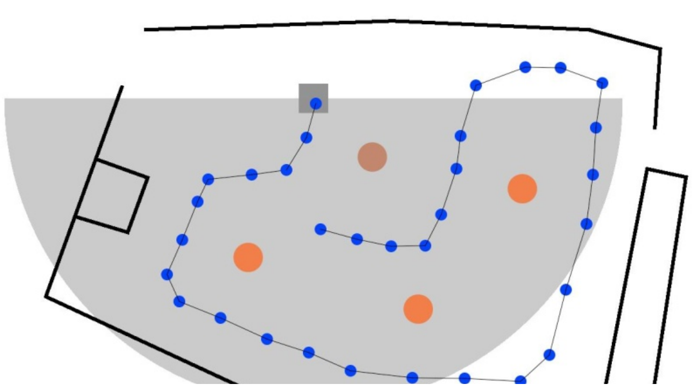


Figure 7: Breadcrumbs made by planner in simulation.

# 6 Planning

The planner is responsible for using the information received from the map provided by  the mapper and goals designated by the user to create a plan that can be carried out by the vehicle controller.  The algorithm used for both challenges is the bug algorithm which plans a straight path to some goal point.  If it detects an obstacle in its path, it will follow the boundary of the object in a clockwise direction until it intersects its original straight line path.  It will then break away from the obstacle and continue on the straight line path to the goal and repeat if necessary.

In the obstacle challenge, the goal points are well defined, allowing the bug algorithm to work quite well.  However, some modifications need to be made for the navigation challenge.  In the navigation challenge the goal points are not as well defined, causing the robot to have to artificially create goal points ahead of it to keep it moving along the path.  Also in this challenge, we are treating incoming lines as obstacles so that the robot will not go through them.  This along with the bug algorithm should succeed in getting the robot to the goal.

The plan the robot creates is in the form of a breadcrumb list.  A breadcrumb is a GPS point coupled with a speed that the robot should be going at that point.  The robot then follows the breadcrumbs until it reaches its goal.

# 7 Steering and Control

To provide smooth control of the wheel speed a closed loop PID controller is used.  PID stands for proportional, integral, and differential.  Each of these three types of gains is applied to the error of the commanded wheel speed to the current wheel speed and can be adjusted to create smooth control.  We chose our gains to give us a quick response, low or no overshoot, and to have an adequate margin of stability.


The motor controllers used were two Victor 885s.  The Victors provide up to 120A peak current and use a PWM signal generated by the NI-DAQ 6211s.  We chose these motor controllers because of their great performance for a relatively small price.

The steering algorithm we used is called the "Wagon-Handle" method.  The method simulates how the robot would behave if it were a wagon led by a 1 meter wagon handle.  As long as the handle is forced to be somewhere along the breadcrumb train, we can guarantee that the robot will also follow the path and thus avoid the obstacles.  This method also reduces the problem of determining appropriate wheel speeds to simple geometry.  These wheel speeds are then sent to the controller which is able to quickly respond to the desired wheel speeds.

# 8 Physical State Estimation

Physical state is one of the most difficult problems in robotics. A good estimation of the position of the robot is vital to almost every other higher level system on the robot. We use a GPS solution that on a clear day is accurate within 10cm. However due to atmospheric effects, water on the ground, and nearby buildings, GPS is not always very accurate.

Kinematic modeling and state estimation can be added to pure GPS to provide some sort of feedback such that the state is reasonable for a longer period of time. Wheel encoders can be used to measure the kinematic movement of the system. In general, the best kinematic models under ideal conditions of minimal wheel slip are only accurate between 5% and 10% of the total distance traveled. Kinematic estimation is still considered open loop – but is much better than dead reckoning alone.

Since a variety of sensors are available for use in state estimation and each sensor has its own downfall, the Kalman Filter can be used to probabilistically determine the most accurate state estimate. Moreover, the Kalman filter will estimate the state recursively and iteratively in real time – constantly driving the uncertainty of the solution downward. The Kalman Filter takes inputs from wheel encoders, GPS, and laser odometery allowing us to determine our position within 20-30 centimeters. With proper analysis and tuning of the Kalman Filter gains, we hope to get our estimation down to within 10 centimeters.

# 9 Expected Behaviors

## 9.1 Speed

The speed of the wheel chair motors as defined by the manufacturer is only up to 5mph. To make sure that our controllers do not exceed the limit, we verified the restriction. To verify, we commanded the wheels to run at max speed and counted the number of revolutions per second. We then determined the max speed of the robot to be just less than 5mph, which is within the rules stated. We also verified the speed using the wheel encoders.

## 9.2 Battery Life

The typical battery life for the robot with all of its sensors running is typically around two hours. This duration should be long enough for each heat of the competition.

## 9.3 Complex Obstacles

Using the bug algorithm, the robot is guaranteed to get to its goal if a path exists. We do not anticipate that complex objects such as switch-backs, center islands, and dead ends will cause any problems during planning; however, testing and verification have not yet been completed.

## 10 Parts

| Part Description | Retail Price | Cost to team |
|---|---|---|
| Wheelchair base | Est. $3000 | $0 |
| Bosch Aluminum Framing | $500 | $500 |
| Power Converters | $345 | $345 |
| Electrical Components | $770 | $770 |
| Victor Motor Controller | $200(x2) | $400 |
| SICK LIDARs | ~$6000(x2) | $10,000 |
| Mac Mini Computers | $800(x2) | $1600 |
| Wheel Encoder | $150(x2) | $300 |
| RS422-USB Converter | $80(x2) | $160 |
| FireFly Camera | $350(x3) | $1050 |
| Camera Lens | $150(x3) | $450 |
| NI-DAQ 6211 | $800(x2) | $0 |
| NovaTel ProPak DGPS | $5490 | $2700 |
| **Total** | $24,835 | $18,275 |

## 11 Conclusion

Harlie is Team Case's newest robot and our entry in the 2008 Intelligent Ground Vehicle Competition. The new design incorporates new ideas and fixes flaws in the old system. These changes have been proven to make the platform safer and more durable. Our new software should also provide Harlie with the intelligence needed to complete all tasks in the competition, making us good competitors for this year's competition.