OHM 2.0 Design Report

2014 Intelligent Ground Vehicle Competition

University of Michigan – Dearborn

Dearborn, Michigan 48128, USA

May 14, 2013

Team: Angelo Bertani, Zachary DeGeorge, Yanchen Shang, Eric Aitken

Advisors: Nattu Natarajan

I hereby certify that the design and engineering of OHM for the 2014 Intelligent Ground Vehicle Competition are significant and equivalent to what might be awarded credit in a senior design course and include the development of GPS navigation, vision lane following capabilities and object detection and avoidance using a scanning laser range finder.

____

Dr. Nattu Natarajan

Table of Contents

1. Introduction

The OHM 2.0 project is one of many projects of the Intelligent Systems Club at the University of Michigan-Dearborn. The goal of this project is to design a robot to compete in the 2014 Intelligent Ground Vehicle Competition (IGVC). The team this year strives to improve upon the successes and failures of the previous system. The members of the club will rely heavily on past robotics competitions ranging from Autonomous snowplows and lawnmowers, to the first generation of the OHM platform entered into the 2013 Intelligent Ground Vehicle Competition. Autonomous robots are becoming prevalent in many aspects of modern society and the IGVC, while being in a controlled environment, provides a great example of how self navigating vehicles could be used, such as going into areas that may be dangerous to humans. This report will describe the mechanical, electrical, safety systems, and control systems of the second generation of the OHM platform.

1.1 Team Composition

The OHM team is comprised of four students, with backgrounds in Computer, Electrical, or Software Engineering.

Name	**Year**	**Major**	**Hours**
Angelo Bertani (lead)	Graduate 4/14	Computer/Electrical Engineering	600
Zach DeGeorge	Graduate 4/14	Computer/Electrical Engineering	300
Yanchen Shang	Graduate 8/13	Electrical Engineering	300
Erik Aitken	Junior	Software Engineering	300

All team members have been active participants in the Intelligent Systems Club and share a passion for intelligent robots. Angelo, Zach, and Yanchen are recent graduates of the University of Michigan-Dearborn and are competing this year as a final test of the skills they have acquired through their education. Erik is the newest member to the team and is taking the opportunity to learn from the senior members and looks to carry on the development of the OHM platform in the future.

2. Design Overview

The design philosophy when developing the OHM platform was one of modularity. The process began by identifying the critical systems needed to complete each stage of the challenge presented by the IGVC.

The team has identified that at a minimum at least three major subsystems are necessary in order to have any chance in completing the course. Those three systems have been identified as follows in Table 1

Table 1: Navigation Modules

Requirement	**Component Used:**
GPS Waypoint Navigation	Garmin: **GPS 19x HVS**
Lane Following	Microsoft: **Lifecam HD 120° View**
Obstacle Avoidance	SICK: **LMS 111 270° Scan**

Table 1 also shows the devices that is used as the main component in completing the modules task. Each module has been developed independent of the others with the final step of the process focusing on integrating all the systems together. Below in Figure 1 a flow chart as an illustrates the process using the three modules identified above.

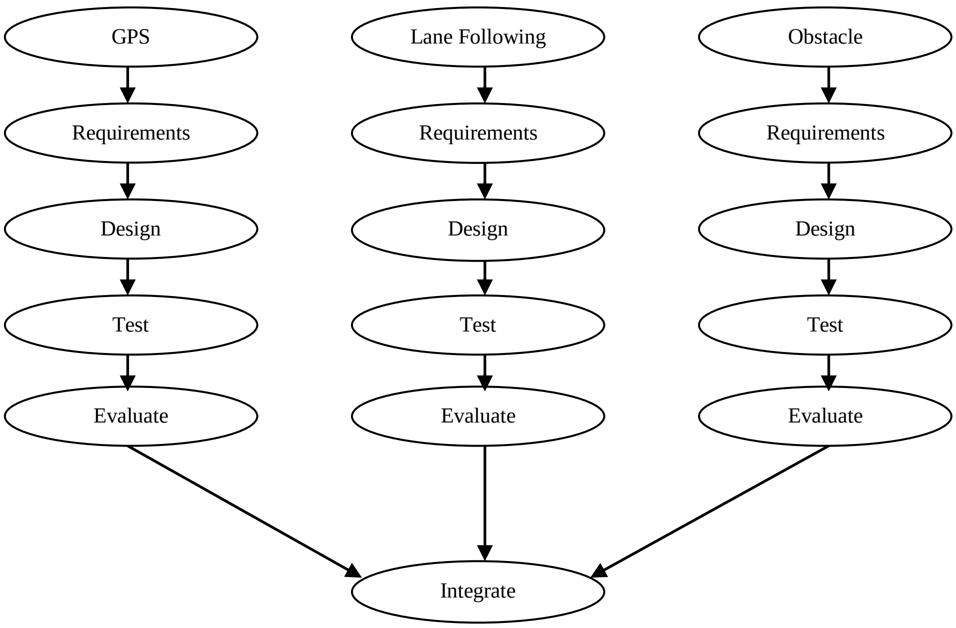

Figure 1: Module Development Strategy

The goal of this process is to ultimately take the information provided by each subsystem integrate into a system which can create a map of the course that can be used for path planning. Additional systems to aid in creating this map and handling critical module failures will go through the same design process and those that are in use will be discussed later in the Intelligent Systems section.

3. Mechanical Design

For this competition OHM will be using a mechanical platform that has been used in previous competitions by the Intelligent Systems Club. This platform is a 3-wheeled design. One of the three wheels is on a caster and the other two are the drive wheels. The robot is steered using differential steering which involves commanding the two drive wheels at different speeds when a turn is desired. This platform was chosen because the overall structure allows for the sensor configuration to be easily changed and the robot itself to be easily repurposed. In fact this same platform has been successfully used in the Institute of Navigation's Autonomous Snowplow competition in 2013 & 2014.

3.1 Chassis

The chassis of the robot consists of three main tiers each made of 19.05mm plywood. The lowest tier has the drive motors attached with a frame where the snowplow can be attached in other configurations. This tier also holds the batteries which power the motors. The middle tier holds the motor controller as well as a separate sensor supply battery. This middle tier also has the caster wheel and safety circuit boxes attached. The bottom two tiers hold all the necessary components that make up the drive train. The layout of the bottom two layers is shown in Figure 2. The last tier of the OHM platform uses another sheet of plywood that holds the computer, cameras, GPS, and LiDAR. This tier is often adjusted to fit the desired sensor suite being used by the current desired use for the robot.

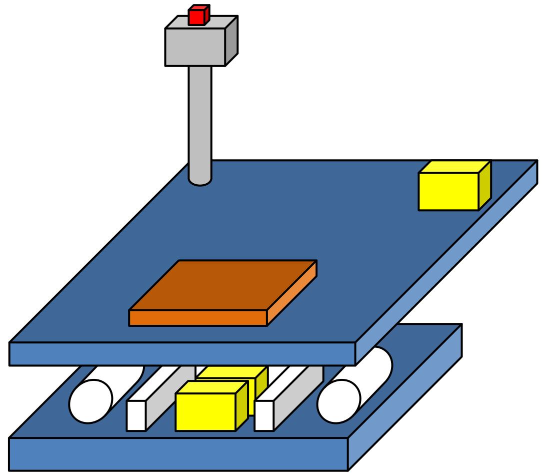

Figure 2: Ohm Platform Layout

3.2 Dimensions

Table 2: OHM Dimensions

Feature	Dimensions/Mass
Chassis Width	0.660m
Chassis Length	0.950m
Chassis Height	1.01m
Wheelbase	0.763m
Battery Weight	19.7kg
Overall Weight	98kg (215lbs)

3.3 Drive Train

The robot is powered by two NPC High Torque 24 VDC electric motors. Each motor is capable of providing approximately 1 horse power at full load. Although this competition should not have the need to exercise the full load capacity of the motors, stress test conducted for the snowplowing configuration have shown that the robot is capable of towing a fully loaded Chevrolet Blazer SUV with high rolling resistance up a 2% grade from an initial stand still. The SUV was towed with engine off and transmission in neutral. Vehicle GVWR is 6500 lb. GVWR was measured by loading the vehicle with 5 passengers and 300 lb. of tools. The robot successfully pulled the fully loaded SUV up the incline at a speed of 2 mph with no issues.

4. Electrical Design

The electrical systems of OHM are designed to integrate the electrical sensors with the mechanical robot in a safe and efficient manner. The sensors run off of either 12 volts or 24 volts so that they can easily be powered from the robots batteries, such as the GPS and SICK LiDAR. 5 volt sensors, such as the wheel encoders and camera, are powered from the computer via USB. The required laptop can also easily be powered off of the batteries by using a simple 12 volt 80+ watt car inverter. Anderson Powerpoles are used to easily make power connectors for all of the different devices.

4.1 Power

Two Optima yellow top batteries are wired in series to provide 24V DC power supply capable of providing 55Ah of power. These deep cycle batteries have a long life, are spill proof and recharge faster than a standard battery. Continuously running the motors at high speeds these batteries provide 2-3 hours of run time under a full load. Since the competition and environment should not put full load stress on the batteries it is expected that the batteries will last at least 4-5 hours. A 4-5 hour battery life should be more than enough time to test and compete in a single day. In the event of a failure spare batteries are on hand.

An additional 12v 15 Ah Werker battery is used as a clean power source to provide power to navigational sensors, an inverter in the event that the laptop needs to be charged while testing, and an onboard router for remote access. Without the inverter running the Werker battery is capable of powering all the sensors for approximately 6-8 hours. However if the inverter is running the battery life drops off significantly to 1-2 hours. The plan for competition day will be to have a fully charged laptop allowing the system to run without the inverter for maximum battery life.

4.2 Motor Control

The robot interfaces with a dual channel Roboteq motor controller to send speed commands to the motors through a serial connection. The robot utilizes differential steering to turn with 2 wheel drive, with the motor on each side of the robot connected to a separate channel output from the Roboteq. To produce values to send to the motors the `x` and `y` axis values of the desired control vector are converted to Speed and Turn values. Equations (1) and (2) are used to calculate the left and right speed values to send to the motor controller. A gain is also applied to the resulting wheel speeds before they are sent to the motor controller.

`Left Speed = (Speed + Turn) * Desired Speed` (1)

`Right Speed= (Speed – Turn) * Desired Speed` (2)

4.3 Safety System

Safety is primary concern when designing these large, autonomous robots. Two emergency stops have been integrated into the electrical system using solenoids and relays so that the power to the motors of OHM can immediately be shut off in the case of an emergency. One emergency stop is located on the rear of the robot, while another is a remote stop that is effective at ranges greater than 50 meters. Below is a schematic of the power and safety systems. In the event that a remote emergency stop is triggered a selflatching relay circuit has been implemented to ensure that power cannot be restored to the robot until the master power has been cycled.

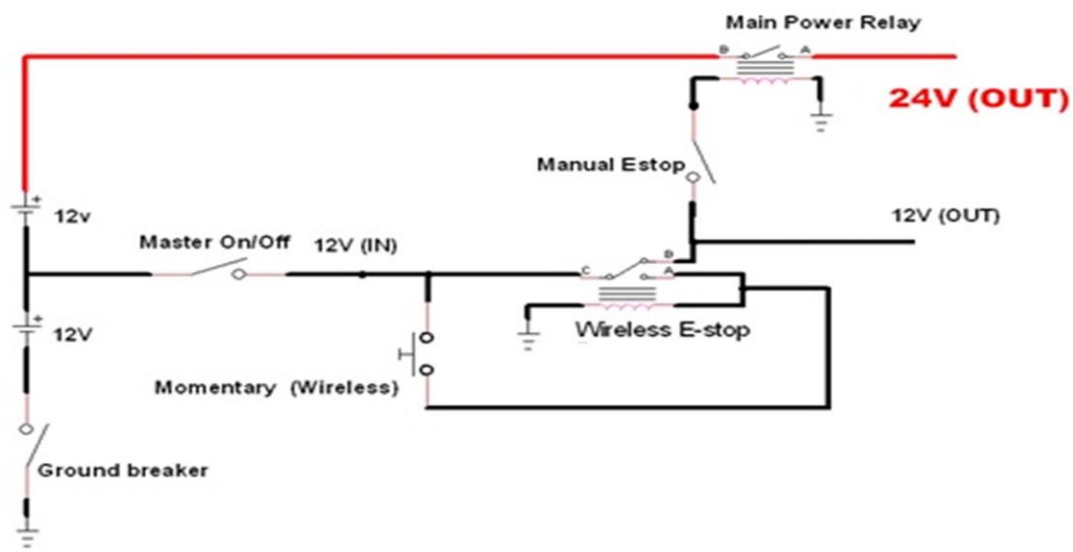

Figure 3 - OHM Power and Safety Circuit

4.4 Sensors and Communication

OHM utilizes multiple sensors in its design, each having their own purpose and communication specifications. Below in Table 3 is a list of all the sensors, their purpose, and the communication protocol that they require.

Table 3: Sensor List

Device	**Purpose**	**Protocol**
GPS	Absolute Localization Heading	Serial
Wheel Encoders	Speed Control	Serial
Compass	Heading	Serial
LiDAR	Object Avoidance Relative Localization	TCP/IP
Camera	Lane Detection	USB Video Stream

4.5 Computer

The processing for OHM is done using a laptop. The reason OHM uses a laptop in place of a typical onboard computer is that it allows multiple team members to independently develop modules and test without the need to worry about onboard computer failure. This setup also allows for the computer to be easily swapped out for another laptop in the event that the first one has a failure. The main laptop used on the OHM setup this year is a 2010 MacBook Pro with an Intel i5 processor and 4GB of RAM. This laptop has been used in several past competitions and has proven capable and reliable in the past both in terms of processing speed and battery life.

5. Intelligent Systems

In order for the robot to be considered an Intelligent System the necessary systems must be integrated and they must be controlled in coordination to create a robot that is capable of completing a given task autonomously. To complete the challenges of the IGVC, the modules identified in the Design Overview section of the report must be developed to complete their desired tasks as standalone pieces. Once the desired task of the module can be completed on its own integration into an autonomous navigating vehicle is the next step. Each module will have its own section outlining the strategies implemented and a final section for integration will describe how the team plans to bring the pieces together.

5.1 Lane Following

The competition course will start by requiring the robotic vehicle to follow a lane painted in the grass outline with white lines. During this portion the vehicle must also avoid static obstacles that try to impede the progress of the robot. In order to effectively locate and stay in the lane OHM will make use of a webcam and the open source computer vision library OpenCV. With the help of OpenCV's image processing capabilities OHM can determine the best route to stay in the lane while avoiding obstacles using computer vision alone.

The vision processing starts by capturing an image from a high definition wide angle webcam. The image is resized to a 480x640 resolution to reduce the overall number of pixels that need to be processed each cycle while still retaining enough information for OHM to navigate. Once the image has been captured the image is sampled pseudo randomly to find the average RGB values of a small percentage of the pixels in the frame. Using this calculated mean the rest of the image is then processed by converting the image to a grayscale image based on the magnitude of the distance of each pixels color values are from the mean. By sampling for the distance from the mean OHM ignores what it sees most of which should be grass and extract the other colors and objects. Figure 4: Lane Extraction shows the raw input converted to a grayscale image.

Using only the method described above, decent lane and object extraction results are produced. However the algorithm also managed to pick up areas of dead grass since their color values are far enough away from the mean to be picked up in some instances. This was corrected by applying a Hough transform on the grayscale image. The Hough transform is a curve fitting algorithm that samples the image by seeing if points in the image lie on the same line as each other. This algorithm takes parameters that allow OHM to adjust the minimum and maximum length of a line as well as the angular resolution of lines. With some testing and tuning OHM has successfully found a middle ground between accepting long lines that are lanes while still allowing for true obstacles to appear as a series of lines in the image.

The final step to determining which path the vision would like to take is to overlay preset mask with various turning angles associated and see which one best fits the current frame. Determining which mask is the best fit is done by performing a dot product operation on the mask image which contains the path desired with the lane and object extraction image. The mask that returns the lowest dot product has the least collisions and is chosen to be the best path for that control cycle. For debugging purposes the image region of interest and the extracted obstacles are overlaid on the raw image to produce a composite image shown in Figure 5.

Figure 4: Lane Extraction

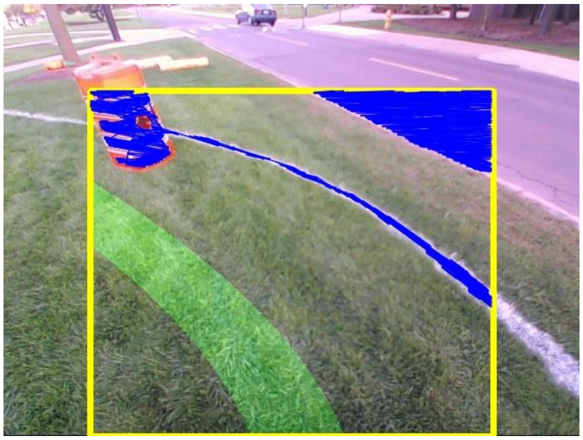

Figure 5: Composite Overlay

5.2 Camera Calibration & World Transformation

The above section on lane following provides insight as to how OHM is able to extract features from its surrounding. When planning to create a map of the environment that can be referenced later, an additional step in the process is required to transform the image pixels into x and y coordinates. This can be achieved by calibrating the camera based on its intrinsic parameters, which involve how the camera distorts the image, and its extrinsic parameters that involve the physical orientation of the camera relative to the robot and ground plane.

When calibrating a camera a 3x3 rotation matrix is used to convert pixels to x, y, z coordinates (OpenCV, 2010). In the case of the camera used on OHM, it is assumed that the camera is at a fixed height, facing directly forward. With these assumptions and some geometry, it can be shown that the rotation matrix for converting pixels to x, y coordinates comes down to five parameters described in Table 4. The resulting equations for extracting the relative x and y coordinates are defined in equations (3) and (4).

Table 4: Camera Calibration Parameters

Parameter	Effect
k_R	Prolong or shorten the world map vertically (y-axis)
k_F	Prolong or shorten height of objects in world map (z-axis)
k_C	Prolong or shorten the world map horizontally (x- axis)
C_l	Adjust the center line of the world map
R_{inf}	Represent the row where sky and ground meet in the camera's view. (horizon)

$$y = \frac{\frac{k_R}{(Row - R_{inf})} - 1}{k_F} \quad (3)$$

$$x = \frac{Col - C_l}{(Row - R_{inf}) * k_C} \quad (4)$$

The resulting effects of camera calibration allow OHM to create an image that more realistically represents distances. The raw calibration image and outcome image after calibration has been completed are shown in Figure 6.

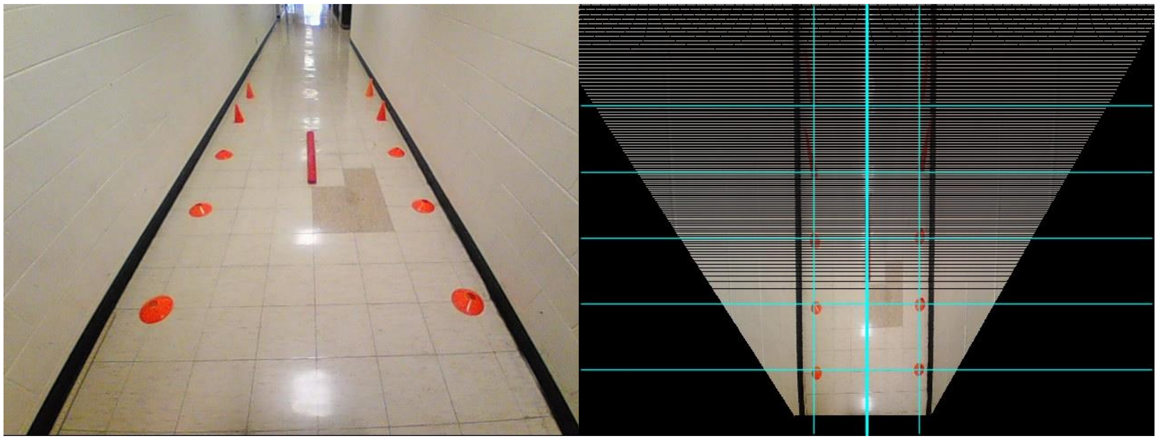

Figure 6: Camera Calibration

5.3 Waypoints Navigation

The first step for waypoint navigation is to obtain the latitude and longitude data from the GPS. The absolute location of the robot is obtained from the device in a standardized NMEA0183 string format. The NMEA0183 protocol provides numerous sentence structures containing various different information regarding receiver position, heading, speed, and number of satellites seen. For the IGVC the team has chosen to use the GPRMC format which contains the recommended minimum GPS data for transit. A typical received string would look like the following (Baddeley, 2001).

```
$GPRMC,220516,A,5133.82,N,00042.24,W,173.8,231.8,130694,004.2,W*70
         1    2    3    4    5     6    7    8      9     10  11 12


      1   220516     Time Stamp
      2   A          validity - A-ok, V-invalid
      3   5133.82    current Latitude
      4   N          North/South
      5   00042.24   current Longitude
      6   W          East/West
      7   173.8      Speed in knots
      8   231.8      True course
      9   130694     Date Stamp
      10  004.2      Variation
      11  W          East/West
      12  *70        checksum
```

This string is passed through a parser that will call a specified function when a complete sentence is received. This function updates the current location status of the robot. When using the GPRMC string it is also possible to get the current heading using only the GPS. To perform waypoint navigation the desired heading is calculated using trig functions based on the current position of the robot and the current target waypoint. The turn value desired is calculated based on the difference between the desired target heading value.

5.4 Obstacle Avoidance

Although the vision algorithm is capable of detecting a majority of obstacles it is possible that an obstacle may appear out of the view of the camera. This is where obstacle avoidance using the SICK LMS LiDAR is implemented. LiDAR is an electro-optical laser measurement system that emits infrared beams and measures the angle of incident and distance of objects that are struck by the beams. Using the measured angles and distances OHM can scan points that are within a buffer zone around itself. This is considered to be the halo safe zone in the control scheme. If any objects are detected in this halo the angles at which the left and right wheel must travel to be clear of the object are calculated (Figure 7).

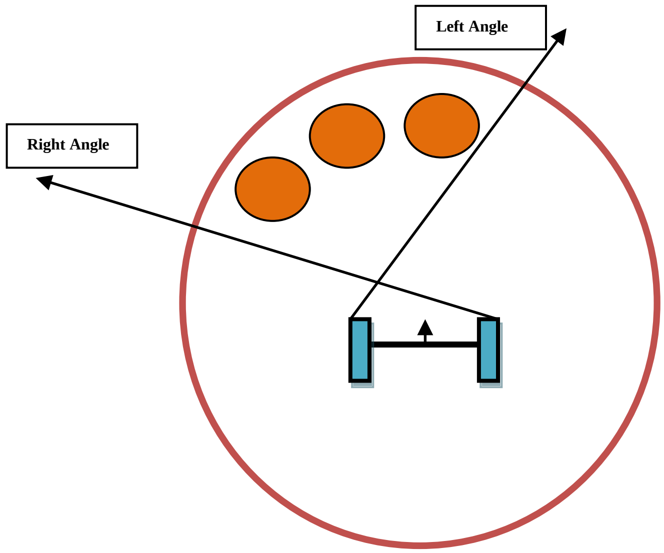

Figure 7: Obstacle Avoidance

Testing for this module involved setting the desired target point to always be directly in front of the robot while travelling down a hallway. Obstacles were then placed along the edges and the goal was to get to the end of the hall while avoiding the obstacles.

Once the hallway test was accomplished the obstacle avoidance with LiDAR was then integrated into the lane following section. In order to ensure that the obstacle avoidance doesn't override the lane following the vision algorithm is first checked and the best mask is determined. Next the obstacle detection process is run and if there is an obstacle the angle that gives the least variation from the vision turn angle is then used to guide the robot during the current control cycle. This decision making process is outlined in Figure 8.

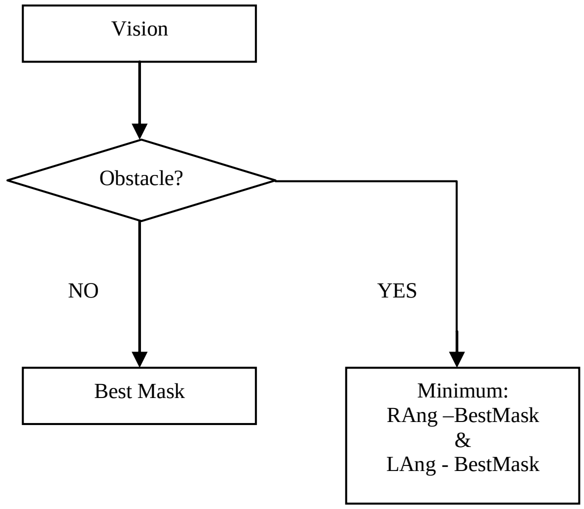

Figure 8: Turn Selection

In the open field a similar strategy is employed, however an additional layer is added to the top of the chart where the desired turn to the GPS point is calculated before the vision and obstacle avoidance processes are run.

5.5 Mapping and Path Planning

OHM creates a map as it encounters obstacles and lanes along the course that it can use as reference points. This map will be built by logging objects and their Cartesian coordinates relative to the starting location of the run through a combination of GPS location, LiDAR object detection, and Vision feature extraction. After having built a database of static objects provided by integrated system the objects can now be considered landmarks. These landmarks are then used with a Simultaneous Location And Mapping (SLAM) algorithm that is run to guide the OHM through the course. OHM makes use of the following when attempting to locate itself in the course:

1. Known landmarks X coordinate
2. Known landmarks Y coordinate
3. Measured distance to landmark
4. Measured angle the landmark was detected
5. Initial guess of Robot X, Y, and θ must be close to actual location in order to converge on a solution.

Through a combination of the Least Squares method and the use of the Marquardt –Levenberg algorithm which attempts to converge on a solution by adjusting the Robots X, Y, and θ until the change in the

Total Error is reduced to a minimum. The Marquardt –Levenberg is an expansion on the Gauss–Newton algorithm of gradient descent. Both methods require the initial inputs to the system to be relatively close to the solution. However, Marquardt –Levenberg is more robust than the Gauss–Newton, which means that in many cases it finds a solution even if it starts very far off the final minimum.

For most systems the Marquardt–Levenberg will converge slower than the Gauss-Newton method but with the added incentive that steps taken in a direction away from the solution can be corrected by increasing the diagonal multiplier λ of the Hessian. As λ increases steps are smaller but are more likely to follow the gradient towards the solution. The Marquardt –Levenberg method can also be viewed as Gauss–Newton with a trust region (Levenberg–Marquardt). Below is the definition of symbols used for the Marquardt –Levenberg method:

Definition of Symbols

X_R	X coordinate of Robot
Y_R	Y coordinate of Robot
θ_R	Heading of Robot
E	Sum of Error Squared
∇E	Gradient of E
N	Number of Landmarks
Δ_{i_x}	Error in X for Landmark i
Δ_{i_y}	Error in Y for Landmark i
f_i	1^{st} Derivative of E with respect to X_R
g_i	1^{st} Derivative of E with respect to X_R
h_i	1^{st} Derivative of E with respect to X_R
H	Hessian Matrix of 2^{nd} Derivatives

$$E = \frac{1}{2}\sum_{i=1}^{N}({\Delta_{i_x}}^2 + {\Delta_{i_y}}^2) \quad (5)$$

$$\Delta_{i_x} = \left[X_i - X_R - d_i * \sin(\theta_R + \phi_i)\right] \quad (6)$$

$$\Delta_{i_y} = \left[Y_i - Y_R - d_i * \cos(\theta_R + \phi_i)\right] \quad (7)$$

$$f_i = \frac{1}{2}\frac{\partial}{\partial X_R}E = -\Delta_{i_x} \quad (8)$$

$$g_i = \frac{1}{2}\frac{\partial}{\partial Y_R}E = -\Delta_{i_y} \quad (9)$$

$$h_i = \frac{1}{2}\frac{\partial}{\partial \theta_R}E = \ -d_i \cos(\theta_R + \phi_i)\Delta_{i_x} + d_i \sin(\theta_R + \phi_i)\Delta_{i_y} \tag{10}$$

$$\nabla E = \begin{bmatrix} \frac{1}{2}\frac{\partial}{\partial X_R}E \\ \frac{1}{2}\frac{\partial}{\partial Y_R}E \\ \frac{1}{2}\frac{\partial}{\partial \theta_R}E \end{bmatrix} = \begin{bmatrix} f_i \\ g_i \\ h_i \end{bmatrix} \tag{11}$$

$$\mathrm{H} = \begin{bmatrix} \frac{\partial}{\partial X_R}\frac{\partial}{\partial X_R}E & \frac{\partial}{\partial Y_R}\frac{\partial}{\partial X_R}E & \frac{\partial}{\partial \theta_R}\frac{\partial}{\partial X_R}E \\ \frac{\partial}{\partial X_R}\frac{\partial}{\partial Y_R}E & \frac{\partial}{\partial Y_R}\frac{\partial}{\partial Y_R}E & \frac{\partial}{\partial \theta_R}\frac{\partial}{\partial Y_R}E \\ \frac{\partial}{\partial X_R}\frac{\partial}{\partial \theta_R}E & \frac{\partial}{\partial Y_R}\frac{\partial}{\partial \theta_R}E & \frac{\partial}{\partial \theta_R}\frac{\partial}{\partial \theta_R}E \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathrm{f_i}}{\partial \mathrm{X_R}} & \frac{\partial \mathrm{f_i}}{\partial \mathrm{Y_R}} & \frac{\partial \mathrm{f_i}}{\partial \mathrm{\theta_R}} \\ \frac{\partial \mathrm{g_i}}{\partial \mathrm{X_R}} & \frac{\partial \mathrm{g_i}}{\partial \mathrm{Y_R}} & \frac{\partial \mathrm{g_i}}{\partial \mathrm{\theta_R}} \\ \frac{\partial h_\mathrm{i}}{\partial \mathrm{X_R}} & \frac{\partial h_\mathrm{i}}{\partial \mathrm{Y_R}} & \frac{\partial h_\mathrm{i}}{\partial \mathrm{\theta_R}} \end{bmatrix} \tag{12}$$

$$\mathrm{H}_{11} = \frac{\partial}{\partial X_R}\frac{\partial}{\partial X_R}E = \frac{\partial \mathrm{f_i}}{\partial \mathrm{X_R}} = \sum_{i=1}^{N} 1 \tag{13}$$

$$\mathrm{H}_{12} = \frac{\partial}{\partial Y_R}\frac{\partial}{\partial X_R}E = \frac{\partial \mathrm{f_i}}{\partial \mathrm{Y_R}} = \sum_{i=1}^{N} 0 \tag{14}$$

$$\mathrm{H}_{13} = \frac{\partial}{\partial \theta_R}\frac{\partial}{\partial X_R}E = \frac{\partial \mathrm{f_i}}{\partial \mathrm{\theta_R}} = \sum_{i=1}^{N} d_i \cos(\theta_R + \phi_i) \tag{15}$$

$$\mathrm{H}_{21} = \frac{\partial}{\partial X_R}\frac{\partial}{\partial Y_R}E = \frac{\partial \mathrm{g_i}}{\partial \mathrm{X_R}} = \sum_{i=1}^{N} 0 \tag{16}$$

$$\mathrm{H}_{22} = \frac{\partial}{\partial Y_R}\frac{\partial}{\partial Y_R}E = \frac{\partial \mathrm{g_i}}{\partial \mathrm{Y_R}} = \sum_{i=1}^{N} 1 \tag{17}$$

$$\mathrm{H}_{23} = \frac{\partial}{\partial \theta_R}\frac{\partial}{\partial Y_R}E = \frac{\partial \mathrm{g_i}}{\partial \mathrm{\theta_R}} = -\sum_{i=1}^{N} d_i \sin(\theta_R + \phi_i) \tag{18}$$

$$\mathrm{H}_{31} = \frac{\partial}{\partial X_R}\frac{\partial}{\partial \theta_R}E = \frac{\partial \mathrm{h_i}}{\partial \mathrm{X_R}} = \sum_{i=1}^{N} d_i \cos(\theta_R + \phi_i) \tag{19}$$

$$\mathrm{H}_{32} = \frac{\partial}{\partial Y_R}\frac{\partial}{\partial \theta_R}E = \frac{\partial \mathrm{h_i}}{\partial \mathrm{Y_R}} = -\sum_{i=1}^{N} d_i \sin(\theta_R + \phi_i) \tag{20}$$

$$\mathrm{H}_{33} = \frac{\partial}{\partial \theta_R}\frac{\partial}{\partial \theta_R}E = \frac{\partial \mathrm{h_i}}{\partial \mathrm{\theta_R}} = \sum_{i=1}^{N} {d_i}^2[\cos^2(\theta_R + \phi_i) + \sin^2(\theta_R + \phi_i)] = \sum_{i=1}^{N} {d_i}^2 \tag{21}$$

$$\mathrm{H} = \begin{bmatrix} N & 0 & \sum_{i=1}^{N} d_i \cos(\theta_R + \phi_i) \\ 0 & N & -\sum_{i=1}^{N} d_i \sin(\theta_R + \phi_i) \\ \sum_{i=1}^{N} d_i \cos(\theta_R + \phi_i) & -\sum_{i=1}^{N} d_i \sin(\theta_R + \phi_i) & \sum_{i=1}^{N} {d_i}^2 \end{bmatrix} \tag{22}$$

$$\begin{bmatrix} X_{R+1} \\ Y_{R+1} \\ \theta_{R+1} \end{bmatrix} = \begin{bmatrix} X_R \\ Y_R \\ \theta_R \end{bmatrix} + \mathrm{H}^{-1}\nabla E \tag{23}$$

$$\begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta \theta \end{bmatrix} = -H^{-1} \nabla E \tag{24}$$

To minimize the errors, one requires 1st and 2nd partial derivatives of E with respect to X_R, Y_R, θ_R. It is convenient to first find the derivatives for each landmark and then adding them. The Levenberg-Marquardt method approximates the second derivatives that involve Δ as show in Equations (12)-(22). The Levenberg-Marquardt algorithm then attempts to find a local maxima or minima, because of this a relatively close initial guess of the current X_R, Y_R, and θ_R is needed in order for the algorithm to converge. The algorithm is an iterative process that should converge within one or two iterations but because processing power to run more iterations is available the algorithm is run up to 100 times to find X_{R+1}, Y_{R+1}, and θ_{R+1} shown in Equation (23).

6. Cost Analysis

#	Description	Actual Cost
Chassis Construction/Mechanical Design		
3	Wood Panels	$100.00
3	Electric Wheelchair Wheels 12.5 Wheel	$300.00
2	Wheelchair Electric Motors	$928.00
Motor Control Circuitry		
1	Roboteq Motor Controller Ax2850	$620.00
Robot Computer		
1	MacBook Pro	$1999.00
Sensors		
1	Sick Laser Sensor	$5273.00
1	Ocean Server OS5000 Compass	$500.00
Camera		
1	Microsoft Lifecam HD	$50.00
1	Camera Enclosure	$4.95
Controller Transmitter and Receiver		

1	Bulldog Wireless Car Starter	$60.00
GPS		
1	Garmin GPS 19x HVS	$195.00
Accessories		
2	2 Optima Yellowtop Batteries	$450.00
2	Multi-Bank Battery Charger: 5/5 amp 2 Bank Charger	$189.00
1	Switches, Wires, Connectors, Electronic Parts, Enclosure, Etc	$200.00
Tax(6%)		$652.14
Total		$11,521.09

7. Design Innovations

The OHM platform strives to constantly improve the field of mobile robotics. The fact that the OHM platform is a multipurpose vehicle capable of carrying a payload (i.e. mail carrier) and easily converting into a snowplow makes OHM a diverse and innovative machine. Future teams can easily use the mechanical platform and repurpose it for a new task. One such possibility that has been briefly explored is the attachment of a lawnmower to make a more practical and consumer market oriented product.

8. Conclusion

The OHM team is returning to the IGVC for the second year in a row and is excited for the opportunity to use what they have learned from last years' experience to improve their performance. The IGVC provides a very challenging task that forces teams to use and integrate a variety of different sensors. OHM will be using 3 main sensors in order to complete the IGVC course: vision camera, GPS, and a LiDAR. Custom C software allows these 3 sensors to work together in order to complete the challenges identified in the Design Overview section which include, lane following, obstacle avoidance, and GPS waypoint navigation. When all these sensors are combined together an intelligent autonomously navigating vehicle is produced. Thank you for this opportunity.

9. Acknowledgments

All of the team members from OHM would like to thank our advisor, Dr. Nattu Natarajan, for all of the time and commitment that he has shown to club, and to the principle of teaching and sharing knowledge. Without his commitment, enthusiasm, knowledge, and experience the Intelligent Systems Club at the University of Michigan – Dearborn would not be able to participate in advanced competitions, such as the IGVC.

10. Bibliography

Baddeley, G. (.-N.-N. (2001, June 20). *GPS - NMEA sentence information*. Retrieved May 15, 2014, from http://aprs.gids.nl/nmea/

Levenberg–Marquardt. (n.d.). Retrieved January 20, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Levenberg%E2%80%93Marquardt_algorithm

OpenCV. (2010). *OpenCV 2.1 Reference*. Retrieved March 20, 2014, from http://opencv.jp/opencv-2.1_org/c/