OHM (Ω) MK-IV

University Of Michigan-Dearborn

Team Members: Michael Bowyer, Siddharth Mahimkar, Brendan Ferracciolo, Matthew Abraham, Saad Pandit, Emmanuel Obi, Zachary Woolston, Ken Yesh, Erik Aitken, Ryan Sheen

Advisor: Samir Rawashdeh

Table of Contents

1 Abstract
2 Introduction
3 Design Process
- 3.1 Design Improvements
- 3.2 Vehicle Cost

4 MECHANICAL DESIGN
- 4.1 Additions and Existing Chassis
- 4.2 Redesigned Wiring and Sensor Location

5 ELECTRONIC COMPONENTS AND DESIGN
- 5.1 Electrical Component Selection
  - 5.1.1 Batteries
  - 5.1.2 Motor Controller
  - 5.1.3 LiDAR
  - 5.1.4 GPS
  - 5.1.5 Wiznet Serial to Ethernet
  - 5.1.6 Arduino and Safety Light
  - 5.1.7 Camera
  - 5.1.8 The Laptop
- 5.2 Wiring Schematic
- 5.3 Safety Considerations

6 Software Platform and Strategy
- 6.1 GPS and Waypoint Navigation
- 6.2 Camera and Vision Techniques
- 6.3 Laser Range Finder and Obstacle Avoidance
- 6.4 Mapping
- 6.5 Debugging and Simulation

7 PERFORMANCE ANALYSIS
8 Conclusion

1 Abstract

This paper presents Ohm MK-IV, a robot designed and used by the University of Michigan - Dearborn for the 24th Annual Intelligent Ground Vehicle Competition (IGVC). Ohm MK-IV used in the 2016 competition is based off of the platform used in previous competitions. There are significant changes to the hardware platform, including a new processor housing unit. The software platform from previous competitions still remains in Ohm MK-IV, but new vision based algorithms and decision-making methods have been changed. Topics such as design procedure, design improvements, safety measures and protocols, and control systems will be discussed.

2 Introduction

The Intelligent Systems Club of the University of Michigan - Dearborn has entered the 2016 Intelligent Ground Vehicle Competition with an almost entirely new team, with only two team members returning from the competition in 2015. This fact paired with a new faculty adviser, who is unfamiliar with the competition resulted in a change in team goal from previous years. The main goal of this year’s team is to learn and establish an understanding of key techniques to improve in future competitions with the new members. 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 entirely of Undergraduates this year, and many plan to participate in future competitions. The team make up is displayed in Table 1.

Name	Major	Graduation
Michael Bowyer	Undergraduate Electrical & Computer Engineering	April 2017
Ken Yesh	Undergraduate Computer Engineering	April 2017
Brendan Ferracciolo	Undergraduate Computer Engineering	April 2019
Ryan Sheen	Mechanical Engineering	December 2017
Erik Aitken	Software Engineering	December 2016
Siddharth Mahimkar	Computer Engineering	December 2018
Mathew Abraham	Computer and Information Science	December 2018
Saad Pandit	Industrial Engineering	April 2020
Zachary Woolston	Electrical Engineering	April 2019
Emmanuel Obi	Computer and Information Science	April 2019

Table 1: Team Member Composition

3 Design Process

3.1 Design Improvements

Designing additions revolved around improving the already existing robot platform from previous competitions. The first step of designing these additions was identifying where Ohm MK-III could be improved and be made into Ohm MK-IV. The following table shows the areas of improvement or which needed to be implemented and how the team decided to make additions and edits. The sensors and electrical components used in previous competitions remained the same other than the motor controller.

Areas to be Improved or Added	Reason for Improvement or Addition	Improvement Design
Overcoming Ramp Obstacles	LiDAR placement at last year's competition led to inability to overcome ramps in competition.	Change LiDAR Position; make various heights available to mount LiDAR.
Improving Vision	Many runs in last year's competition led to failure because of failure to recognize lines in shadows, and bright day light.	Re-design line detection Algorithm so it can adapt to changing light conditions.
Adding in Pot Hole Avoidance	New addition, not implemented in previous competitions. Necessary for this year's Competition.	Identify Potholes using Vision.
GPS Reception	GPS reception around buildings and trees was poor.	Move GPS to highest point on robot.
Waterproofing	Any amount of rain could damage electronics due to lack of waterproofing.	Waterproof spray paint on wooden parts of robot, and add weather stripping.
Laptop Compartment	Previous design did not allow for CPU to be housed, and was stored on top of robot.	Design Compartment to store main CPU (laptop), so it is safe, and easy to access.
Flag path	Was not tested at previous competition.	Validate Flag detection and improve if necessary.
Wiring Organization	Small electrical changes could take hours to fix due to lack of organization.	Create removable Wiring Shelf for quick access using disconnects

Table 2: Design Improvements

3.2 Vehicle Cost

A majority of Ohm MK-IV's cost is in its electrical system. Over 90% of the overall cost is spent on electrical equipment. Most of this cost is in the sensors used. Most of the sensor price itself is due to the expensive laser range finder which costs ~$7,000.

The mechanical design of the robot is relatively cheap. The majority of this price comes from the drive train. This includes the motors, wheels, and the single caster. If a cheaper motor was chosen then this price could be reduced even further as each of the two motors costs ~$450. The category and overall prices of Ohm MK-IV can be seen in the following table.

Electrical Component Category	Price
Sensors	$7.259.97
Processor Cost	$1,139.02
Battery Cost	$368.00
Misc. Electrical	$322.14
Total Electrical Cost:	$9,089.13

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:	**$10,843.93**

Table 3: Vehicle Costs

4 MECHANICAL DESIGN

4.1 Additions and Existing Chassis

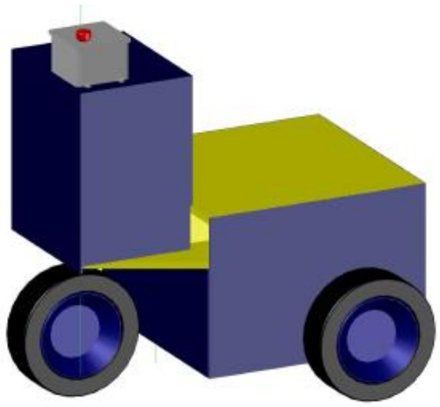

Figure 1: 3D Model of Ohm MK-IV

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 levels of 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 itself is a single caster and two drive wheel system. 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 and center of gravity. The vehicle is propelled by two 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.

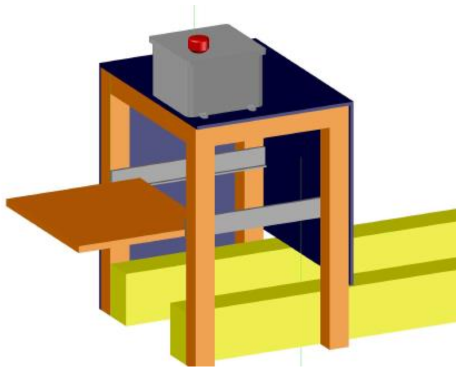

Figure 2: 3D Model of laptop compartment addition

The newly added compartment for the laptop features a sliding tray, which will make it easy to place and remove the laptop from inside of the robot, while also allowing for the robot to be reprogrammed comfortable in the field. The location of the emergency stop box has been moved to the top of the laptop box so that it is readily accessible and easy to see from any angle around the robot.

Figure 1 shows the 3D model of the base of Ohm MK-IV. The addition to the robot itself is the blue compartment on the back of the robot in the figure. This compartment was added to store the main laptop used in the robot, and create more space for electrical components. The emergency stop box was placed on top of this compartment. The inside of the compartment is pictured in Figure 2.

4.2 Redesigned Wiring and Sensor Location

This year a new wiring system was designed and the placement of all of the electrical equipment was changed. All of the wiring was designed to be placed on a removable board, making all electronic components easily visible, repairable, and modifiable. This made any maintenance tasks easier without having to take apart the entire robot. The new wiring system was equipped with many quick disconnects to allow for the wiring board, with all electrical components on it to be quickly removed, and worked on at a workbench instead of on the robot itself.

This year the LiDAR can now have its position adjusted by height so that the robot would be able to change its plane to see a variety of obstacles at different heights. In last year's run, the LiDAR was directly mounted onto one of the wooden panels on the robot. It was originally placed 18" from the ground. One of last year's obstacles was a 2' tall ramp. Unfortunately, the LiDAR mistook the ramp as a wall and the robot was unable to overcome it. This year's adjustable LiDAR should be able to prevent a problem like that from happening again.

The GPS unit's location was also changed. Previously, it was on the middle level of the robot, and occasionally received poor signal due to its height. It is now mounted on the top of the robot's camera mast this year so that the GPS can a better signal.

5 ELECTRONIC COMPONENTS AND DESIGN

5.1 Electrical Component Selection

5.1.1 Batteries

The mechanical design of the battery compartment allows for a maximum of two 12V Lead Acid Optima Yellowtop batteries, both 55Ah. The batteries are connected in series to provide the 24V supply to both the motors and to the LiDAR. The 12V output from the first battery is used to supply power to the 12V sensors and a 12V 55W inverter. The inverter also charged the battery of the onboard laptop when necessary. A 12V to 5V converter was also used to power a USB hub for USB peripherals.

5.1.2 Motor Controller

The motor controller selected for use in Ohm MK-IV was the RoboteQ MDC2460. This motor controller is capable of supplying 60A output per channel, up to two channels. The motor controller has many communications capabilities; the one chosen was RS232 Communication. The motor controller command latency from command submission was at most 1ms. The controller communicated using a baud rate of 115200, and was sent speed commands to the motors in open loop fashion. In autonomous mode, the PID guidance used acted as closed loop control when in use.

5.1.3 LiDAR

The main sensor used for obstacle avoidance was a laser range finder produced by SICK, the LMS111. With an Angular step width of .25° resolution the LiDAR scans a field of view of 270°. The rotational frequency of the spinning laser in the LiDAR is 50 Hz. The LiDAR utilizes TCP/IP communication over Ethernet to provide the main program a structure of data points from each sampled angular frequency. The high price tag for the sensor was justifiable as it was incredibly reliable, but the only drawback to the sensor is its boot up time when powered on. This can delay the start of an autonomous run up to 45 seconds if not powered on prior to the run.

5.1.4 GPS

The GPS used for Waypoint navigation was a Garmin 19x HVS. One of the major design changes to the GPS receiver from previous competitions was the placement. The GPS was placed on top of the Camera Mast this year. This was a risky decision, as the GPS could get wet if it were to rain while outside. But this GPS is waterproof up to 1 meter, so this design change was approved, and improved GPS reception.

The GPS was capable of an update rate of 1 Hz. This meant that the best-case time scenario for GPS updates was every second. The GPS also provided heading when a change was detected from previously recorded coordinates. Although, to get an accurate heading, the robot needed to be moving to continuously or else the GPS unit may think the robot is spinning in place.

The GPS communicated over RS232 standards with a baud rate of 38400.It was found that the main program of the autonomous navigation sometimes used old GPS data when using RS232 due to the FIFO used in serial communication. To overcome this, a WizNet Serial to Ethernet converter was used to retrieve the most recent GPS information.

5.1.5 Wiznet Serial to Ethernet

A WizNet serial to Ethernet converter was used for GPS communication. The Wiznet continuously reads in serial data, and creates a string of read in serial commands. Once the WizNet is queried by the main program, the entire string of data is sent to the main program, and the main program only uses the most recent data in the string to guide itself to waypoints. This prevents the main program from using old data, which was retrieve and placed into the serial FIFO.

5.1.6 Arduino and Safety Light

To meet the requirement for a flashing light when in autonomous mode, and solid when in manual mode, an Arduino Micro-controller was used to control the safety light. The safety light, along with the Arduino, was placed in the emergency stop box on top of the robot. The main program would send a command to the Arduino over RS232 communication standards whenever the main program changed states from autonomous, to manual control or vice versa. When the state was changed, the Arduino would control an electromagnetic relay, which would connect or disconnect 12V to control pin on the safety light. The change of state in the relay would change the state of the safety light to either flashing or solid. The Arduino is powered off of a USB A to USB B Cable, using the laptop as the main power source.

5.1.7 Camera

A wide-angle camera was selected to stay in lanes and avoid potholes. The Logitech C525 HD Webcam was selected because of its wide view capabilities. This allowed for lines on the side of the robot to be seen when the camera was placed on top of the camera mast. This proved to be useful as many trials caused the robot to leave the lanes due to the inability to see lines near 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.

5.1.8 The Laptop

Previous competition's codebase was written in C and compiled on laptop computers using utilities, such as OpenCV for image processing. Since many of the core functions still remained from the previous code base, a laptop was selected again for this year's competition instead of an onboard microprocessor running some form of a real time operating system.

There were several requirements for the laptop; the first was the size requirement. The newly constructed laptop compartment with the sliding tray required the laptop to be less than 12" deep, and under 15" wide in order to fit in the compartment with enough space for the peripherals to be plugged into the laptop. The laptop also needed to be rugged in the case of dropping, as it will be in the field often. So the laptop selected had a magnesium roll cage within it. The Laptop needed to have at least 4GB of RAM as the utilities and main program on average take up about 3GB of RAM. For safety, a laptop with 8GB of RAM was selected.

5.2 Wiring Schematic

The main wiring schematic was compiled after all of the known sensors and equipment, which would be used from the previous section, were confirmed. The main control of the electrical power is all routed through the emergency stop box (E-stop box). There are switches in this box to control the 12V and 24V LiDAR circuits inside of the robot, along with the safety light, which was controlled by an Arduino. The USB hub, which the USB cable to communicate with the Arduino was powered from the 12V to 5V down converter. The motor controller Serial to USB cable and the remote control transceiver dongle were also plugged into this USB hub. Other main sensors were powered off of the 12V auxiliary (or switched) power. Figure 3, shown below, displays the wiring schematic for the robot.

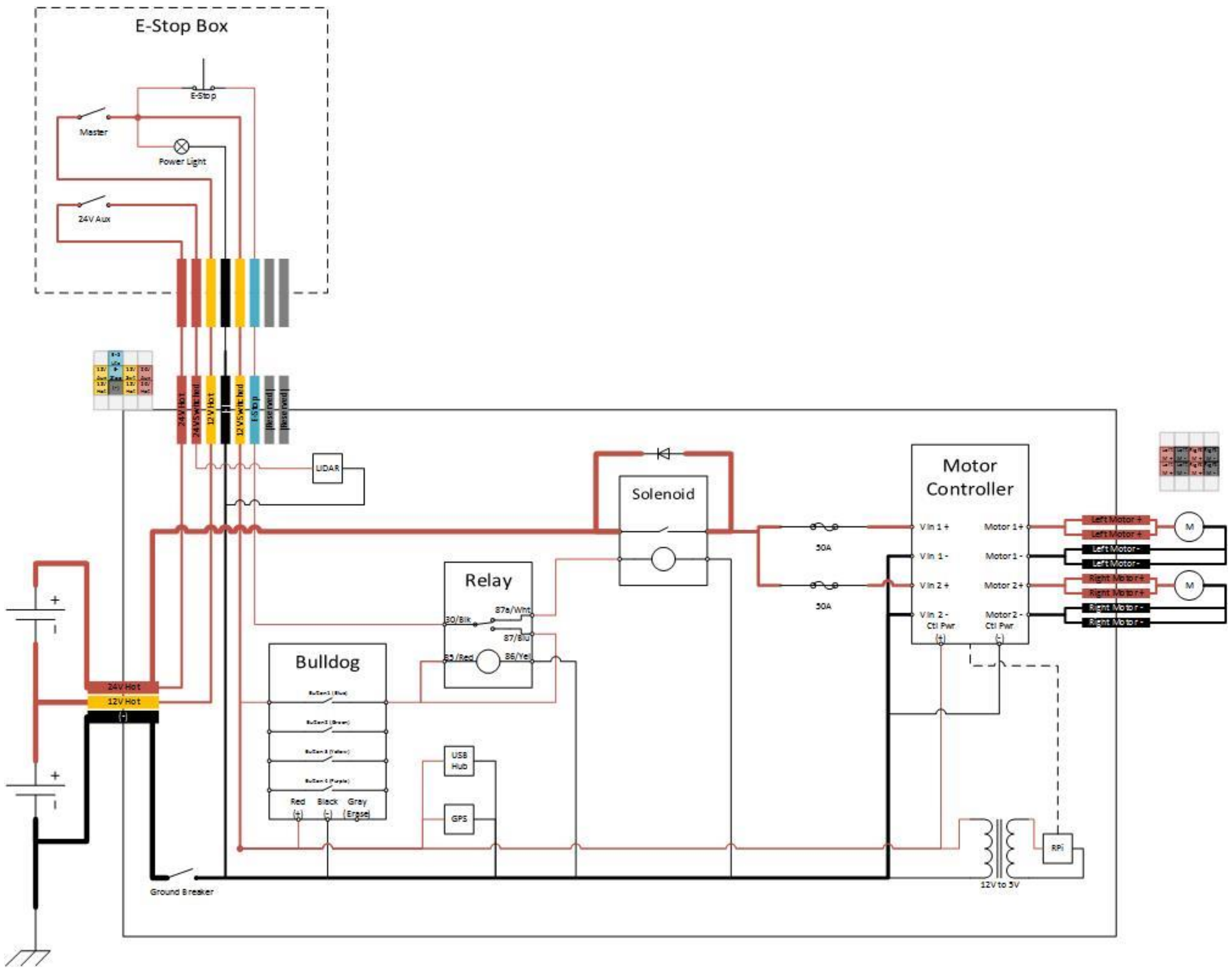

Figure 3: Wiring Schematic

5.3 Safety Considerations

To provide safety when working on and operating the robot many safety precautions were taken. The safest way to work on electrical equipment inside of the robot is to remove any “hot” wires, which are connected directly to the batteries and provide voltage. To overcome this, quick disconnects were utilized to remove the main wiring platform entirely from the batteries. This allowed for the main electrical components to be entirely removed from any power, making it entirely safe to work on. A ground breaker was added into the main ground to prevent any current flow through the robot when small mechanical additions were being made, such as mounting sensors, and removing the plywood top level of the robot.

Another safety consideration was the remote emergency stop function. To ensure the robot could stop from a distance, a wireless momentary switch was utilized. The momentary switch is titled “Bulldog” in Figure 3; this is because the product name is “Bulldog Remote Anything”. The momentary switch used was designed to trigger a self-latching safety circuit, when triggered. As seen in the schematic in Section 5.2, the relay used will permanently latch to output 87 causing power to be removed from the motor channels of the motor controller by disengaging the normally open solenoid. This circuit continues to be latched until power is removed from it. Other electrical equipment, such as fuses, was placed on various electrical components to prevent damage to the equipment, and prevent possible fires.

6 Software Platform and Strategy

The already-existing software platform was built off of an accumulated set of libraries built using C. The each library is used for either retrieving data from a sensor, or algorithms performed on the data retrieved from sensors. The main program is where decisions are made according to the algorithm results.

Our control system required two states: manual and autonomous operations. In these two states, the motor controller of the robot would receive commands from one of two places, an Xbox 360 gamepad for manual operation, and the commands sent to it by the main program that ran the autonomous code. While in autonomous mode, Ohm MK-IV uses the flowchart in Figure 4 to decide which direction to turn.

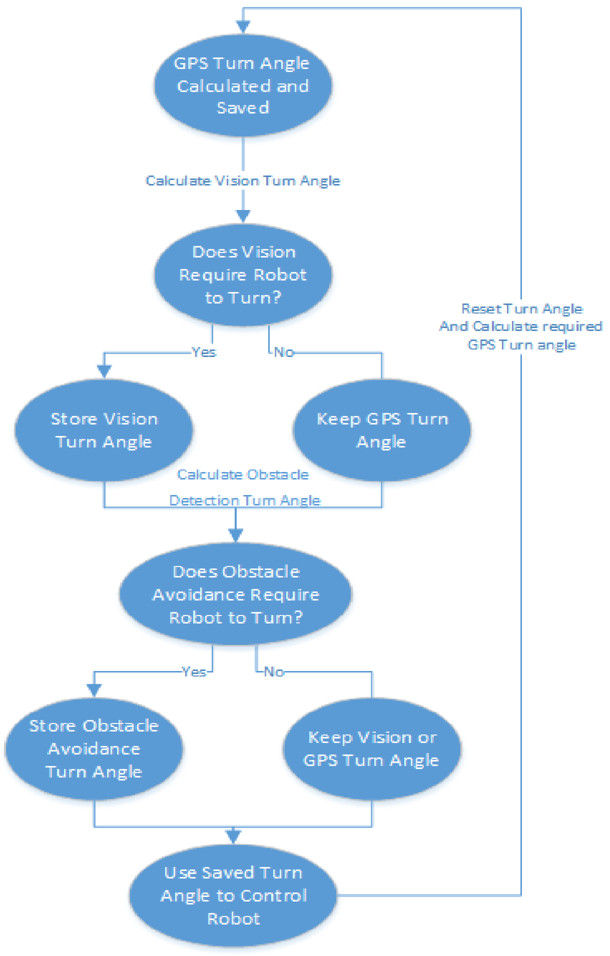
GPS Turn Angle Calculated and Saved
Calculate Vision Turn Angle
Does Vision Require Robot to Turn?
Yes
No
Store Vision Turn Angle
Keep GPS Turn Angle
Calculate Obstacle Detection Turn Angle
Does Obstacle Avoidance Require Robot to Turn?
Yes
No
Store Obstacle Avoidance Turn Angle
Keep Vision or GPS Turn Angle
Use Saved Turn Angle to Control Robot
Reset Turn Angle And Calculate required GPS Turn angle

Figure 4: Decision Making Flowchart

6.1 GPS and Waypoint Navigation

The techniques used for waypoint navigation required first converting the robot's location with respect to the course starting point into XY coordinates. Each following waypoint was calculated in a similar fashion transforming the latitude and longitude coordinates to XY in regard to the starting coordinate, which was set to (0,0).

Once each coordinate was calculated into XY coordinates, the first step in guidance was to calculate an arc from the current robot's location to the next waypoint. Once the arc was calculated the turn angle required to make the arc was calculated. The arc guidance technique is displayed in Figure 5.

The calculated turn angle was then passed into a PID controller to smooth out the robot’s guidance path. The PID also helped account for accumulated error from turning to avoid obstacles and stay within lanes. Each time a waypoint was reached the PID controller variables were reset to prevent compensation from previously accumulated error.

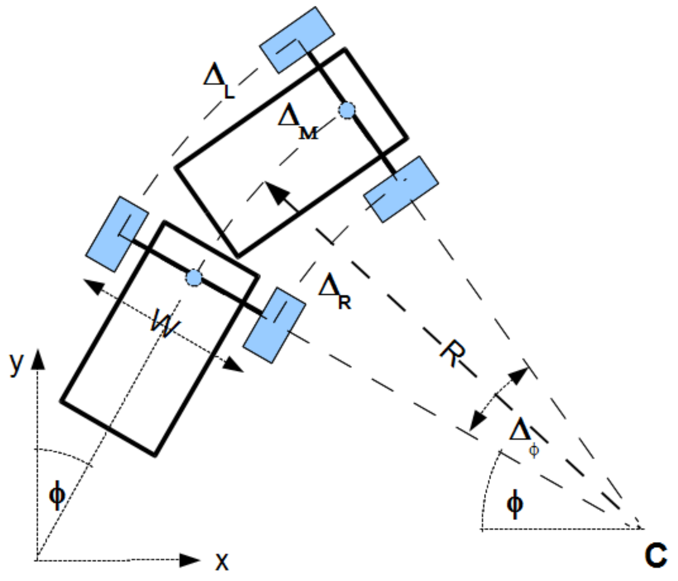

Figure 5: Arc Waypoint Navigation

6.2 Camera and Vision Techniques

There were main three functions a camera was used for: detecting lanes, detecting flag lanes, and detecting potholes. The wide-angle camera allowed for ease of use when detecting the lanes. Each function of the vision strategy began with capturing an image using the webcam, and then filtering the image by selecting pixels from the captured image from the webcam which fit a certain criteria of color. The criteria of color would be altered for the various functions.

For lane tracking function the image is searched for white pixels, and turns any pixels found to be mostly white to be totally white, and any non-white pixels to be black. The filtered image is then searched for the two strongest lines on each side of the image. Another line is then drawn exactly in between the two found lines. The slope of this line is then converted into a turn angle and used to control the robot.

The pothole avoidance technique combines the filtering of an image for white pixels, and then detecting any large circles within the image. If there are large circles detected, then two turn angles are calculated to avoid the pothole. This technique is very similar to the one described in 6.3 to avoid obstacles.

To avoid flags, the same technique is used to stay within lanes. The strongest red line, and strongest blue line are selected from the image. The turn angle is then calculated by finding the line exactly in between the two extracted red and blue lines.

6.3 Laser Range Finder and Obstacle Avoidance

The strategy utilized to avoid obstacles was to manipulate the data read in from the LiDAR using TCP/IP communication. The LiDAR data structure is an array of a total of 1082 data points all holding the distance measure at a distinct angle within the scanning field. The LiDAR data is then filtered using a method known as using a “Halo”. The purpose of the Halo is to determine how far ahead to robot should begin to avoid obstacles. When a LiDAR data point is outside of the distance defined by the Halo, that data point is ignored. The Halo’s distance around the robot can be changed.

With the remaining data set, it is first calculated that if the robot must make a turn. To explain further, if the robot is to go perfectly straight, will the robot run into an obstacle? If not, then no turn needs to be calculated. If the robot will hit an obstacle, then the robot will calculate two turn paths to get around the obstacles. This concept is shown in Figure 6.

To calculate these two turn angles, three functions are used. “getBufferAngles” is the function to determine the two turn angles required to avoid an obstacle. Once the two turn angles are calculated, the left and right wheel scan functions convert the turn angles into motor control commands. The “checkAngle” function confirms if the robot can make the calculated turn according to the motor control commands. The call graph of these functions can be seen in Figure 7.

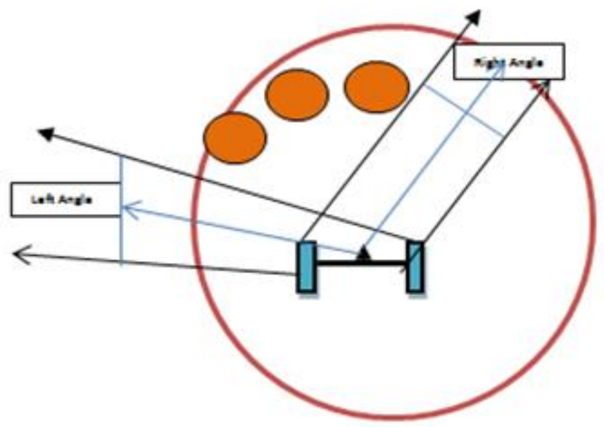

Figure 6: Calculating two turn angles Diagram

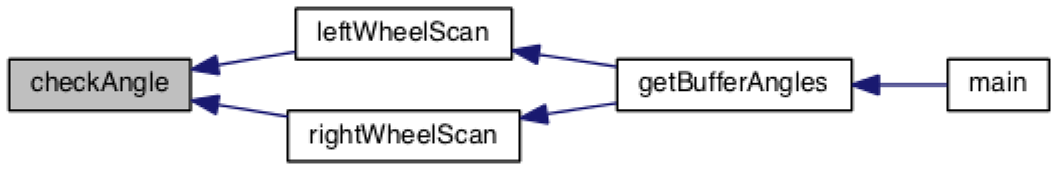

Figure 7: Determining and Checking the Calculated Turn Angles

6.4 Mapping

Previous mapping techniques proved to be too rigorous to attempt to alter and improve on given the amount of time the team had before competition. Due to this, it was decided that mapping would not be attempted at this year's competition.

6.5 Debugging and Simulation

In order to debug software while testing it was essential to have some sort of simulation, instead of testing on the robot itself. This was accomplished by using an application titled “Virtual Robot” which was created by our past faculty advisor. The virtual robot grabbed an image from Google Maps at the entered starting GPS location, and then monitored a serial port for motor control commands. When proper motor control command was seen the virtual robot would begin to turn a displayed robot in XY coordinates, and then change the GPS location on the grabbed Google Maps image. A screenshot of the simulation can be seen in Figure 8.

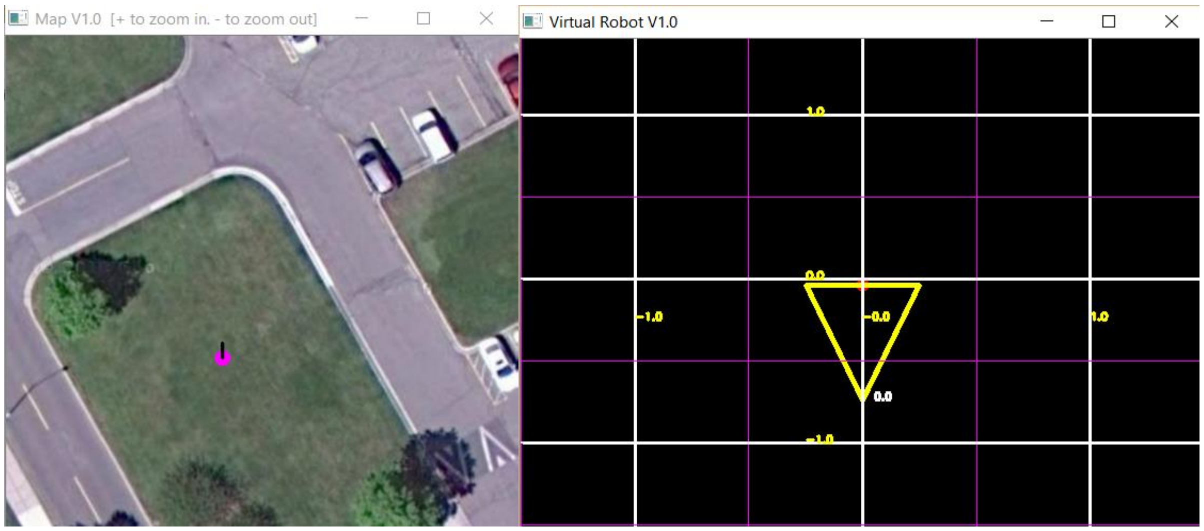

Figure 8: Virtual Robot Simulation Application

While testing on the robot it was difficult to fine tune navigation parameters by following the robot around. The code could be tested quickly and remotely by using a Wiznet Serial-to-Ethernet adapter to relay commands to the motor controller over an established wireless network from an onboard router. This saved time having to setup the robot to be driven after making changes to the code.

7 Performance Analysis

The following table was produced when analyzing how well the robot could perform in various situations. It was found that the robot was able to complete each task individually fairly well. That is, when only avoiding an obstacle, or staying in between lines, the robot was able to complete its task. But when introducing two or more tasks into a single scenario the robot sometimes struggled.

Category	Analysis	Counter Measure
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 MK-IV 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 ~1 seconds. For LiDAR and vision is ~2 seconds at worst case.	When no white lines or circles are detected, skip vision-mapping algorithm. Also occurs when no obstacles are detected for LiDAR.

Battery Life	Batteries can provide normal continuous usage for just under 2 hours. Approximately 4 hours under moderate stress.	Charge laptop using external battery, charge batteries every opportunity.
Distance of Obstacle Detection	Maximum obstacle detection is 20m away. This is normally limited to about 10m.	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 make a decision.

Table 4: Performance Analysis Table

Another analysis category investigated by the team was the possible failures that could take place during a run. The purpose of this task was to assess various risks during a run. Each failure is identified in the following table, along with the likelihood, severity, and any counter measures that could be taken to offset the failure.

Possible Failures	**Failure Likelihood**	**Failure severity**	**Counter Measures**
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	Moderate	End of run likely	Secure any 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 that compensates for lost sensor data stream.
Emergency Stop Failure	Very Low	Possible damage to robot or injury	Software and hardware emergency stop options have been implemented.

Table 5: Risk Assessment and Countermeasures.

8 Conclusion

This year’s competition is all about learning and team Ohm MK-IV has done just that. By learning the techniques and methods used in previous competitions, the new members have learned all of the necessary techniques to establish a solid foundation for future competitions.

OHM MK-IV

2016 Intelligent Ground Vehicle Competition

DEARBORN

ISC

Intelligent Systems Club

University of Michigan - Dearborn
Dearborn, Michigan 48128, USA

May 15th, 2016

Team Members: Michael Bowyer, Siddharth Mahimkar, Brendan Ferracciolo, Matthew Abraham, Saad Pandit, Emmanuel Obi, Zachary Woolston, Ken Yesh,Erik Aitken, Ryan Sheen

Advisor: Samir Rawashdeh

I hereby certify that the design and engineering of OHM MK-III for the 2016 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.

S. Rawashdeh 5-13-2016

Dr. Samir Rawashdeh