OHM MK-III Design Report

2015 Intelligent Ground Vehicle Competition

University of Michigan – Dearborn

Dearborn, Michigan 48128, USA

May 15, 2015

Team: Angelo Bertani, Jason Spurlock, Hwarang Kim, Eric Aitken, Michael Bowyer, Jean Paul Denoyer, Ryan Sheen, Aaron Cofield

Advisor: Nattu Natarajan

I hereby certify that the design and engineering of OHM MK-III for the 2015 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.

N. Natarajan
____

Dr. Nattu Natarajan

1. Introduction

The team from the Intelligent Systems Club (ISC) at the University of Michigan – Dearborn is excited to present their latest creation for use in 2015 Intelligent Ground Vehicle Competition (IGVC). The team has made use of experience gained in previous IGV Competitions to produce OHM MK-III. Design and construction of OHM MK-III began in early February after the team returned from competing in the 5th Annual Institute of Navigation Autonomous Snowplow Competition.

1.1 Team Composition

The OHM team consists of students, with backgrounds in Computer, Electrical, Software, and Mechanical Engineering. The overall team was broken down into smaller sub teams to handle individual tasks required to have a successful outcome. Table 1 shows each team members area of study.

Table 1: Team Member List

Name	Major	Graduation
Angelo Bertani	Masters Electrical Engineering	April 2017
Jean Paul Denoyer	CIS	April 2015
Hwarang Kim	Electrical & Computer Engineering	April 2015
Erik Aitken	Software Engineering	December 2015
Michael Bowyer	Electrical & Computer Engineering	April 2017
Ryan Sheen	Mechanical Engineering	December 2017
Aaron, Cofield	Computer Engineering	August 2018
Jason Spurlock	Robotics Engineering	April 2018

2. Design Innovations

The OHM platform is in its third generation and each successive generation attempts to improve upon the last. OHM MK-III features an entirely new chassis using the successful portions of previous designs as a basis while improving on the shortcomings. Key innovations are:

1. Spring and damper suspension system.
2. New wrong way detection algorithm.
3. Flag detection algorithm.
4. Dead Reckoning.
5. Improvements to existing modules.
6. Code ported from C to C++ for an object oriented approach.
7. Addition of controllable safety light.

3. Mechanical Design

3.1 Overview

Several considerations influenced the design of the OHM MK-III entry in the 2015 IGVC. A cap was not placed on design cost, but an effort to find the least expensive option still suitable for use was made. First the design was to be reconfigurable to allow future expatiation and experimentation with various sensor packages. Second the vehicle should be highly maneuverable to navigate the narrow passages posed by certain obstacle arrangements. The third concern was keeping weight down to reduce power requirements when not carrying a payload. Finally the most important consideration was to design a platform that would be easy to manufacture. Ease of manufacturing was critical due to the limited tool set available in the lab and at the competition site. Recent platforms built by the ISC for plowing snow required use of specialized equipment to complete construction and were difficult to maintain, repair, or modify. Many of the concepts for the OHM MK-III platform were borrowed from the success of the OHM 2.0 platform that placed 2nd in the 2014 IGVC AutoNav Challenge.

3.2 Frame

The design conceived consists of aluminum rails bolted together to form a four wheel robot. Two front mounted drive wheels and two rear mounted castors to provide maximum maneuverability while optimizing useable frame space. A camera bar extends upwards on the front of the vehicle to a total height of 5ft. The LiDAR and GPS units are mounted in front center on the deck of the vehicle. The motors are mounted to swing arms which are mounted to the frame with 1200 N rated bike springs. Finally a sheltered area on the rear of the deck provides space for environmentally-sensitive components.

The base of the vehicle is an aluminum frame of bolted aluminum construction. Extruded 6061 aluminum in square tubular 1.25x1.25x0.125 inch and angle1x1x0.125 inch forms are combined with bent 5052 c-channel aluminum to form a ridged structure. Components are fastened together with ¼-20 steel bolts and nuts, or by use of 3/8in. or 1/8in. aluminum rivets. The use of these materials allows for rapid assembly; all pieces can be rapidly cut from the raw stock with a hacksaw. Holes for the mechanical fasteners can be rapidly drilled in the material with conventional power drills. The aluminum frame is also much lighter when compared with a similar frame of steel.

The structure is based around two parallel rails each composed of one piece of square tube sandwiched between two pieces of c-channel. Perpendicular pieces of c-channel span the rails creating a very rigid rectangular structure. A superstructure was then built off this base, providing containment for batteries and electronics, as well as mounting points for wheels and motors. The majority of the structure is beneath the main rectangular base, when the vehicle is in proper orientation, leaving a flat top surface upon which payloads may be carried. A camera mast and sheltered payload area protrude above the top of the base. The overall dimensions of the vehicle are roughly 42in. by 32in by 60in (36in without the camera mast, which can be readily removed from the frame by unfastening two bolts). This size allows the vehicle to be easily transported via van and fit through ADA approved doorways.

3.3 Suspension

The front of the frame features experimental sprung swing arms. Motors are attached to the underside of these swing arms, which pivot about the center of each parallel rail. Mountain bike springs are used for the damping elements. The purpose of this suspension is to mitigate up and down oscillations experienced by the camera. Testing will be conducted to see if this setup provides any improvement versus a rigid chassis.

Figure 1: Rear Right View (top), Front Right View (bottom)

3.4 Powertrain

The vehicle is propelled by twin 24 volt NPC DC motors with integrated 24:1 gearboxes, providing a maximum of .81 horsepower each. The motors are attached to the frame via the sprung swing arms at the front of the chassis. The mounting arrangement creates a setup where the vehicle will use differential steering to turn. Twin castors are mounted in the rear for the vehicle for increased stability and useable chassis space as opposed to a three wheel design.

Two 13in diameter foam filled rubber tires are used as the front drive wheels. The tread pattern of these wheels provides suitable grip on grass and turf surfaces. In the back two 8.5in pneumatic are attached to castor assemblies allowing free rotation. It was decided that 4 wheels would be used (two drive wheels and two free rotating castors) over three (two drive wheels and one free spinning castor) for a couple reasons. First using 4 wheels frees up space on the chassis for components such as batteries (having only three wheels would require a wheel to be mounted centrally on the width of the frame, splitting usable space in half). Second the four wheeled design allows avoidance of potholes or other small size obstacles by straddling the obstacle beneath the center of the vehicle.

4. Electrical Design

The electrical system on the OHM platform was designed to provide the required power to all sensors and motors in a manner that is safe and efficient. The target battery life of the system is 4-6 hours under moderate stress.

4.1 Power

The mechanical design of the battery compartment allows for a maximum of three Optima Yellowtop batteries, two 55Ah and one 75Ah, and four smaller 7Ah Werker batteries. All batteries are 12V batteries. The 55Ah batteries are connected in series to provide the 24V supply to both the motors and LiDAR. The 75Ah battery is intended for use with a DC/AC Inverter to charge the laptop. The 7Ah batteries provide power to all other 12V sensors. 5V sensors such as a compass are powered from a USB hub. The hub is provided external power through a 5V, 3A battery pack typically used for charging cell phones.

For the competition two 18Ah batteries have been substituted in place of the 75Ah battery. The decision to keep the four 7Ah batteries out of the battery compartment ensures that in an emergency case charged batteries are available to continue testing or competing.

4.2 Safety System

Safety when designing any autonomous process is a primary concern. The safety circuit on OHM has been designed with both a physical and a remote estop circuit. Both circuits perform the same function of shutting down the power to the motors on the vehicle. An additional safety feature using a self-latching relay is implemented in the remote shutdown circuit. The self-latching setup allows a team member time to reach the robot and debug code and sensors that led to the need for remote shutdown while ensuring that motion cannot be restored without first shutting off the master switch. The schematic for the power and emergency stop circuits is shown in Figure 2.

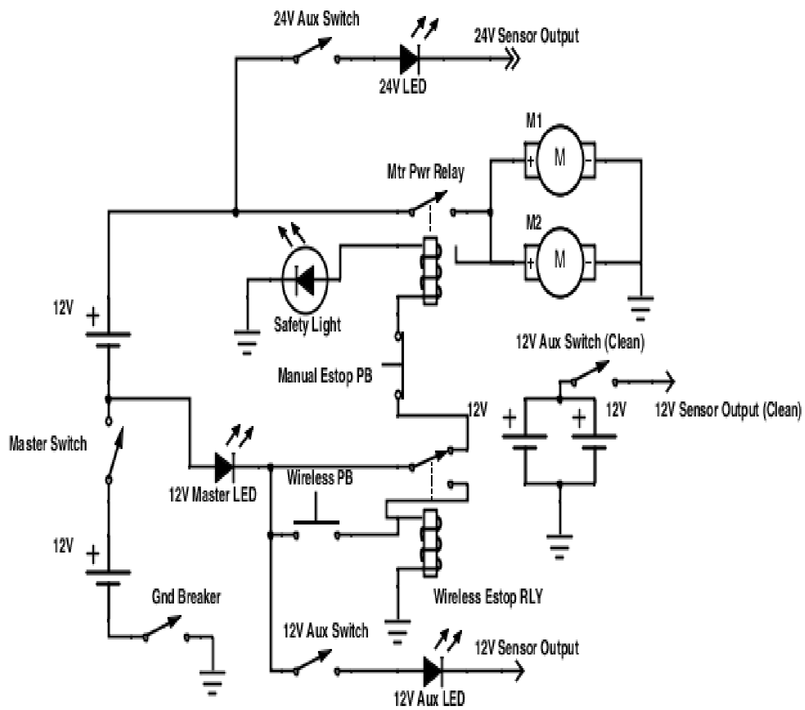

Figure 2: Power & Safety Circuit

In addition to the safety circuits a strobe light is fixed to the vehicle that is powered anytime there is power to the motors. The light used was chosen because it has the capability of being toggled into a solid mode. The toggling ability allows the team to comply with the rule requiring a safety light go from solid to flashing when going from manual to autonomous modes. The control function is performed using an Arduino and a relay. The relay connection is in a normally open setup that forces the light to always strobe in the event the Arduino is malfunctioning, ensuring that bystanders are warned that there is a potential for motion.

4.3 Sensors and Communication

OHM utilizes multiple sensors in its design, each having their own purpose and communication specifications. Below in Table 2 is a list of all the sensors, their purpose, and the communication protocol that they require.

Table 2: Sensor List

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

4.4 Computer

Processing of sensor data and output command formation is done through a laptop. This year the primary computing machine is a HP ProBook with an Intel i7 processor at 2.90 GHz and 8GB of ram. The operating system in use is Windows 8.1. The use of a laptop in place of an on board computer has proven useful to the team in the past. This approach allows team members to develop code and test on the machine without the need to transfer files to an onboard computer. In an extreme case an onboard Raspberry Pi Model 2 is onboard and capable of performing basic manual control. Autonomous navigation with the Pi is still under development.

5. Intelligent Systems

5.1 Overview

An Intelligent System is a machine that has the capacity to gather, interpret, and analyze data and use this data to communicate with other systems in an effective and productive manner. To compete at a high level in the IGVC software must be designed to incorporate the data from onboard sensors, analyze it, and produce a command to guide the robot to the finish line.

The software development team used the same principles of modularity and ease of use as the mechanical team. The process of developing software started by first identifying the necessary modules required to complete each portion of the IGVC. The team identified three primary modules and the sensor used to complete the task in Table 3.

Table 3: Module List

Requirement	Component Used:
GPS Waypoint Navigation	Garmin: **GPS 19x HVS**
Lane Following	Logitech: **C525 HD Webcam**
Obstacle Avoidance	SICK: **LMS 111 270° Scan**
Motor Control	Roboteq: XDC2230

Figure 3 shows a flow chart of how an individual module is developed. Each module goes through the same process. A design is considered complete when the module in question can successfully perform its purpose with no outside interference. For example the GPS Navigation module is considered complete when it can guide the robot from waypoint to waypoint regardless of lanes or obstacles. Once each subsystem can perform on a standalone basis the next step is to integrate them into a cognitive application that will plan a path through the course.

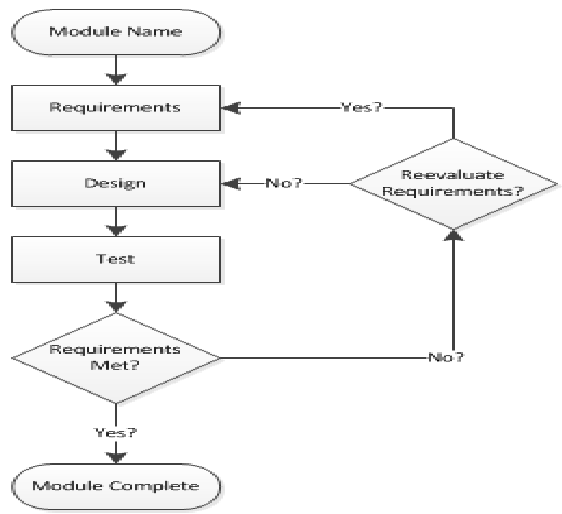

Figure 3: Module Design Flow Chart

5.2 Motor Control

The robot interfaces with a dual channel Roboteq motor controller to send speed commands to the motors through a serial connection. One channel on the Roboteq is used per motor. 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. The formulas are derived from the method used to control the robot with a single 2-axis joystick where the X value of the joystick is turn and the Y value is speed.

$$\text{Left Speed} = (\text{Speed} + \text{Turn}) * \text{Desired Speed} \qquad (1)$$

$$\text{Right Speed} = (\text{Speed} - \text{Turn}) * \text{Desired Speed} \qquad (2)$$

5.3 Vision

5.3.1 Overview

The IGVC requires that an autonomous vehicle be able to successfully track a lane. During the first portion of a heat the lane is defined by two lines and guides the vehicle into the open field or “No Man’s Land.” Once in the open field the perimeter of the competition field is again outlined by a single white line. At the end of the course the vehicle must go through a set of red and blue flags keeping red flags on the right and blue on the left. The overwhelming use of vision to complete the course makes the vision module extremely critical, failure of the vision module will likely result in a failed run.

OHM MK-III performs the task of lane tracking by using a webcam and the open source computer vision library OpenCV, which provides access to many image processing algorithms and features. All captured images are resized to 120x160. The reduced image size cuts down on overall processing time by reducing the number of pixels that need to be sampled. OHM MK-III operates in the RGB color space, testing with HSV color space is still under development.

5.3.2 Template Matching

The vision system is divided into a set of filters that produce a greyscale image. The greyscale image is then convolved with a set of predefined templates that correspond to various turn angles. The best path is one that returns a minimum value, or has the least overlap with the filtered image. Each template is assigned an importance value that will break ties between templates of equal value. Figure 4 shows an example of the template matching process.

Figure 4: Lane Extraction

5.3.3 Lane Tracking

In order to track a lane OHM must first determine which pixels in the image belong to the lane and which do not. During this process a filter is applied to the image that looks for pixels that are close to white. A Hough transform is then performed to find lines within the image. The Hough transform can be parameterized to prefer long lines over short lines. With proper parameterization this process effectively removes small patches of white that are not part of a lane creating an output better suited for template matching than the raw white filter.

5.3.4 Flag Detection

For the colored flag detection portion of the course two separate filters for red and blue are used. The same process used for lane extraction is also used for flag detection. An advanced strategy for performing the flag task uses the algorithm discussed in the Landmark Based Navigation section. The flags captured in the image are used as the landmarks and help guide the vehicle through the flags.

5.3.5 Camera Calibration & World Transformation

Previous sections of the vision system are run using a system that does not transform pixels in the frame to useable world coordinates. To get useable X, Y coordinates from extracted features the camera on OHM must be calibrated. The calibration process makes a few assumptions to simplify the rotation matrix. First it is assumed that the camera is at a fixed height, and front facing. With these assumptions the rotation matrix for converting pixels to x, y coordinates is reduced to five parameters described in Table 4. Equations for x and y coordinates are defined in equations (3) and (4). The transformed pixels can then be used in other systems.

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)

$$y = \frac{\frac{k_R}{(Row-R_{inf})}-1}{k_F} \quad (3) \qquad x = \frac{Col-C_l}{(Row-R_{inf})*k_C} \quad (4)$$

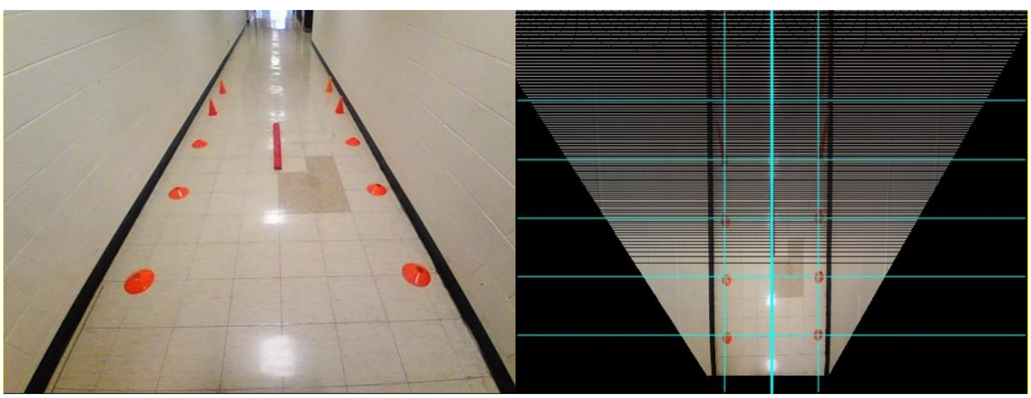

Figure 5: Camera Calibration

5.4 Waypoints Navigation

OHM MK-III uses a PD control algorithm for waypoint navigation. This algorithm adjusts wheel speeds based on present and past errors. Error is defined as θ_d the difference between θ_r and θ_w where θ_r is the current robot heading and θ_w is the calculated heading to the waypoint. Equation (5) shows the calculation of θ_d. The error is then passed to the motor control function as a turn value.

$$\theta_d = \tan^{-1}\frac{X_r - X_d}{Y_r - Y_d} \qquad (5)$$

5.5 Obstacle Avoidance

5.5.1 Vision

Although the primary task of the vision system is to track lanes and detect flags it is also capable of detecting and avoiding obstacles. Once the image has been captured the image is randomly sampled to find the average RGB values and creating a covariance matrix from a small percentage of the pixels in the frame. Using the calculated mean and covariance the full image is then converted to greyscale based on the magnitude of the distance from the average. This process assumes that the majority of the image will be grass. The resulting output typically ignores grass and brings out all objects, including lanes. Data from the output image is transformed into world coordinates using the equations from section 5.3.5. The transformed coordinates can then be used in the avoidance algorithm.

5.5.2 LiDAR

Using only vision OHM MK-III is capable of avoid the majority of obstacles. Obstacle avoidance can be improved using a scanning laser range finder or LiDAR. The device chosen is the SICK LMS 111 LiDAR with a 20m range, 270 degree scan field, and a 0.25 degree step resulting in a total of 1081 data points per measurement. Using the distance and angle measurements from the range from the LiDAR physical objects can be detected more accurately than with just vision. Because of the range capability of the LiDAR data must first be filtered down into a reasonable zone before the avoidance algorithm is run.

5.5.3 Avoidance System

Once data is filtered the avoidance system calculates the minimum angle the robot must travel at to avoid an obstacle. The algorithm starts by taking the desired angle to travel which may be given by the waypoint navigation or lane tracking depending on the mode. The given is angle is the first angle tested, if this angle contains an obstacle successive angles are tried branching out from the start angle to cover the entire view of the sensors. The system returns two angles, one for avoiding to the right, and another for avoiding to the left. Figure 6 shows a simple example of the avoidance algorithm.

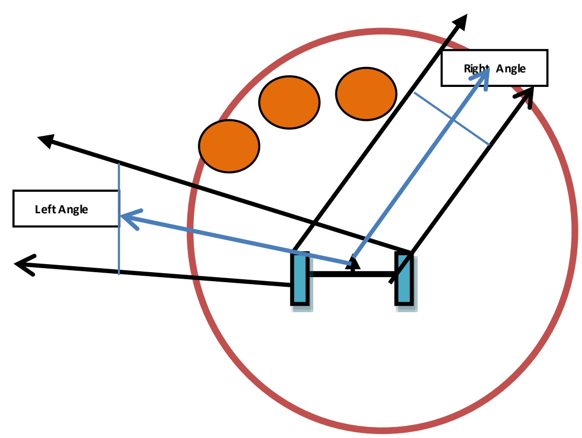

Figure 6: Obstacle Avoidance

5.6 Landmark Based Navigation

OHM is capable of another form of navigation using landmarks. This system builds a database of known objects that are used to perform Simultaneous Location And Mapping (SLAM). The SLAM system does not attempt to create a global map. Instead it creates local maps which can be used to guide the robot to temporary waypoints over short distances. This system could be used for global navigation in the event of a GPS failure. The requirements for successful SLAM are listed in Table 5. Equations showing how the SLAM algorithm is implemented can be found in the Appendix.

Table 5: SLAM Requirements

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

5.7 Wheel Encoders

5.7.1 Overview

Wheel encoders provide OHM the capability of performing two major functions. The first being speed control and the second Dead Reckoning. In order for wheel encoders to provide useful information they must first be calibrated to determine the number of pulses per meter.

5.7.2 Speed Control

Speed control is performed using a PI controller. Outputs in the motor control equations are adjusted based on the difference between the current speed and the desired speed. If the current speed is under the desired speed more power can be provided to the wheels to bring the speed up and power can be reduced if the speed is greater than desired. Power reduction is useful in the event of wheel slippage and power increase when travelling up an incline.

5.7.3 Dead Reckoning

Use of wheel encoders allows OHM to perform read reckoning. Dead reckoning could perform global waypoint navigation but because errors when performing dead reckoning accumulate over time it is not an ideal solution. Dead reckoning is only used to perform sharp turns where the GPS heading may be inaccurate or a GPS signal is lost. Dead reckoning can be combined with the GPS unit in a Kalman filter to greatly improve location accuracy; however the team has not yet implemented this feature.

5.8 Wrong Way Correction

This year the team has developed a module that will detect when the vehicle is traveling the wrong way through a lane. Last year multiple teams experienced this problem and OHM attempts to solve it by logging past GPS coordinates going through the lane and ensuring that the distance to the first waypoint is being reduced. If it is detected that the distance to the first way point is increasing and the distance to a set of previous lane coordinates is reducing the robot will enter the wrong way correction routine. In this routine the vehicle will use dead reckoning and obstacle avoidance to turn and face the first way point before proceeding.

6. Cost Analysis

Table 6: Bill of Materials

Item	Quantity	Cost	Total
OPTIMA® Batteries 8012-021 YELLOWTOP	2	210.00	420.00
OPTIMA® Batteries 8051-160 YELLOWTOP	1	280.00	280.00
Socket bolts, Stainless steel 18-8, 1/4"-20 x 1" (Box-50)	2	13.00	26.00
Socket bolts, Stainless steel 18-8, 1/4"-20 x 1-1/2" (Box-25)	4	6.00	24.00
Socket bolts, Stainless steel 18-8, 1/4"-20 x 2" (Box-25)	4	10.00	40.00
Socket bolts, Stainless steel 18-8, 1/4"-20 x 2-1/2" (Box-25)	4	20.00	80.00
Hex Nuts, Grade 8 carbon steel 18-8, 1/4"-20 (Box-100)	5	8.50	42.50
Metal	1	501.05	501.05
Driv e Wheels	2	90.00	180.00
Camera - Logitech C525	1	59.99	59.99
Estop Pushbutton	1	56.25	56.25
Wireless Estop - Bulldog RA-10	1	87.20	87.20
Safety Light - Blue/Yellow	1	51.00	51.00
Wiznet Ethernet To Serial	2	34.04	68.08
Joystick - Logitech F710	1	39.99	39.99
Router (Correct)	1	69.99	69.99
Solenoid - White-Rogers 586-902	1	62.99	62.99
Power Pole Kit	1	49.99	49.99
6x6x4 Junction Box	1	19.99	19.99
Superbight Blue LED light strip	1	13.00	13.00
Superbight Yellow LED light strip	1	12.00	12.00
Main Breaker (150 Amp resettable breaker)	2	29.00	58.00
Bussmann HHX Maxi In-Line Fuse Holder	2	10.50	21.00
60-Amp Bussmann Fuse	6	5.20	31.20
40-Amp ATC Fuses (50 pack)	1	10.20	10.20
50-Amp ATC Fuses (2 per pack)	4	7.00	28.00
Yellow Ring Terminals (Bag of 100)	1	11.00	11.00
4" Zipties (Bag of 50)	2	4.15	8.30
Bike Shocks	2	10.95	21.90
Polycarb/ Plastic Panels	1	214.15	214.15
Wheel Hubs	4	45.43	181.72
-----Reused Parts-----			

Laptop	1	600.00	600.00
LiDAR - SICK LMS 111	1	6,999.99	6,999.99
GPS Unit - Garmin HVS 19x	1	199.99	199.99
Motors	2	450.99	901.98
Casters	2	100.00	200.00
Motor Controller - Roboteq XDC2230	1	425.00	425.00
Parts Needed Total			2,769.49
Total			12,096.45

7. Conclusion

The OHM team is in its 3rd year of competition and is excited to compete in the 2015 IGVC. The team wants to improve upon last year's 2nd place finish that was 2 feet short of crossing the finish line on the advanced course. The bar is set high but the team is confident that past experiences have adequately prepared them for the challenges the competition poses.

8. Acknowledgments

The team members from OHM would like to thank our advisor, Dr. Nattu Natarajan, for his unwavering commitment to the education of ISC members. His knowledge, experience, and pure enthusiasm are a large part of why the Intelligent Systems Club at the University of Michigan – Dearborn is able to participate in advanced competitions, such as the IGVC. The team would also like to thank all the staff and volunteers that make the IGVC possible each year. Thank you for the opportunity.

9. Appendix

Table 7: SLAM Symbol Definitions

Symbol	***Definition***
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
$\mathbf{N}$	Number of Landmarks
Δ_{i_x}	Error in X for Landmark i
Δ_{i_y}	Error in Y for Landmark i
f_i	1st Derivative of E with

	respect to X_R
$\boldsymbol{g_i}$	1st Derivative of E with respect to X_R
$\boldsymbol{h_i}$	1st Derivative of E with respect to X_R
$\boldsymbol{H}$	Hessian Matrix of 2nd Derivatives

Below are equations that define the process for performing Least Squares error reduction using the Marquardt –Levenberg method in landmark based SLAM. The algorithm attempts to converge on a solution by adjusting the unknown parameters (Robots X, Y, and θ) until the change in the Total Error (E) is reduced to a minimum. The Marquardt –Levenberg is an expansion on the Gauss–Newton gradient descent algorithm. Both methods require the initial inputs to the system to be relatively close to the solution. For most systems the Marquardt–Levenberg will converge slower than the Gauss-Newton method but with the added incentive that steps taken that increase Total Error 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).

$$E = \frac{1}{2}\sum_{i=1}^{N}\left({\Delta_{i_x}}^2 + {\Delta_{i_y}}^2\right) \qquad \text{Sum of error squared} \qquad (1)$$

$$\Delta_{i_x} = \left[X_i - X_R - d_i * \sin(\theta_R + \phi_i)\right] \qquad (2)$$

$$\Delta_{i_y} = \left[Y_i - Y_R - d_i * \cos(\theta_R + \phi_i)\right] \qquad (3)$$

$$f_i = \frac{1}{2}\frac{\partial}{\partial X_R}E = -\Delta_{i_x} \qquad (4)$$

$$g_i = \frac{1}{2}\frac{\partial}{\partial Y_R}E = -\Delta_{i_y} \qquad (5)$$

$$h_i = \frac{1}{2}\frac{\partial}{\partial \theta_R}E = -d_i\cos(\theta_R + \phi_i)\Delta_{i_x} + d_i\sin(\theta_R + \phi_i)\Delta_{i_y} \qquad (6)$$

$$\nabla E = \begin{bmatrix} \frac{1}{2}\frac{\partial}{\partial X_R}E \\ \frac{1}{2}\frac{\partial}{\partial Y_R}E \\ \frac{1}{2}\frac{\partial}{\partial \theta_R}E \end{bmatrix} = \begin{bmatrix} f_i \\ g_i \\ h_i \end{bmatrix} \qquad (7)$$

$$\mathrm{H} = \begin{bmatrix} \frac{\partial}{\partial X_R}\frac{\partial}{\partial X_R}E & \frac{\partial}{\partial Y_R}\frac{\partial}{\partial X_R}E & \frac{\partial}{\partial \theta_R}\frac{\partial}{\partial X_R}E \\ \frac{\partial}{\partial X_R}\frac{\partial}{\partial Y_R}E & \frac{\partial}{\partial Y_R}\frac{\partial}{\partial Y_R}E & \frac{\partial}{\partial \theta_R}\frac{\partial}{\partial Y_R}E \\ \frac{\partial}{\partial X_R}\frac{\partial}{\partial \theta_R}E & \frac{\partial}{\partial Y_R}\frac{\partial}{\partial \theta_R}E & \frac{\partial}{\partial \theta_R}\frac{\partial}{\partial \theta_R}E \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathrm{f_i}}{\partial \mathrm{X_R}} & \frac{\partial \mathrm{f_i}}{\partial \mathrm{Y_R}} & \frac{\partial \mathrm{f_i}}{\partial \theta_\mathrm{R}} \\ \frac{\partial \mathrm{g_i}}{\partial \mathrm{X_R}} & \frac{\partial \mathrm{g_i}}{\partial \mathrm{Y_R}} & \frac{\partial \mathrm{g_i}}{\partial \theta_\mathrm{R}} \\ \frac{\partial h_\mathrm{i}}{\partial \mathrm{X_R}} & \frac{\partial h_\mathrm{i}}{\partial \mathrm{Y_R}} & \frac{\partial h_\mathrm{i}}{\partial \theta_\mathrm{R}} \end{bmatrix} \quad (8)$$

$$\mathrm{H}_{11} = \frac{\partial}{\partial X_R}\frac{\partial}{\partial X_R}E = \frac{\partial \mathrm{f_i}}{\partial \mathrm{X_R}} = \sum_{i=1}^{N} 1 \quad (9)$$

$$\mathrm{H}_{12} = \frac{\partial}{\partial Y_R}\frac{\partial}{\partial X_R}E = \frac{\partial \mathrm{f_i}}{\partial \mathrm{Y_R}} = \sum_{i=1}^{N} 0 \quad (10)$$

$$\mathrm{H}_{13} = \frac{\partial}{\partial \theta_R}\frac{\partial}{\partial X_R}E = \frac{\partial \mathrm{f_i}}{\partial \theta_\mathrm{R}} = \sum_{i=1}^{N} d_i \cos(\theta_R + \phi_i) \quad (11)$$

$$\mathrm{H}_{21} = \frac{\partial}{\partial X_R}\frac{\partial}{\partial Y_R}E = \frac{\partial \mathrm{g_i}}{\partial \mathrm{X_R}} = \sum_{i=1}^{N} 0 \quad (12)$$

$$\mathrm{H}_{22} = \frac{\partial}{\partial Y_R}\frac{\partial}{\partial Y_R}E = \frac{\partial \mathrm{g_i}}{\partial \mathrm{Y_R}} = \sum_{i=1}^{N} 1 \quad (13)$$

$$\mathrm{H}_{23} = \frac{\partial}{\partial \theta_R}\frac{\partial}{\partial Y_R}E = \frac{\partial \mathrm{g_i}}{\partial \theta_\mathrm{R}} = -\sum_{i=1}^{N} d_i \sin(\theta_R + \phi_i) \quad (14)$$

$$\mathrm{H}_{31} = \frac{\partial}{\partial X_R}\frac{\partial}{\partial \theta_R}E = \frac{\partial \mathrm{h_i}}{\partial \mathrm{X_R}} = \sum_{i=1}^{N} d_i \cos(\theta_R + \phi_i) \quad (15)$$

$$\mathrm{H}_{32} = \frac{\partial}{\partial Y_R}\frac{\partial}{\partial \theta_R}E = \frac{\partial \mathrm{h_i}}{\partial \mathrm{Y_R}} = -\sum_{i=1}^{N} d_i \sin(\theta_R + \phi_i) \quad (16)$$

$$\mathrm{H}_{33} = \frac{\partial}{\partial \theta_R}\frac{\partial}{\partial \theta_R}E = \frac{\partial \mathrm{h_i}}{\partial \theta_\mathrm{R}} = \sum_{i=1}^{N} {d_i}^2 [\cos^2(\theta_R + \phi_i) + \sin^2(\theta_R + \phi_i)] = \sum_{i=1}^{N} {d_i}^2 \quad (17)$$

$$\mathrm{H} = \begin{bmatrix} N + \lambda & 0 & \sum_{i=1}^{N} d_i \cos(\theta_R + \phi_i) \\ 0 & N + \lambda & -\sum_{i=1}^{N} d_i \sin(\theta_R + \phi_i) \\ \sum_{i=1}^{N} d_i \cos(\theta_R + \phi_i) & -\sum_{i=1}^{N} d_i \sin(\theta_R + \phi_i) & (\sum_{i=1}^{N} {d_i}^2) + \lambda \end{bmatrix} \quad (18)$$

After calculation of H, updated location and heading of the robot are calculated with (19) & (20).

$$\begin{bmatrix} X_{R+1} \\ Y_{R+1} \\ \theta_{R+1} \end{bmatrix} = \begin{bmatrix} X_R \\ Y_R \\ \theta_R \end{bmatrix} + \mathrm{H}^{-1} \nabla E \quad (19)$$

$$\begin{bmatrix} \Delta \mathrm{X} \\ \Delta \mathrm{Y} \\ \Delta \theta \end{bmatrix} = -\mathrm{H}^{-1} \nabla E \quad (20)$$