Design Report
University of Michigan Dearborn
Ohm 5.0

Date Submitted: 5/24/2017

I, Dr. Samir Rawashdeh of the Department of Electrical and Computer Engineering at the University of Michigan Dearborn, certify that the design and development of this fifth iteration of the Ohm vehicle by the individuals on the design team is significant, unique to this iteration of the vehicle, and is equivalent to what might be awarded credit in a senior design course.

X S. Rawashdeh

ABSTRACT

This paper presents Ohm 5.0, a robot designed and used by the University of Michigan - Dearborn for the 25th Annual Intelligent Ground Vehicle Competition (IGVC). Ohm 5.0 used in the 2017 competition is based off of the platform used in previous competitions. There are minor changes to the hardware platform, including a GPS mast for the new differential GPS, and routine maintenance. Ohm, the software platform has been completely overhauled. New to Ohm 5.0 is LIDAR and Vision sensor fusion using an occupancy *mapping* technique and a *path planning* algorithm. The Robot Operating System (ROS) framework was also adopted. Topics such as design procedure, design improvements, safety measures and protocols, and control systems will be discussed.

INTRODUCTION

The Intelligent Systems Club of the University of Michigan - Dearborn has entered the 2017 Intelligent Ground Vehicle Competition with a majority of members returning from 2016 and 2 new members. The main goal of this year's team is to learn and establish an understanding in Robot Operating System (ROS), sensor fusion, and advanced mapping and path planning techniques to improve the overall efficiency of the robot and to mature the team's and club's knowledge of various robotics concepts. This year's strategy is to learn from previous competition entries, utilize key successes from the already existing robot platform, and learn how to improve on weaknesses from previous failures.

The team consists of all undergraduate students, and many plan to participate in future competitions. The team member composition is displayed in Table 1.

Table 1. Team Ohm Composition

Name	Email	Class	Role
Samir Rawashdeh	srawa@umich.edu	Assistant Professor, Electrical and Computer Engineering	Faculty Advisor
Siddharth Mahimkar	smahmka@umich.edu	Computer Engineering, Junior	Captain
Michael Bowyer	mbowyer@umich.edu	Electrical and Computer Engineering, Senior	Software & Electrical
Brendan Ferracciolo	bferracc@umich.edu	Computer Engineering, Junior	Software
Matthew Abraham	mjabraha@umich.edu	Computer Science, Junior	Software
Daniel Vanden Berg	djvanden@umich.edu	Electrical Engineering, Junior	Electrical, Mechanical
Benjamin DiDonato	bddidona@umich.edu	Electrical and Computer Engineering, Junior	Software
Cristian Adam	cradam@umich.edu	Computer Science, Senior	Software
Kenneth Yesh	kyesh@umich.edu	Electrical and Computer Engineering, Senior	Software
Emmanuel Obi	ecobi@umich.edu	Software Engineering, Junior	Software

Saad Pandit	snpandit@umich.edu	Industrial Engineering, Junior	Mechanical
Brendon Bergstresser	brbergst@umich.edu	Electrical Engineering, Junior	Software

DESIGN INNOVATIONS

This year’s team intended to improve on previous year’s accomplishments by redesigning the vehicle's main software platform, and replacing/adding sensors in areas of need. Table 2 describes the areas which needed improvement and why, as well as what was completed to improve the vehicle. Table 3 describes the cost. The remainder of this report will discuss these improvements and how they were implemented.

Table 2. Design Innovations and Reasoning

Areas to be Improved or Added	Reason for Improvement or Addition	Improvement Design
New Software Architecture	Allows for easier integration, testing, mapping planning, and sensor fusion	Use Robot Operating System (ROS)
Path Planning	Path Planning was never done before	Use ROS and sensor fusion
Differential GPS	Improved heading accuracy. Last year robot would loose heading when moving slowly.	Design mast to house GPS. Software to take advantage of the heading accuracy improvement.
Simulation	Did not have any simulation environment, sometimes the best way to decide between two alternate options is to test both, but physical testing is resource intensive.	Created simulated competition environment in Unity, including simulated camera and LiDAR sensor on robot.
Processor Compartment Ventilation	Previous design did not allow for the processor to be sufficiently cooled in warm weather.	Ventilation fans were added, LED strip, switch added
Electrical design	Needed new wireless e-stop. Prevent center tapping batteries, and improved safety stop circuit.	Use of DC-DC and DC-AC converters. Utilize longer range wireless emergency stop.

Table 3: Vehicle Cost

Electrical Component Category	Price
Sensors	$12.259.97
Processor Cost	$1,139.02
Battery Cost	$368.00
Misc. Electrical	$322.14
Total Electrical Cost:	$14,089.13

Mechanical Component Category	Price
Frame/Assembly	$260.00
Drivetrain Cost	$1,181.98
Misc. Mechanical	$311.98
Total Mechanical Cost:	$1,753.96

Overall Category	Price
Electrical	$9,089.13
Mechanical	$1,753.96
Total Robot Price:	**$15,843.93**

MECHANICAL DESIGN

The vehicle used for this year's competition is one that has been around the University of Michigan-Dearborn for some time now. It has participated in numerous IGVC and Autonomous snowplow competitions in the past. The vehicle is made primarily of wood, and uses a differential drive steering control scheme which is aided by a trailing caster. This year the main changes of the mechanical design involved new sensor mounts and maintenance. The CAD model of the robot is shown in Figure 1a as well as a photo of the vehicle on the title page of this report..

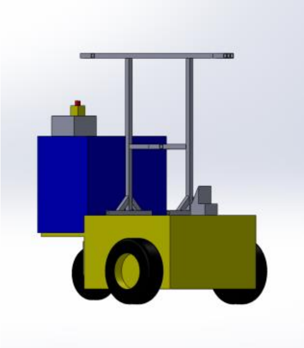

Figure 1a. Robot design

Decision on frame structure, housing, structure design

The mechanical design base of the robot was originally designed for the Autonomous Snowplow Competition in 2010, and has been repurposed for the past few years for the Intelligent Ground Vehicle Competition. The robot is made almost entirely of wood with four long metal threaded poles. The robot has four pieces of plywood, each which act as a level within the robot. The metal poles are placed vertically and threaded through each piece of plywood. Each level is secured to the metal poles using two nuts to hold each level in place.

The robot utilizes a single rear caster and two side mounted drive wheels. There is a camera and GPS mast mounted in the center of the front top platform for elevation. The elevation is needed for better GPS reception and the ability for the camera to look downward to detect lines and potholes. Batteries are housed in the center of the lower level to provide a low center of gravity. 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 bolted to the lowest level of the robot. The single caster is also bolted to the middle level of the robot and allows for almost zero degree turns.

A newly added gps mount for the differential gps was added this year. Constructed out of 1” hollow square aluminium tube, 40” long and 37” tall, this allows each of the GPS antenna to be mounted at opposite ends thereby giving an accurate heading, and higher off the ground to give the receivers better reception. Figure 1b shows the CAD model of the GPS mast.

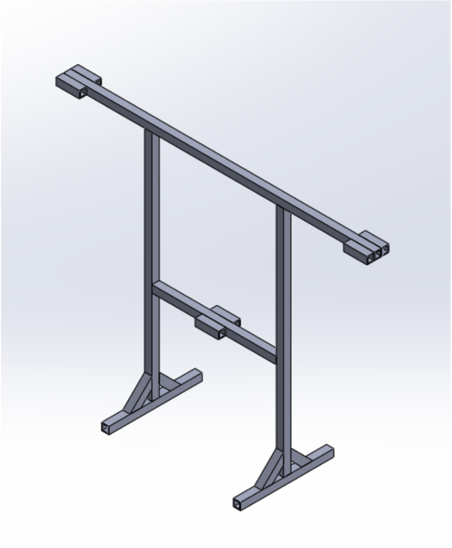

Figure 1b. GPS mast

Additionally a payload tray has added as well as well as silicon caulk. The payload tray was added so that the payload will be able to sit securely without the need for bungee cords. Silicone caulk was added around the laptop control box to prevent water from entering the main electronicas area of the vehicle.

ELECTRICAL COMPONENTS AND DESIGN

The electrical design for the robot has some important changes but is not entirely different from the past year. Improvements to the electrical systems has been implemented, such as safety, efficiency, robustness, and prolonging the battery life. Figure 2 shows the electrical schematic of the entire robot.

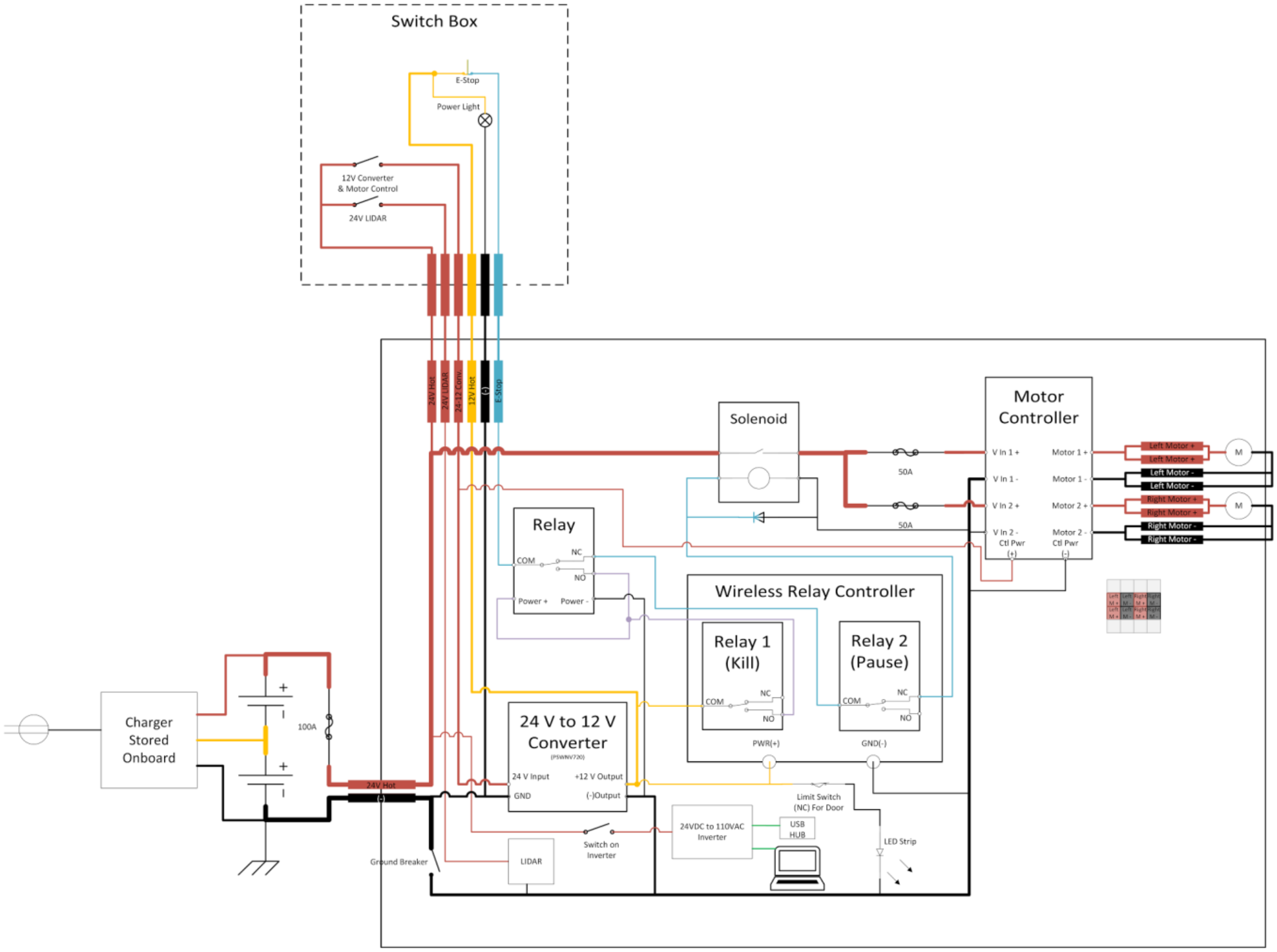

Figure 2 Electrical Diagram

POWER DISTRIBUTION SYSTEM

As part of routine maintenance, two new Yellow Top Optima 12V batteries in series for a total of 24V and 55 amp-hours. The second major change to help contribute to extending battery life is instead of center-tapping the two series 12V batteries to get access to 12V, a PYLE Audio 24V DC to 12V DC Power Step Down Converter (PSWNV720) was added. The loss by using a converter is less significant than the impact of center tapping one of the 12V batteries. It helps prolong the life of the batteries by not discharging them at different rates. The 24/12V converter is fused to 30A to protect the 12V components.

A ProMariner battery charger was also added to the robot to make charging the batteries easier. It individually charges each battery using 12V, and just needs to be plugged into an AC outlet, which easily extends from the inside of the vehicle.

Another small but efficient change was adding a small limit switch to control the internal LED lights. The internal lights are 24 Watts when on, which was a large contributor to overheating issues in the past.. When the door to the control box is closed, the lights turn off, but turn on when opened. This not only helps keep the overheating issue, but also helps conserve more power. To help prevent overheating of the laptop and electronics, a small DC fan was added to help cool the control box. With these changes, idle current draw was around 3A, and an average driving (on grass, uphill) current usage of approximately 14A, thus giving us a runtime of just under 4 hours with our 55AH batteries.

SAFETY SYSTEM

To improve the safety of our robot, a new wireless relay control system manufactured by National Control Devices was installed to work in tandem with a physical emergency stop button to control a solenoid. The manufacturer of this device claims it has a range of 750 feet. We have tested it reliably out to 250 feet. The relay directly controls the solenoid which controls power to the motors, which when removed causes the vehicle to come to a halt.

The wireless relay remotes have two buttons that are configured into two modes: pause and kill. “Pause Mode” acts like a toggle switch to temporarily disable the robot. It is intended for just keeping the robot stationary when there is no dangerous situations, such as testing or when setting up the robot. Pressing the button again will re-enable the robot. “Kill Mode” on the other hand will instantly shut off the solenoid and will lock itself in. In order to re-enable the robot, the physical emergency stop button must be cycled. This is intended for preventing dangerous situations. In either case, if the 12V master switch, or power is removed from the wireless relay module, power will immediately be removed from the motors, causing the vehicle to come to a halt.

PROCESSOR

Ohm uses a Lenovo Thinkpad X260. This laptop was used in previous years due to its robustness, small form factor, and durability. It uses an i5-6300U cpu, 8GB ram, and is dual booted with Windows 10 Pro and Ubuntu 16.04. During competition Ubuntu 16.04 is used. The processor is the main interface between sensors, and actuators. The processor on board takes input from the sensors and wireless controller, and controls the actuators accordingly

GPS

This year Ohm will utilize the VectorNav VN-300 differential GPS. This system is used because of increased accuracy and very high heading accuracy. Along with better accuracy, the system also has a built IMU which it uses in conjunction with a built in Kalman filter to prevent jumps in heading and position. This is a very large improvement from last year’s single point GPS which consistently had heading accuracy issues. This is a large improvement which was absolutely necessary as the mapping approach used this year, which will be discussed later on. The GPS only task is simply to estimate the position and heading of the vehicle.

LiDAR

Ohm uses a SICK LMS-111 Lidar. This LiDAR is used due of its high reliability and accuracy. This LiDAR has been used in multiple different competitions, in various weather conditions (such as snow). The scan range of the LiDAR is 20 meters, scans at a frequency of 25HZ with an angular resolution of 0.25°, and has 270° field of view. The main purpose of the LiDAR is to determine where surrounding obstacles and objects are in the vehicle's environment. This information is then stored in an occupancy map, which will be discussed later on.

CAMERA

A wide angle camera was selected to assist the vehicle detect the lanes, potholes, and obstacles.. The Logitech C525 HD Webcam was selected because of its wide view capabilities with a viewing angle of 120° . This allowed for lines on the side of the robot to be seen when the camera was placed on the GPS mast and angled downward. This proved to be useful as many trials caused the robot to leave the lanes due to the inability to see lines on the side of the robot. This also allowed the robot to see potholes near its wheels, and to continue to avoid them until the robot had passed the pothole. This year the camera’s images are scanned to find the location of the lanes, potholes, and obstacles with respect to the robot. Using the robot's current location (estimated by GPS and dead reckoning) the location of the lanes, potholes, and obstacles, are added to the occupancy map.

WHEEL ENCODERS:

The main purpose of these wheel encoders is to improve accuracy and confidence in the vehicle's absolute position in the field. In previous competitions, the vehicle was only equipped with one position sensor (a GPS), and the data provided by that sensor was assumed to be true. This lead to unpredictable behavior of the vehicle due to variations in the sensor data. Although the GPS sensor used this year is much improved compared to previous competitions, better position estimation is still desired. This is where the wheel encoders are used, as they make dead reckoning a possibility. The end goal is to obtain a more accurate position estimation of the vehicle by comparing dead reckoning and GPS position estimations than either estimation on its own. This is of utmost importance this year because of the occupancy map which is being built for the path planner. In order to build an accurate map, an accurate position of the vehicle is required.

To obtain a position estimation using dead reckoning the distance a wheel travelled is required. The wheel encoders do just that, they simply return the amount of rotation each wheel experiences. This information is used to determine the distance each wheel has traveled, because the gear ratio and circumference of the wheels is known. Once that information is determined, the position of the position and heading change is estimated. Once the estimation of the position and heading is completed, it is compared with the GPS position, and filtering of the position estimations are completed in order to estimate a more precise position of the vehicle, with the the end result being that the robot's position and heading estimations will have improved accuracy when compared to using dead reckoning or GPS estimation alone.

SOFTWARE DESIGN

Our codebase last year was one monolithic project, but this year we have overhauled it to work with the Robot Operating System (ROS) architecture. This means all major parts of the code are split into separate, independent pieces called nodes. This allows us to better work on the code in parallel and more easily test and swap out parts of the code. We have also developed a new strategy for navigating obstacles: rather than move reactively as the robot encounters obstacles it is about to hit, Ohm tries to use all the current information to more intelligently plan its route a few meters in front of it by updating an occupancy map. Figure 3 shows the software diagram and the various nodes which are responsible for the tasks which their title describes.

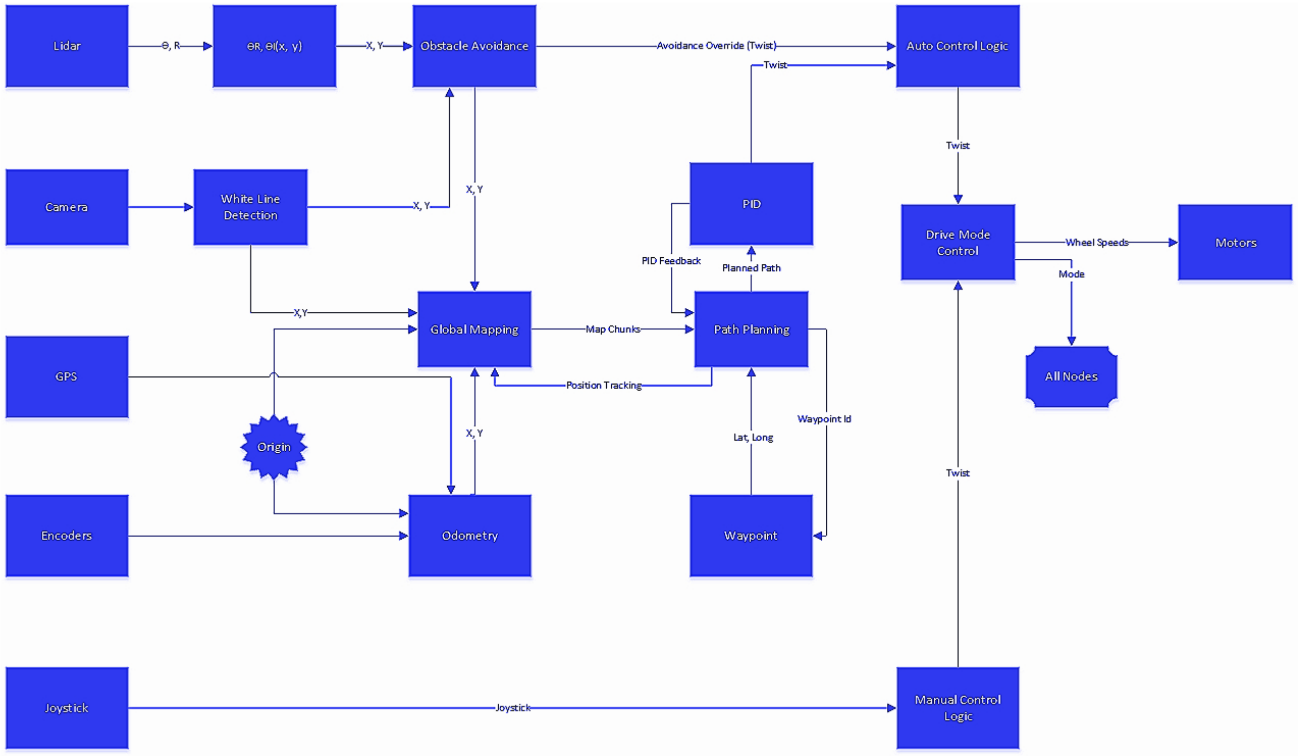

Figure 3: Software Architecture Diagram

SOFTWARE STRATEGY AND PATH PLANNING

The aforementioned map is used by the path planner to find the best route through the obstacles. This can only be completed by updating the map with what the vehicle's sensors sense. By determining obstacles positions with respect to the vehicle, and then utilizing the vehicle's estimated position, a global map is created. Once the map has been updated with the surroundings, Ohm is able to optimize its route around the obstacles within its map. This allows for earlier avoidance of obstacles once a map has been built, instead of solely reacting to the presence of obstacles, lanes, and potholes which was the strategy of previous competitions. .

LOCALIZATION AND MAPPING

LiDAR and camera data are interpreted as binary images, and combined in a map that can be saved in any common image format. Because of this combination, objects seen by the LiDAR and camera can both be considered obstacles. The map itself is a simple, fixed size occupancy grid. Points gathered from sensors are associated with a given grid square and count towards its occupancy until it reaches a given threshold. Once a consistent number of occupying obstacles/lanes are seen in that grid, the grid is considered occupied.

All software components return data points in a frame relative to the vehicle, and then added to the global map by considering the vehicle's GPS position and heading. The global map mentioned is in XY coordinate format, where an arbitrary GPS point is chosen as the origin of the XY field. Map coordinates are defined in reference to a point in world coordinates representing the left-top of an image. Only points that are also legal map coordinates will be saved by the map.

OBSTACLE AND LANE DETECTION

Ohm uses both a LiDAR and camera to detect obstacles. Both the white lines and the barrels are considered obstacles. The LiDAR returns an array of points with a distance and an angle relative to the robot. These points are placed in the map. The camera detects objects by first doing a perspective transform to effectively give a top down view of the area it sees. Since the field can be considered flat, it is safe to assume that an object at a given pixel coordinate has an associated distance. Once the calibration is done, any pixel that is white is put into the map by passing its pixel coordinate into a line equation and solving for the distance in meters, then those distances are sent to the map. Line equations are shown below.

$$\square = \square * \square\square + \square$$

$$\square = \square * \square\square + \square$$

Where ***U*** and ***V*** are the X and Y distance respectively in meters relative to the robot. ***a,b,c,d*** are constants. ***Xp*** and ***Yp*** are the pixel coordinates. Figure 3 shows before and after perspective transform and figure 4 shows lane detection.

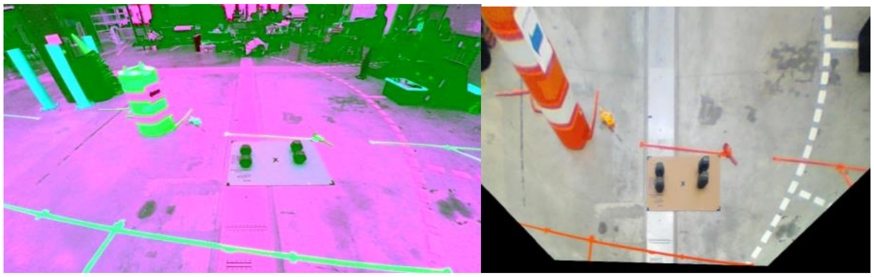

Figure 3:Before and After

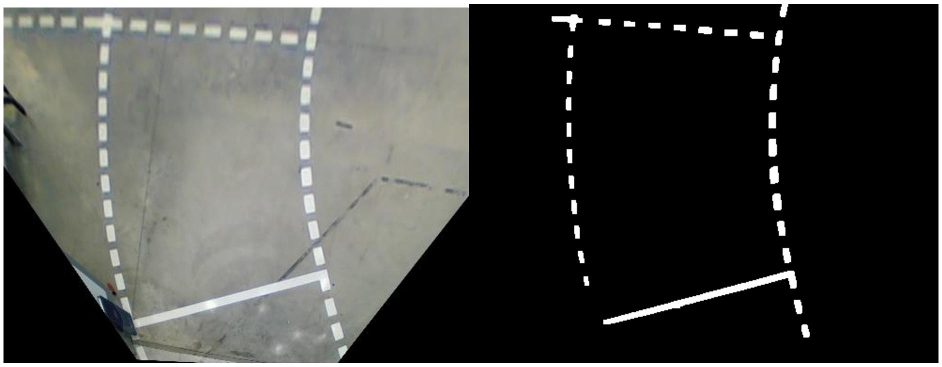

Figure 4: Lane detection

GOAL SELECTION AND PATH PLANNING

For waypoint navigation, an ordered list of GPS targets is loaded a priori into the program, and the path planner selects the first waypoint in the list. This GPS waypoint is then converted to the maps XY coordinates. From there, the path planner loads portions of the global map, known as the local map of the overall map to apply the A* pathfinding algorithm. The path planner has a set distance it can plan ahead before it must stop and place an intermediate waypoint. This distance is configurable and can be be made higher or lower based on confidence in the map. Once the path planner is within a few meters of the waypoint, it begins planning towards the next waypoint.

SIMULATIONS

A simulation environment was created in Unity which incorporates LiDAR and camera sensor simulation data. A rough approximation of the course was obtained via Google Maps satellite imagery which showed the white line markers, upon which we added obstacles such as the construction barrels based on videos and images from prior competitions as far as the layout. This environment is being used for testing various map-building and traversal algorithms, especially for proof-of-concept ideas using ideal sensors, which will then allow us to focus our efforts with real-world testing on working algorithms for path planning, and tuning them to work with the various physical limitations of our sensors such as drift and noise.

RISKS AND FAILURE MITIGATION

The main points of failure for the vehicle described in the previous sections as well as mitigation actions to combat these failures is summarized in Table 4.

Table 4. Failure Modes Likelihood and Countermeasures

Possible Failures	Failure Likelihood	Failure severity	Countermeasures
Laptop Power Failure	Low	End of run	Charge laptop during run in case battery is knocked loose.
Battery Power Failure	Low	End of run	Battery should be charged prior to each run.
Sensor Connector Disconnect	Low	End of run likely	Secure all sensor connectors, ensure each sensor is working prior to each run.
Blowing Fuse on Motor Power Lines	Low	End of run	Testing to ensure that robot will not exceed current rating of fuses.
Sensor Failure	Low	End of run likely	Software which compensates for lost sensor data stream.
Emergency Stop Failure	Very Low	Possible damage/injury to robot/personnel	Software and hardware emergency stop options have been implemented.
Motor Mount	Low	End of Competition	Drive slow on bumps, no rash driving, metal base plate to attach motors

SUMMARY

Table 5 show the performance summary of Ohm 5.0

Category	Analysis	Countermeasure
Speed	Maximum Speed is roughly ~6 MPH, and can be limited using software.	Speed will be fine tuned at qualification and will only be lowered if necessary.
Ramp	Ability to climb 30° on wet surface is a risk. If Ramp is taller than scanning plan of LiDAR, Ohm 5.0 will think the ramp is an obstacle.	LiDAR can be placed at various heights. Ramp incline could prove fatal as original robot design did not consider inclines.
Reaction Times	Worst Case: a GPS data update is ~5 Hz. For LiDAR(Vision) is 25 Hz(10 Hz)	The Map will update with data as it becomes available
Battery Life	Approximately 4 hours under moderate stress.	Charge laptop using external battery, turn off lights, charge batteries every opportunity.
Distance of Obstacle Detection	Maximum obstacle detection with LiDAR is 20m away. Normally limited to 10m. Camera can effectively see only 3 m in front and 1m on either side	Change distance as needed to fine tune control.
Accuracy of Waypoint Arrival	Entirely variable through software.	Edit target reached threshold as GPS accuracy deteriorates.
Dealing with tricky situations	Dead ends can be overcome by backing up or making zero degree turn. Different obstacles/tasks in single situation occasionally cause robot to be unable to make decision.	Robot has backup algorithm when at a dead end or is unable to to make a decision.

Table 5: Performance Summary

This year's competition is all about developing and learning the foundations in mapping and path planning along with maturing the team's and club's knowledge in ROS. Team Ohm has done just that with Ohm 5.0. By learning the techniques and methods for mapping the course, the team gained valuable insight necessary to establish a solid foundation for future iterations of the robot.

ACKNOWLEDGMENTS:

The team would like to thank Dr. Samir Rawashdeh, Dr. Michael Putty, and Dr. Lin Van Nieuwstadt for providing valuable insight in all aspects of the design and implementation of the robot. We would like to thank Miles Johnson a research scientist at Toyota for providing insight and helping us understand mapping and path planning concepts. Lastly we thank the Open Source Robotics Foundation who are the maintainers of the ROS framework.