IGVC 2017

Bob Jones University

LAZARUS

Date submitted: May 15, 2017

Team Captain: Nathan Woehr, Nathan.woehr@gmail.com

Team Members:

Fleet Belknap	fbelk122@students.bju.edu	Austin Kim	hkim929@students.bju.edu
Maverick Cowland	mcowl664@students.bju.edu	John Smoker	jsmok127@students.bju.edu
Sevrin Dyer	sdyer891@students.bju.edu	Brandon Woods	bwood254@students.bju.edu

Instructor: Will Woodham, M.S.P.D., wwoodham@bju.edu

(See attached statement of integrity)

1. Introduction

Lazarus is the latest incarnation in a line of autonomous ground vehicles designed to compete in the Intelligent Ground Vehicle Competition (IGVC). It utilizes a sensor array consisting of LIDAR, a digital compass, a digital camera, and a GPS to help the robot navigate and see obstacles in its path. Lazarus solves many of the problems of its predecessor. The overhauled body design lowers the center of gravity, and reduces the overall weight of the robot. The redesigned power system not only reduces the weight of the robot, but also cleans up the interior space of the robot and improves operational safety.

2. Design

2.1 Design Teams & Organization

The 2017 robotics team divided into three sub-teams for working on the robot: Mechanical Design, Electrical Design, and Software. There were several factors that determined the team's focus for work this year. The robot was clumsy, top-heavy, and was overweight. In response, each team focused on a few of these areas.

Table 1: Student Contribution

Team Members	Academic Department & Class	Sub-Team	Hours Invested
Nathan Woehr, Captain	Engineering, Senior	Software	44
Maverick Cowland	Engineering, Freshman	Software	110
Austin Kim	Computer Science, Freshman	Software	45
Fleet Belknap	Engineering, Freshman	Electrical	141
Brandon Woods	Engineering, Junior	Mechanical	155
Sevrin Dyer	Engineering, Junior	Mechanical	45
John Smoker	Engineering, Junior	Mechanical	16

2.2 Design Goals

For this year, we chose to redesign Kezia, our robot from last year, to minimize costs while still improving upon a few key aspects of our robot.

These were our goals for the redesign:

- Maintain the required specifications to compete in IGVC 2017
- Mechanical
    - Reduce overall weight of the robot
    - Lower robot's the center of gravity
    - Maintain the same ground clearance
    - Improve the traction of our drive wheels
    - Replace the omni-wheel with something that works well in a grassy field
    - Do not allow water to contact the electronics in any potentially damaging way
    - Maintain acceptable aesthetics
- Electrical
    - Reduce the complexity of the electrical systems
        - Design new power system
        - Layout wires in a controlled wiring harness
        - Label all wires
    - Create a wiring schematic for future teams to study
- Software
    - Implement a new LIDAR module
    - Fix issues when the camera sees clover

2.3 Design Process

Whenever a decision needed to be made, we first decided what we thought the root of the issue was then brainstormed possible solutions to fix the issue. After brainstorming we chose the best solution based on the goals outlined above.

2.3.1 Mechanical Design Process

For Lazarus's mechanical design, we modeled everything in SolidWorks first, using as many of Kezia's components as possible. Lazarus required quite a few new components to be fabricated or old components modified, which we did mostly in our engineering lab with the equipment we have available. However, we do not have access to any welders, so the 4 components that required welding we had to outsource to the campus weld shop. For these components, we created drawings in SolidWorks and sent those drawings to the weld shop.

2.3.2 Electrical Design Process

When we were looking at the overall performance of the previous robot, one big area that we saw for improvement was in the complexity of the power system. We decided to take a simplistic approach. We calculated what voltage and current was needed for each system and designed our new system from those requirements. Thus, Lazarus has a much smaller, lighter, and more efficient power system than its predecessor.

2.3.3 Software Design Process

For this iteration of the software the software team wanted to improve the image processing code and add in driver support for the new LIDAR hardware. Since we had to rewrite the LIDAR software, we decided to try and move all our sensors out of the main loop and into their own parallel loops. This provides better leverage of LabVIEW's parallel execution support allowing each sensor to work as fast as it can without slowing down the main loop's reaction to data.

3. Innovations

Table 2: Innovations

Mechanical	Electrical	Software
Payload Placement	Rapid Replace Battery System	Low Cost LIDAR Integration
Drive Wheels	Wire Strain Relief	

3.1 Payload Placement

The new payload placement utilizes nylon straps to suspend the payload securely beneath the robot. The system is secure, lowers the center of gravity, and allows easy enough access for replacing the payload (*Figure 1*). To maintain the same ground clearance, we chose to increase the diameter of the new drive wheels to 16".

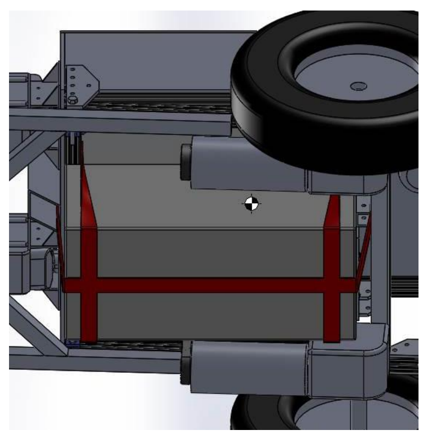

Figure 1: Payload Placement

3.2 Drive Wheels

We chose to switch wheels and tires to maximize traction. We attempted to do this by minimizing the contact patch under the drive wheels. This, while counterintuitive, increases the contact pressure which forces the rubber to conform more to the surface it is rolling across, generating more contact forces than frictional forces. Since we needed 16" tires for ground clearance we could not decrease contact area by reducing the tire radius, so we chose the skinniest tires we could find with decent tread.

3.3 Rapid Replace Battery System

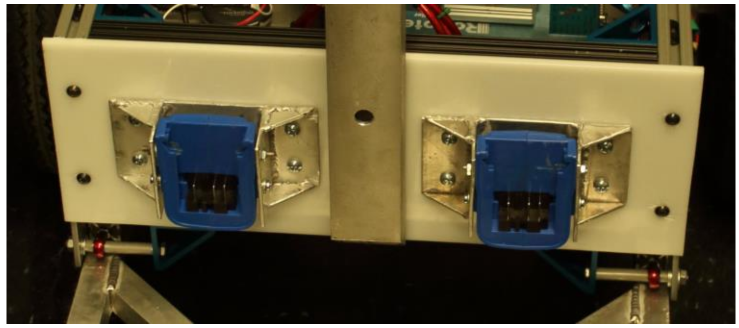

Figure 2: Quick release adapter plates

One problem with the previous robot was the placement of the batteries. They were stowed beneath the floor of the robot, and it required partial disassembly of the robot to access them. The rapid replace battery system utilizes Kobalt 24-volt power tool batteries for high power capacity in a small package, light weight, very quick battery changing, and rapid recharging capability. These batteries are located on the rear of the robot for easy access not only for battery swaps but also for emergency removal of the batteries if it is ever necessary. The batteries fit into quick release adapter plates that were fashioned from modified Kobalt flashlights. One internal upgrade that we did inside the adapter plates was change the power carrying wires from #22 AWG to #18 AWG. This small change increases the safety of the new system.

3.4 Wire Strain Relief

One challenge in previous robots was the problem of wires disconnecting due to vibration and parts moving in the system. We mitigated that risk in this robot by securing the wires into a cohesive wiring harness. Another area of failure is the fragile connectors that connect to the motor encoders. We designed and built brackets to make these connections rigid and reduce the probability of damage during use.

Figure 3: Motor encoder wiring protection bracket

3.5 Low Cost LIDAR Integration

The Hoyuko LIDAR was replaced by an Slamtec RPLIDAR A2. This was done to experiment with whether a low-cost LIDAR module could be safely integrated into an autonomous navigation system and still supply accurate data.

4. Other Improvements

4.1 Weight

The weight of the robot last year was 220.1 lbs. This year, the robot weighs 125.2 pounds, for a net decrease in weight of 94.9 pounds or 43%. *Table 3* details the weight changes, not including the miscellaneous fasteners, wires, and body panels because they were not finalized until too late.

Table 3: Items removed or added and their weights

Item Removed	Weight (pounds)	Item Added	Weight (pounds)
Generator	29.6		
Lead-Acid Batteries	27.2	Lithium Ion Batteries	4.0
Power Convertor	9.6	Voltage Convertor Board	0.025
Omni Wheel and rear suspension	15.4	Rear Caster Assembly	17.2
Old Drive Wheels	15.2	New Drive Wheels	18.0
Old LIDAR	1.4	New LIDAR	0.7

4.2 Center of Gravity

Every component of our robot was mounted at the lowest possible location allowed by the other design constraints. According to the calculations available in SolidWorks we lowered the center of gravity from 14.63 inches to 12.38 inches, 2.25 inches or 15.4% lower.

4.3 Rear Wheel

The previous robot utilized an omni-wheel for its rear wheel. While this appeared to be a great choice in the initial tests on pavement, it performed poorly when operating on grass. Our solution to that problem was a balanced rear caster wheel. This wheel is angled from subframe of the robot at a 3° angle to keep the caster mount level. This allows the rear caster to freely swivel in any direction.

4.4 Power System

The previous power system was a hybrid. It utilized a generator, batteries, charge controller, and inverter. While a hybridized system is advantageous in many ways, it dramatically inflated the weight of the robot. Since we were trying to lower the weight of the robot this year, we decided to revamp the entire power system. We removed the generator, lead-acid batteries, and power converter. The components removed weighed 66.4 pounds. We replaced it with a very simple battery system consisting of the Rapid Replace Battery System (see section 3.3), and a voltage convertor board for powering the USB hub. These components weigh a total of 4.025 pounds. (see *Table 3*)

4.5 Internal Sensor Mast Wiring

The Sensor Mast on the previous robot had all the wires routed outside the mast. While this allowed for the mast to be easily removed for transport, it exposed all the cables to the elements and had very little strain relief. We decided that we would trade the ability to fold the mast for better strain relief and protection from the elements.

5. Vehicle Description

5.1 Frame

The structural members, excluding fasteners, of Lazarus's frame are constructed entirely out of aluminum. The frame is constructed primarily out of 1" T-slot extrusion bars and 1" square tubing. Lazarus is dimensioned very close to the minimum width and length requirements. Most of the robot's mass sits very close to the ground, while the sensor mast extends to the maximum permitted height.

5.2 Power Analysis

Previous BJU robots were designed with a dual voltage system capable of supplying both 12 and 24 volts, and all components ran off one of these two voltages. Since Lazarus uses many of the same components from previous years, we decided to keep the basic dual voltage design. Calculations indicate that the electrical system would need to supply roughly 11 amps to drive at the max steady-state running conditions of 4 MPH on a 15% grade.

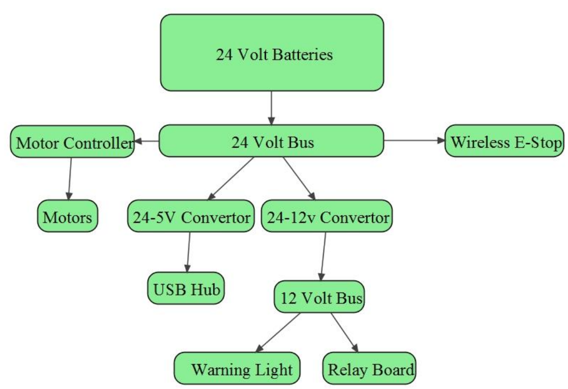

Figure 4: Power Flow Chart

5.3 Wiring

In any electrical system with many wires there is potential for messy wiring. To address this problem, wiring connections on Lazarus were made using Wago DIN rail terminal blocks. The blocks are in the electronics compartment at the front of the robot, and almost all electrical connections were made there. To easily identify the purpose of each wire, every wire was labeled with the corresponding function name. The function name also correlates with the wiring schematic.

5.4 Motor Controller

The motor controller is a Roboteq AX2850. It features dual channel motor control, allowing Lazarus to steer by sending different outputs to each of the two main wheels. The setting Lazarus employs is closed loop separate speed control. The motor controller also includes the E-Stop function used on Lazarus.

6. Autonomous System Design

6.1 Situational Awareness Design

6.1.1 LIDAR

To allow for increased accuracy in sensing the depth of obstacles in Lazarus's path, we are using a Slamtec RPLIDAR A2 LIDAR scanning laser sensor. This sensor sweeps a laser across a 220° arc, software limited, at rate of 10 Hz (600 RPM) to detect reflections off obstacles up to 6 meters away.

To maximize the usefulness for our new LIDAR unit we considered field of view, protection, and shading from direct sunlight. To shade it from direct sunlight we mounted the LIDAR unit under a panel on the front of the robot. This also protected it from weather and potential physical impacts. To maximize the field of view we mounted it as far forward on the plate as possible.

6.1.2 Camera

The Microsoft LifeCam Cinema gives Lazarus a reliable, compact camera input with a wide field of view and sufficient image quality to detect lines and flags. The camera remained at an altitude of about 5 feet 8 inches off the ground.

6.2 Auto-Navigation Design

6.2.1 Hardware

Lazarus's propulsion is provided by a National Power Chair R81 series motor attached to each of the front wheels, through a worm driven gearbox. The motors are controlled by a RoboteQ AX2850 Motor Controller set to closed loop separate control. The combined power draw of the motors is approximately in the range of 270-300 watts. (24V @ 12.5 A MAX)

6.2.2 Software

The software systems implemented in Lazarus's design were developed in National Instruments LabVIEW. LabVIEW is a visual programming language that makes use of a unique dataflow design structure. LabVIEW was used for all data manipulation, obstacle detection, waypoint navigation, and obstacle avoidance. An overview of the navigation algorithm is given below.

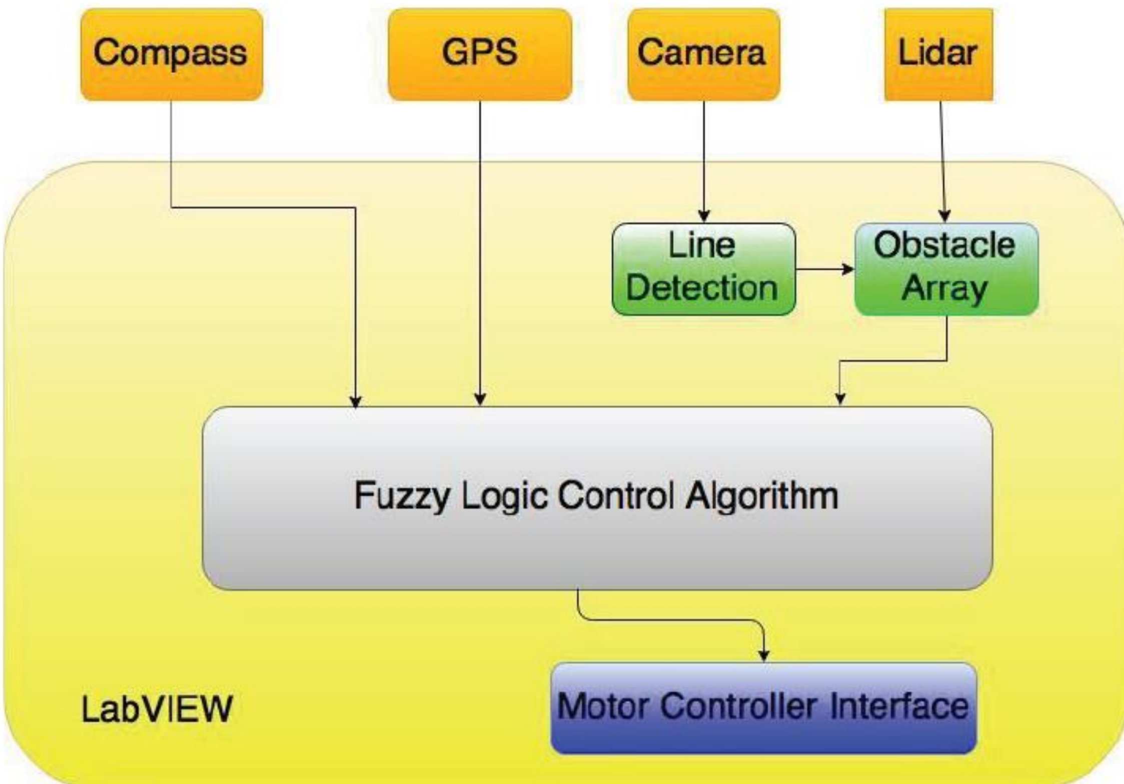

Figure 5: Navigation system block diagram

6.3 Navigation Strategy

6.3.1 General Mapping Strategy

Lazarus detects obstacles with two sensors: A camera for line detection and a LIDAR for solid object detection. Each obstacle is represented as a polar point whose origin is the robot's center, 0° points to the robot's right, and 90° points straight ahead. These two data sources are then combined

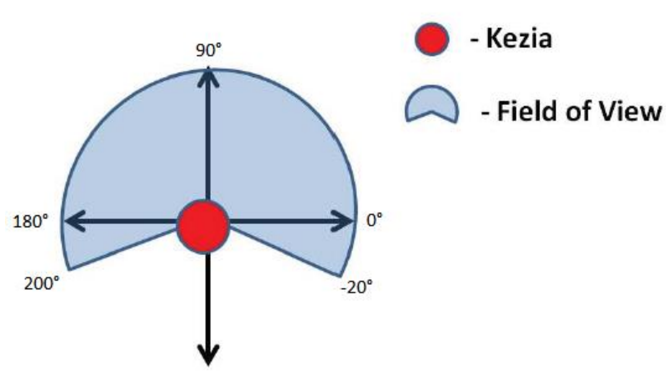

Figure 6: Field Mapping

into one "Obstacle Map" and passed to the path planning code. Path planning finds the closest obstacle and runs a *fuzzy logic* algorithm based on it. From this algorithm, we derive speed values for each motor which allow the robot to either make a turn or go straight. Since we have no rear facing sensors we do not allow the robot to drive backwards.

6.3.2 Solid Obstacle Detection

Solid obstacles are detected using a Slamtec LIDAR. Data from the LIDAR provides a distance measure for each degree of the scan range. The scan range is set to cover a 220° swath in front of Lazarus to prevent the front suspensions and main body from registering as obstacles. This list is then combine with the results from line detection and sent to the control algorithm as a single array of obstacle points.

6.3.3 Line Detection

Lines are detected using a Microsoft LifeCam web camera. Images from the camera are passed through two algorithms: Image Processing and Pixel-Distance Conversion.

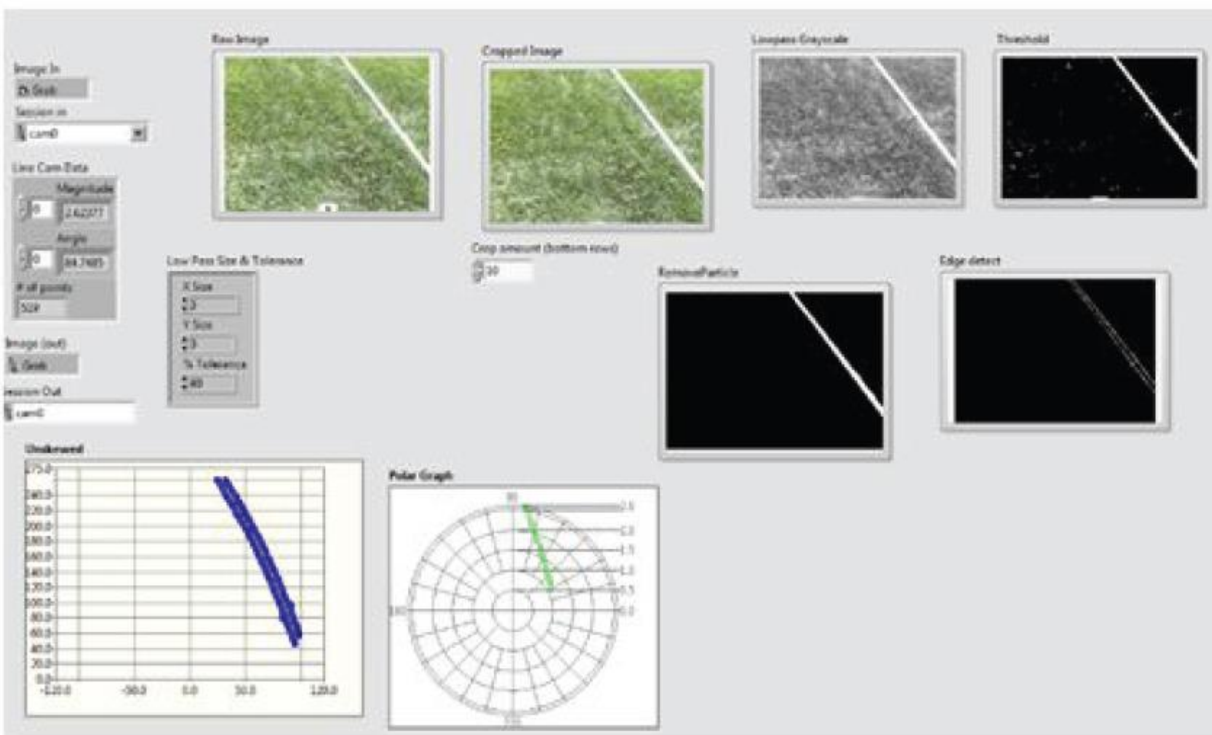

Figure 7: Image processing sequence and final mapped obstacle points

For image processing, white lines are extracted from the image by going through a series of steps: cropping, grayscale with low-pass filtering, mixed channel threshold, particle removal, and finally edge detection (see *Figure 7*). First, we crop the bottom of the raw image to exclude any sunlight reflecting off the robot's nose which can be mistaken for a white line. Next, we use a custom mixed-channel low-pass grayscale filter. After that, we grayscale the image because we have found that the blue components of a gray-scaled RGB image show the most contrast between grass and lane lines (see *Figure 8*). Then we send the image through a low-pass filter to blur noise particles from white to gray. Next we run a color threshold only based on the blue components of each pixel. Pixels below a defined threshold are set to black and ones that are above to white. This step is the most critical because we found a simple grayscale with a threshold accounting for all color components, even when not all thresholds are equal, is not enough for the noise reduction required for good results from the fuzzy logic algorithm. After thresholding, the image goes through a particle removal filter. Finally, we run edge detection on the image and extract an array of pixel locations from the original image.

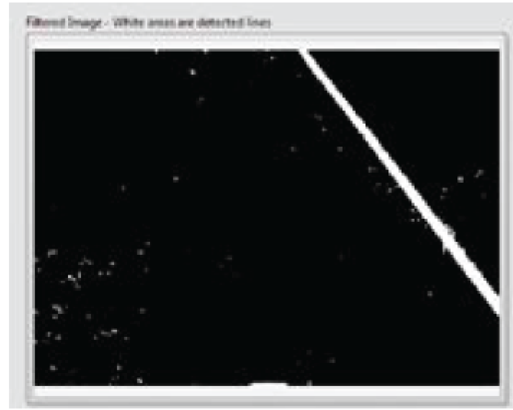

Figure 8: Thresholding based on blue pixel values only

Next the array of pixel locations is passed through the Pixel to Distance Conversion algorithm. This algorithm processes the pixel locations into polar points describing the real-world distance between the robot center and white lines. First due to the camera's slant, the pixels are skewed so that pixels along the image's bottom represent less distance than pixels along the top. We pass each pixel through two equations which convert them from a skewed to a plan view. One equation translates the rows into y-axis locations and the other translates columns into x-axis locations. The y-axis location depends only on the

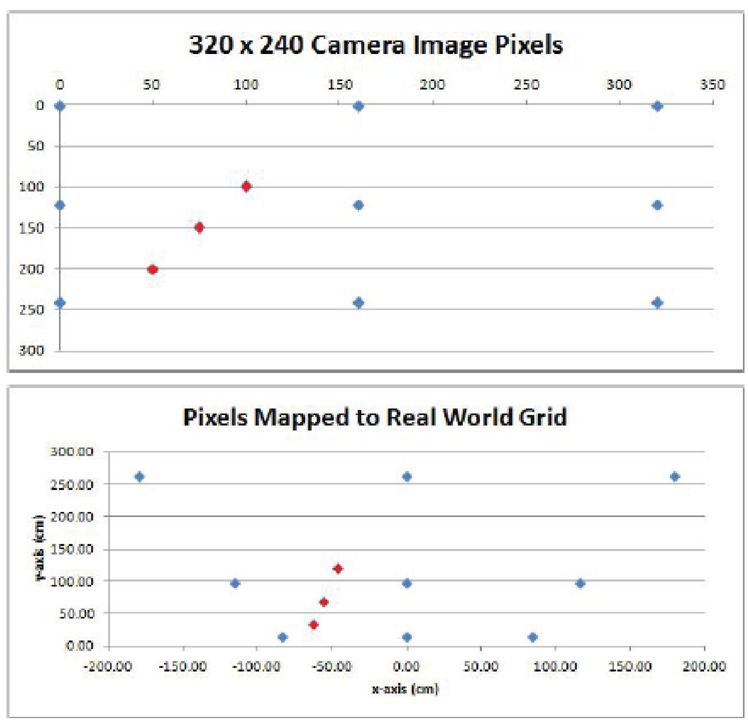

Figure 9: Pixel mapping to plan view

pixel row location and is quadratic in nature. However, the x-axis location depends on both the column and row because of the skewing of the camera image. We used a trigonometric function to account for both variables and map into plan view (see *Figure 9*). The final step is a simple conversion of the (x, y) coordinates into polar form and insertion into the obstacle point array.

6.3.4 Obstacle Preprocessing and Fuzzy Logic

For path planning, we combine the line and solid obstacle arrays into one obstacle array, locate the closest obstacle, and run this obstacle through a Fuzzy Logic algorithm. When combining the two sensor datasets we only keep the closest obstacles. For example, if there is a solid obstacle behind a white line we only keep the line. After the obstacle map is made, the software simply loops over each value to locate the closest obstacle. Finally, we pass this obstacle to the main algorithm.

Lazarus employs a custom designed fuzzy logic control algorithm that operates six linguistic variables. The first four linguistic variables are the system inputs. These given crisp inputs of obstacle distance, obstacle angle, waypoint distance, and waypoint angle are fuzzified through chosen membership functions. The remaining linguistic variables are outputs and are inferred from the rule base and then defuzzified using the Center of Sums method. The crisp outputs are the base speed and base turn ratio. The overview of Lazarus's fuzzy control system is shown in *Figure 10*. The control algorithm updates in real time to adjust to environment changes as they are encountered while directing Lazarus

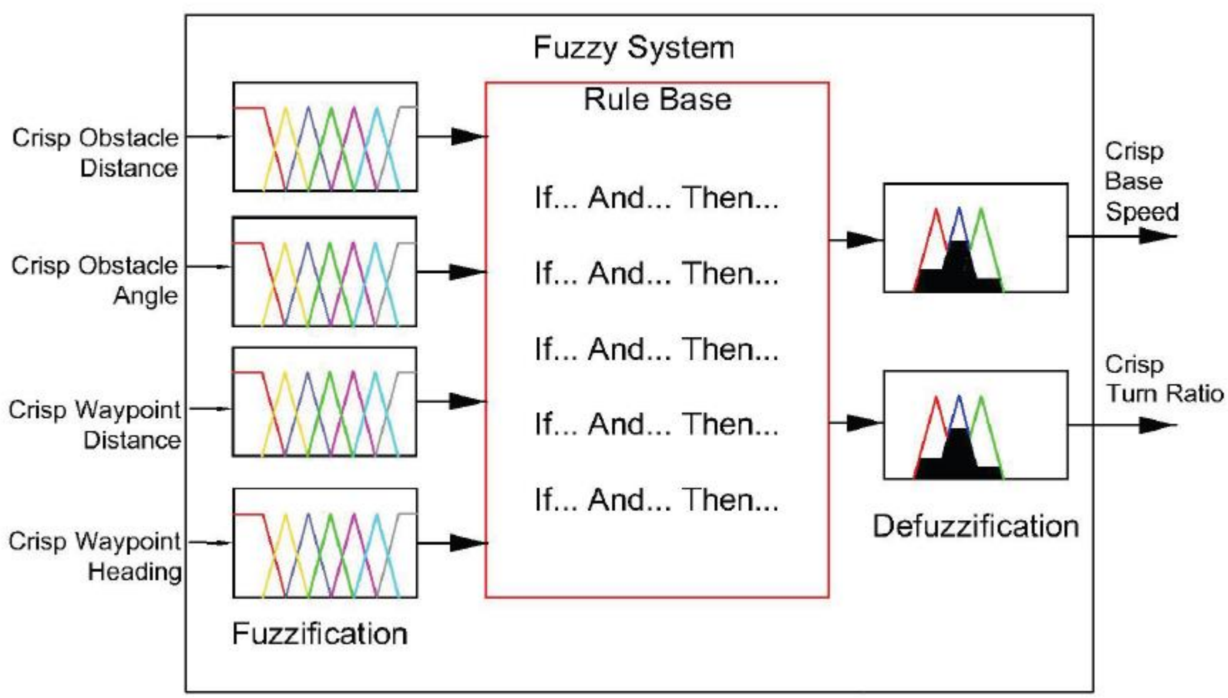

Figure 10: Fuzzy Logic Control Overview

to GPS waypoints. Using fuzzy logic allows Lazarus to be tuned using a larger variety of variables and results in speed adjustments depending on the density of obstacles and lines per unit area. In wide open space Lazarus will drive faster, but in tight spaces he drives slower to ensure enough time to react to obstacles yet undetected. The specific membership functions and rule base used in the fuzzy logic are given in *Figure 11*. Additionally, an example of the rule base used is given in *Table 4*.

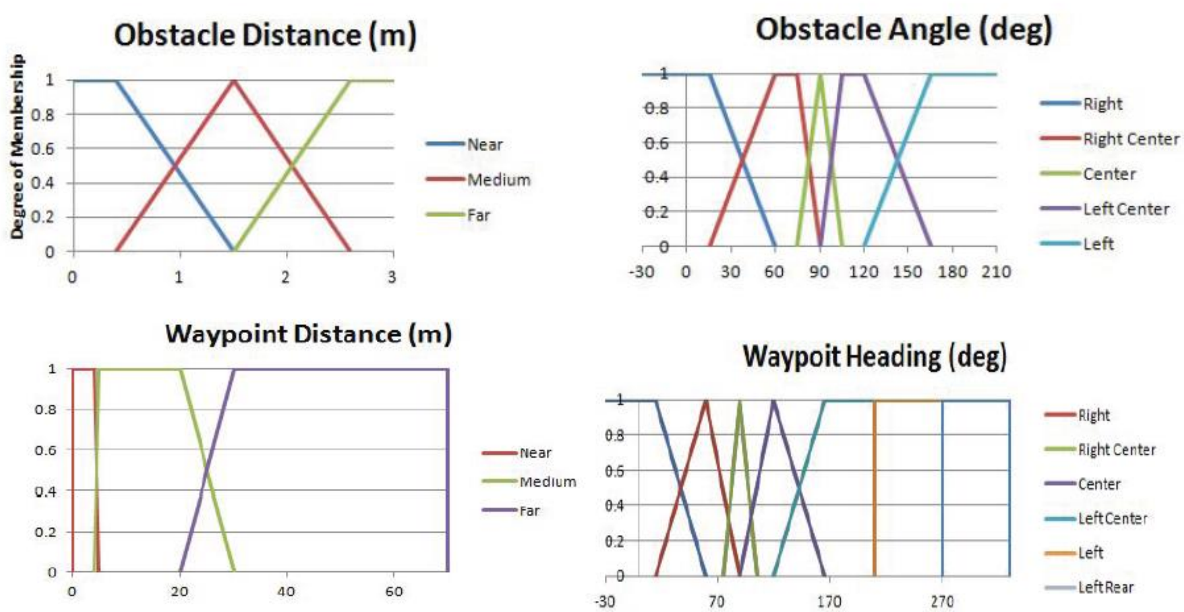

Figure 11: Input Membership Functions

Table 4: Obstacle Avoidance

		Obstacle Angle			
		Right	Right of Center	Left of Center	Left
Obstacle Distance	Near	Slow / No Turn	Slow / Spin Left	Slow / Spin Right	Slow / No Turn
	Medium	Medium / No Turn	Medium / Big Left	Medium / Big Right	Mediu / No Turn
	Far	High / No Turn	High / Small Left	High / Small Right	High / No Turn

7. Failure Analysis and Resolutions

7.1 Possible Vehicle Failure Points

We are currently worried about several possible failure points. First, the computer hardware is currently the major source of failure. We have observed the computer go through random shutdowns both the BSoD variety and suddenly with no indication why. Second, the payload is now located underneath the robot between the two motors. We are worried that removal and attaching of the payload may damage the data connectors to the motor. Third, the LIDAR was dropped during installation. It was tested afterwards and is still functional, but may have been damaged mechanically. Fourth, we have had limited testing of the LIDAR software integration. There are numerous points in which this can fail from slow code execution to unexpected behavior.

7.2 Failure Prevention

Unfortunately, we have had no success in tracing the source of the computer crashes so there is little we can do without acquiring new hardware. We plan to reset the laptop before each run which will hopefully reduce the possibility of a crash.

For protecting the motor control cables, we have added a metal guard for the connectors so if the payload does accidentally knock against them it should prevent damage. In addition, we have established a procedure to always tip the robot onto its nose when attaching or removing the payload. This will give us more control over the payload and reduce the chance for an accident to occur.

We believe a mechanical LIDAR failure to be fairly low risk. Unfortunately, we do not have time to order a spare so if the LIDAR does fail we have no backup.

For preventing LIDAR software integration failure, we will need to perform additional testing to find and fix any problems

7.3 Failure Recovery

Recovering from computer crashes is a simple reset and if the LIDAR integration has issues we can perform code changes and testing on-site without too much trouble.

If the connectors are damaged again we will have to re-solder in the field. We plan on bringing additional tools and materials for any needed repairs.

If the LIDAR fails mechanically there is little we can do. We may be able to repair it onsite, but a successful repair is unlikely. If the LIDAR does fail, we will be out of the competition.

7.4 Safety and Reliability

7.4.1 E-Stop System

Lazarus's emergency stop system can be activated in one of two ways: by pressing the red E-Stop button in the center of the control panel, or by pressing the button on the e-stop remote. The wireless e-stop is a small, black remote with a single red button. The remote has been successfully tested to a range of 50 meters. The E-stop system operates at a frequency of 433.92 MHz with 3 milliwatts of output power.

The emergency stop system takes advantage of the e-stop built into the motor controller. If the e-stop pin is grounded, it will disable the controller. Activating the e-stop through either of the two methods will ground this pin, stopping the robot. The wireless portion of the e-stop system is operated using a HORNET-S1-ND wireless relay from RF Solutions. The HORNET features an antenna that can be separated from the relay unit by a cable. This allows the antenna to be mounted on top of Lazarus's sensor mast while keeping the relay unit hidden in the electronics compartment.

7.5.2 Pedestrian Safety

Lazarus's safety light is a yellow Banner Engineering K50 Beacon EZ-Light. This light was chosen because it can be easily seen from all directions, is bright enough to be visible in daylight, and can be powered from Lazarus's 12-volt power bus. The light is continuously on while the robot is remotely operated, but switches to a blinking pattern when the robot is in autonomous mode. Blinking is achieved using a software-controlled Numato Lab 2-Channel USB Relay Module and a software program running as a separate thread in the main program.

Sharp edges on Lazarus's frame have been rounded off, to minimize injury or damage in the event of a collision. The electronics bay and all areas with power carrying lines are enclosed with ABS plastic and/or plexiglass to reduce the chance of a pedestrian touching a live wire.

8. Conclusion

Many improvements were made to Lazarus, from improved mobility in grassy fields to a more user-friendly electrical system to the implementation of a LIDAR system that is much easier on the budget, and we believe that Lazarus is ready for IGVC 2017.

Table 5: Itemized Cost Estimate

Subsystem	Cost
Frame	$990
Actuators	$669
Electrical	$558
Control System	$1,150
Sensors	$1,163
Suspension	$600
Body	$75
Total:	**$5,205**