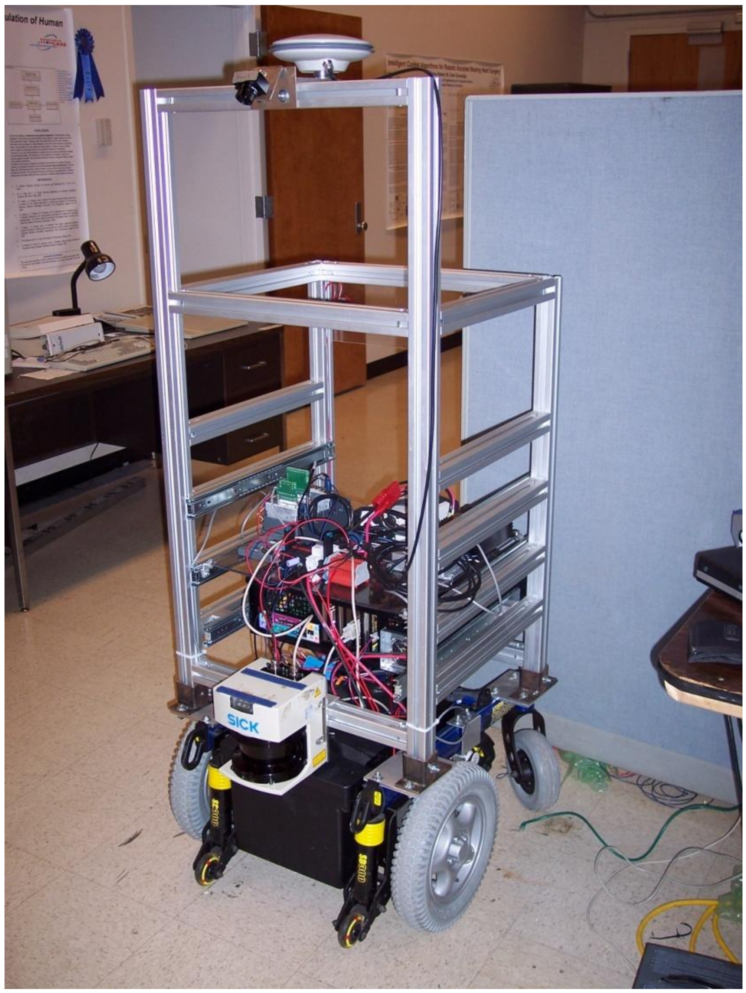


ALEN

Faculty Statement: “ALEN” was designed and built by our student team over the course of a year. For this substantial design effort, the students senior project and independent study credit.

1 Introduction

After experience from the 16th IGVC competition, students from Case Western Reserve University began working on ALEN to assess the flaws found in the design from previous years. Although the structural and power designs are similar to Harlie (also a competitor in this year's competition) there are crucial changes to the hardware and software designed specifically for the competition objectives.

It is important to note that students working on the ALEN project participated in the 16th Annual IGVC competition (Harlie). The team kept the electrical design, software architecture, and structural design (with minimal changes) when designing ALEN. The significant effort put forth in this project focuses primarily on the software components of the vehicle including vision, physical state estimation, etc.

2 Structural Design

The basic shape and structure of ALEN is very similar to both Roberto (14th and 15th year IGVC competitor) and Harlie (16th and 17th year IGVC competitor). All platforms were constructed using Bosch framing to allow for a flexible layout and for rapid prototyping. Both platforms also used 1/4th 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 last year's competition, our team opted for one horizontally mounted panel and two vertically mounted panels that can be accessed from either side of the robot with Harlie. 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. This design was flawed however because components on the robot were overheating because of lack of air flow. Although ALEN's design still keeps the same frame as previous work, it is modified to mimic a rack server where each subsystem of the hardware is located on sliding horizontal shelves. It is modified to be slightly wider than the original design of Harlie to handle mounting the rack server used. This not only opens up space inside the robot for air flow but it allows for easy access to any of the components by removing a shelf to replace or modify hardware. The decision for this design is explained a little further in the electrical design section.

2.1 Electrical Design

Harlie's electrical design from the last competition proved successful for our safety standards and thus the electrical system was ported to ALEN and designed with a system of bus bars that are fully protected like those shown in figure 1. Secondly the Electrical system is designed for modularity by having several "drawers" that are electrically hot swappable, allowing quick and easy diagnostics of the robot without the worry of causing damage to component on a given drawer when removing it.

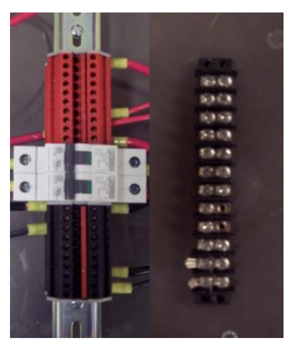


Figure 1: New terminal blocks (left) and old bus-bar (right).

2.2 Software Design

Because the team who designed ALEN competed in last year's competition with Harlie the software architecture of ALEN stayed the same (Figure 2) with changes made to the communications between the components.

Figure 2: Software architecture.

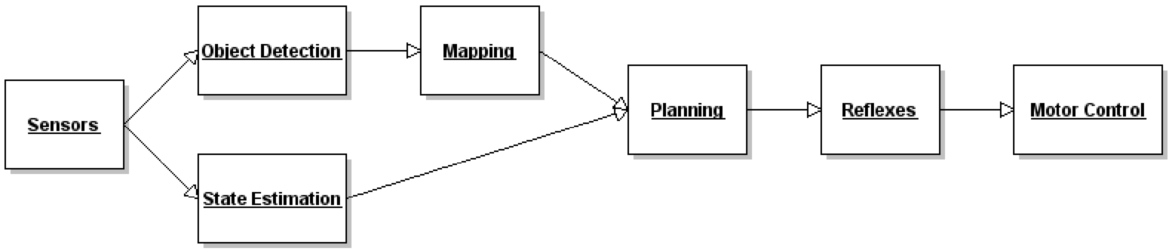


Figure 2: Software architecture

Previous years have used what is known as Data Socket (National Instruments built in communication in LabVIEW). The Data Socket server is the quickest way to have communications, yet we have found it to be a weak point in the field. The server locks up at random, misses packets, and crashes the LabVIEW software making for a timely repair each time. The communication between each component of the design of ALEN is handled through a series of UDP commands to send and receive UDP datagram messages over ALEN's local IP network. The design allows for ALEN to have a stable communication protocol for its needs.

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

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

Figure 3: Wheel chair base provided by Invacare.

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 the hot swappable shelves and the rack server for high level computations. The design was chosen to maximize accessibility and surface area. 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 last year developed custom mounts that would allow us to level the frame on top of the sloped base. Having a level frame made mounting the LIDAR easier since they were required to be mounted parallel with the ground.

4 Electrical System

The LIDAR requires a reliable and clean 24V with a low current draw to operate quickly, but the motors require a high current at 24V. The main server computer would require 24V 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, a Samlex 24-12 DC-DC converter to generate 12V, and a custom-made 24V-24V regulator that produced regulated clean power for the LIDAR and National Instruments cRIO. 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 4: Circuit diagram for the electrical system.

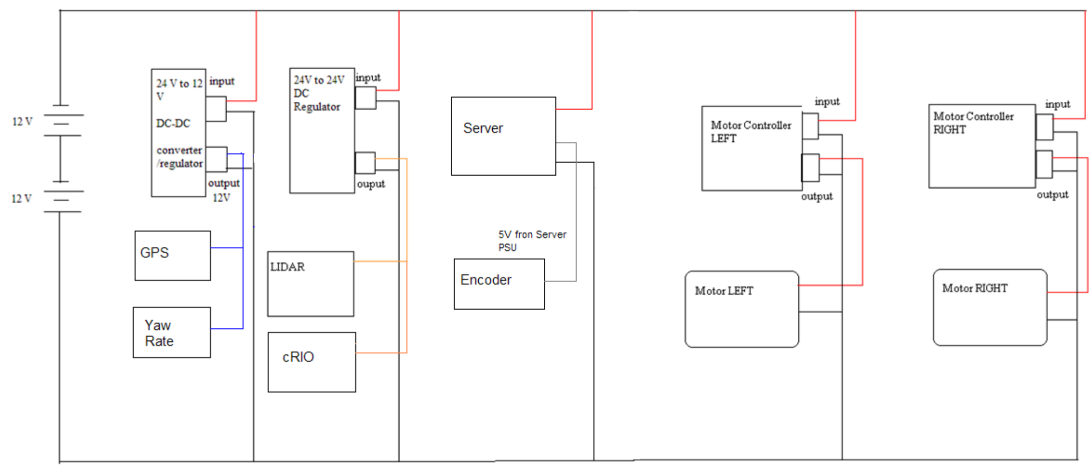


Figure 4: Circuit diagram for the electrical system

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 one SICK LIDAR, two wheel encoders, two motor shaft encoders, one GPS, and one yaw rate sensor. 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. Like the previous design, 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. An addition to ALEN is the use of stereovision. We found from previous years that obstacles like a-frames are hard to detect using LIDAR. Figure 5 clearly shows that stereovision allows for the robot to easily detect obstacles like these to avoid them safely.

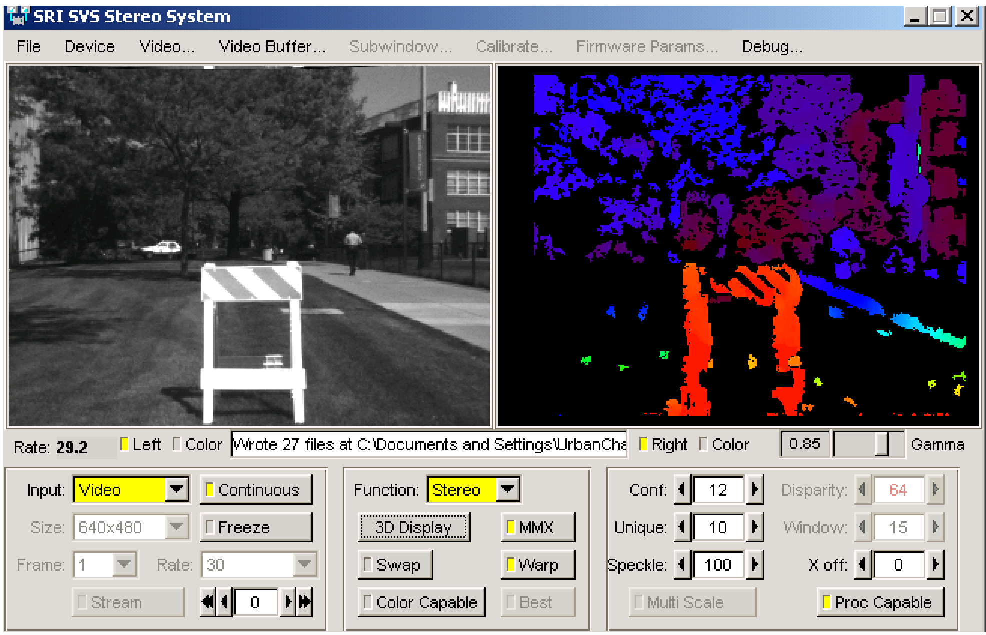


Figure 5: Stereovision's use in detecting obstacles that LIDARs fail to observe.

The camera processing is done by categorizing. An algorithm is set to categorize sections as lines, the image is segmented based on categories. The results of the algorithm are shown below in Figure 6. These results comprise one layer to an occupancy grid used for mapping describing all the lines as obstacle or avoided sections of a grid.

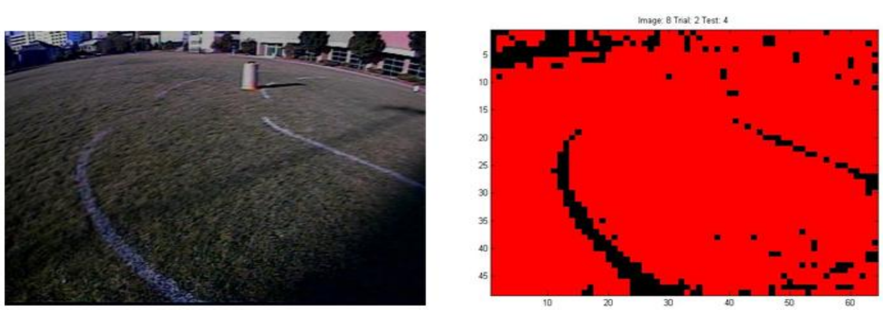


Figure 6: View from Camera(left) and Binary image depicting "seen" white lines (Right)

5.2 LIDAR

Obstacle detection is primarily done through the use of a SICK LMS range finder. 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 connected directly to the linux server allowing for communication 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 UDP 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.

5.3 Yaw Rate Sensor

The yaw rate sensor is used in order to improve on one of the most difficult aspects in localization; the determination of the heading. The yaw rate sensor is a MEM device from Analog Devices that outputs an analog voltage which corresponds to the angular velocity in degrees per second. This output is sampled at a rate of 1kHz giving measurements ever millisecond. However this voltage will need to be converted to an angular velocity. Thus it must be determined what the volts per degree per second are. In this situation this will be in mV/deg/s. Thus by dividing the output voltage by the mV/deg/s the angular velocity (deg/s) can be obtained. This value can then be integrated in order to provide a heading.

The PCB design takes into account several factors about the yaw rate sensor when developed. The choice to have the PCB accept the yaw rate development board is because of the ease of soldering in conjunction with the multiple yaw rate sensor development boards already in place waiting to be used on other robots. The main goal of the PCB is to have a sustainable 5 V input to the yaw rate sensor with separated digital and analog supplies. This is accomplished through two 5 V linear regulators that keep the yaw rate input voltage at a steady 5 V despite any minor voltage changes in the battery supply of the robot. This is important as the input voltage tolerance of the yaw rate sensor is ±.25 V. The second major goal was to have the ability to adjust the null voltage. This is accomplished through a potentiometer which adjusts the bias current into the yaw rate sensor in order to center the output voltage when not moving, thus giving the yaw rate the ability to detect rotation equally in either direction. This means that since the yaw rate sensor outputs in the range from 0 V to 5 V, the bias current could be supplied to be sure that when sitting still the yaw rate sensor outputs 2.5 V. Finally the board needs to be designed in a way that eliminates as much noise as possible since this is an analog system. This is accomplished in several ways. One is having each output be twisted with ground inside the cat5e cable in order to create a shield for the transmitted data. Secondly the RJ45 connector is shielded and separate ground planes are made for the analog and digital section of the sensor. This setup should vastly improve the noise characteristic of the received data in the cRIO from the yaw rate sensor by eliminating any additional noise from outside sources.

Due to bias updates that will be fed to the PSO (heading corrections based on vectors from GPS) the yaw rate sensor will provide differential or delta measurements. These are measurements that are based on the previous state of the PSO. This means that the yaw rate will provide the change in heading from the previous cycle. Since the PSO and yaw rate sensor will be nearly synchronous the yaw rate should be providing the change in heading from the previous state of the PSO.

In order to handle the bias and drift of the yaw rate sensor, it will be sampled at a rate of 1 kHz and then the bias we be updated via a feedback system based on the change in heading as described by the encoders in the model of the PSO at a rate of 20Hz. This will ensure that yaw rate does not drift out of control and continue to provide reliable information to the PSO. However it is important to not to provide too much feedback to the yaw rate sensor as this could cause the yaw rate sensor to fully believe in the encoders and provide no more reliable data than the encoders do alone. This topic will be further discussed in the results section when an appropriate feedback gain for the system is determined. This feedback system also handles changes in temperature that may occur causing the bias to skew.

5.4 Quadrature Encoders

Quadrature encoders are used to develop the model for the PSO to be used in the Kalman equations. This is further explained in Error! Reference source not found. section. The encoders' use two signals, an A channel and a B channel which can be used to determine in the robot is moving forward or backward. When the A channel leads the B channel the robot is said to be moving forward. Alternatively if the B channel leads the A channel then the robot is said to be moving backward. This can also be used to determine heading by finding the difference between the right and left encoder counts.

5.5 Global Position System

The global positioning system, or GPS, is a sensor that uses multiple satellites to compute an absolute global position. The receiver will receive signals from at least four satellites and uses geometry to solve for the position. The GPS system used in this project is capable of getting measurements accurate to within a tenth of a meter if it has a clear view of the southern sky.

However, the measurement can be unreliable when there is a significant amount of moisture in the air, if the receiver is under a tree, or near a building. From previous work with the GPS unit, the measurement has been observed at varying by ±5m. The receiver also has moments of "drop-out" where it gives no measurement at all. This can occur if the receiver does not get a good signal from at least four satellites and is common.

For determining our global position we used a NovaTel ProPak LB-Plus Differential GPS. We used Canadian differential signals with corrections from Omnistar. 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.

6 Planning

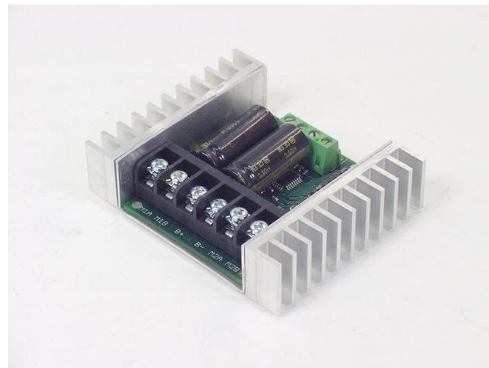
The planner is responsible for using the information received from the map provided by the mapper and goals designated by the user to create a plan that can be carried out by the vehicle controller. The algorithm used for both challenges is the bug algorithm which plans a straight path to some goal point. 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 navigation challenge, the goal points are well defined, allowing the bug algorithm to work quite well. However, some modifications need to be made for the obstacle challenge. In the obstacle course 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 controller used on ALEN differs from previous versions of the robot. The original design used two Victor 885s chosen for durability and price. However the controllers had difficulty driving the robot because of weight limitations. ALEN is now equipped with a Sabertooth 2X25 Regenerative Dual Channel Motor Controller designed for high powered robots – up to 300lbs allowing for better control of the vehicle when climbing ramps and hills.

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. The wheelchairs also have the ability to spin in place so this was added to the steering algorithm as a “mode”. The planner could call for a spin in place if a sharp turn is needed to avoid an obstacle safely. The planner sends the desired heading and an angular velocity the steering after a complete stop is made will spin until the desired heading is met within some desired threshold. After a spin in place is completed, the vehicle returns to wagon-handle again to complete the path.

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 hoped to get our estimation down to within 10 centimeters.

In last year's competition Harlie's team implemented a previous version of the Physical State Observer (PSO). This version was plagued by constant hardware failures and continuing issues regarding GPS dropout, where the GPS loses communication with its satellites and provides no position information to the PSO. This problem was detrimental to the team's success during the 2008 competition. This implementation included only quadrature encoders as the model and GPS data for model correction.

A robust Physical State Observer was built for ALEN that is able to run for extended periods of time and be reset several times by the user during operation. The Physical State Observer is built with the ability to accept input from any additional sensors that could provide state estimation information via UDP network packets. This was taken advantage of by adding laser odometry (lodo), a method of tracking objects within view of a laser range finder, (a device that sends our a series of laser "pings" in a 180 degree pattern and .5 degree resolution, returning the distance of those "pings") to the Physical State Observer (PSO). (Nemeth & Harper, 2008) This method was used in order to assist the Physical State Observer during times of GPS dropout where the GPS would lose communication with its satellites and provided no information to the PSO.

Under this implementation the PSO was able to determine its position within ±.5m and within ±1° in heading after traveling 100m when using all of the sensory data. In the case of having static objects to track; the PSO with lodo but no GPS was able to determine its position within ±1m and ±2° in heading after traveling 100 meters. In the case of a dynamic environment with moving objects; the PSO with lodo but no GPS was able to determine its position within ±10m and ±5° in heading after traveling 100m. However, the PSO can be further improved by adding an additional sensor. The goal of this project would be to improve the case of PSO in the dynamic environment with moving objects, by cutting the error in half to ±5m in position and ±2.5° in heading. This can be accomplished by using a yaw rate sensor that will measure the angular velocity of the robot. The angular velocity can be integrated in order to determine the heading of the robot which then can be fed into the PSO.

With the integration of the yaw rate sensor into the PSO, the PSO was able to further improve on the case of GPS dropout where the LODO had fell short. The yaw rate sensor produced an accuracy of ±2.5 meters in position and ±3 degrees in heading after traveling 100 meters without GPS.

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	$125	$125
SICK LIDARs	~$6000	$5,000
Server Computer	$1100	$1100
Wheel Encoder	$150(x4)	$600
FireFly Camera	$350(x3)	$1050
Camera Lens	$150(x3)	$450
NovaTel ProPak DGPS	$5490	$2700
NI Compact Rio	$3000	$800
Modules for cRIO	$500(x3)	$1200
Total	$23,930	$14,640

Team Members:

Tony Yanick – Team Lead, MS Student Computer Engineering, EECS Department

Chase Nemeth – Undergraduate, Electrical Engineering, EECS

Beom Koh -- Undergraduate, EECS

Avinash Karamchandani – Undergraduate, EECS