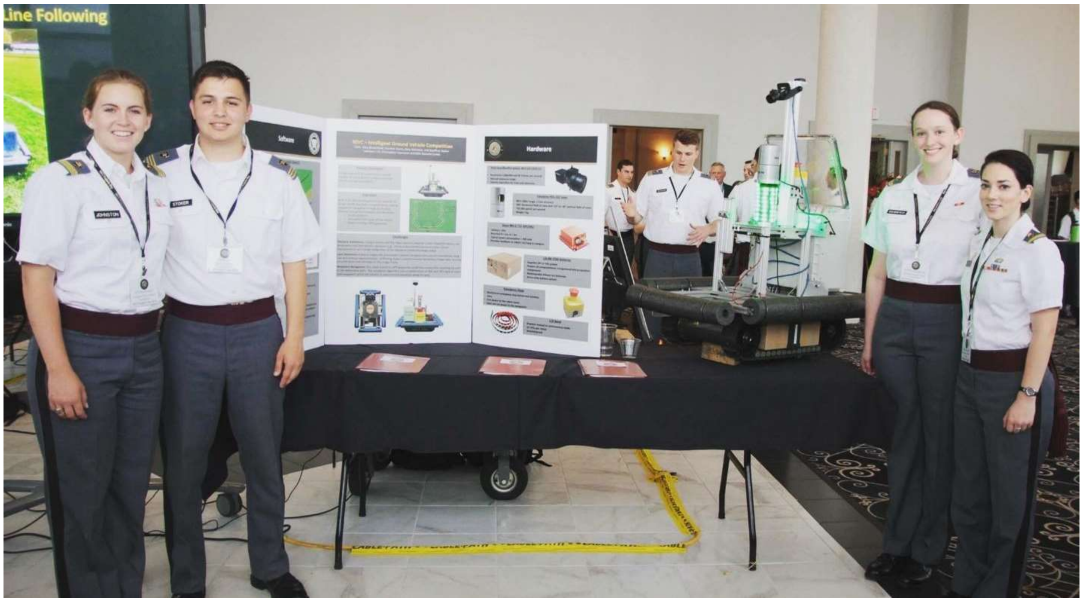
Intelligent Ground Vehicle Competition
2018 Design Report

Electrical Engineering and Computer Science
The United States Military Academy at West Point
606 Thayer Road
West Point, New York 10996
Faculty Advisors: LTC Christopher Lowrance and MAJ Dominic Larkin



Team Members:

(TL) Amy Johnson: Amy.Johnson@usma.edu
Eliza Brownfield: Eliza.Brownfield@usma.edu
Caroline Harris: Caroline.Harris@usma.edu
Geoffrey Stoker: Geoffrey.Stoker@usma.edu
Taylor Sharpsten: Taylor.Sharpsten@usma.edu

Date Submitted: 15 May 2018
Statement of Integrity: Reference "Advisor Statement Memorandum"

DEPARTMENT OF DEFENSE
UNITED STATES OF AMERICA

DEPARTMENT OF THE ARMY
UNITED STATES MILITARY ACADEMY
WEST POINT, NEW YORK 10996

MADN-EC 15 May 2018

MEMORANDUM FOR Intelligent Ground Vehicle Competition (IGVC) Committee, Oakland University, Rochester, MI 48307

SUBJECT: Advisor Statement of Integrity

1. This memorandum serves as a statement of integrity which is required for entry into the IGVC.

2. I certify that the design of IZZY by the United States Military Academy team was completed by the aforementioned cadets as part of their two-semester senior design sequence that totaled seven credit hours.

LOWRANCE.CHRISTOPHER.JOHN.1240102460
Digitally signed by LOWRANCE.CHRISTOPHER.JOHN.1240102460
Date: 2018.05.15 08:37:46 -04'00'

CHRISTOPHER J. LOWRANCE
LTC, FA26
Assistant Professor

1. INTRODUCTION

The Intelligent Ground Vehicle Competition (IGVC) is an annual university challenge where teams compete in designing an autonomous vehicle that can successfully navigate an obstacle-laden course outlined in painted lines. This year's IGVC team representing the United States Military Academy (USMA) at West Point is sponsored by the United States Army Tank Automotive Research Development and Engineering Center (TARDEC), which provides funding for our hardware budget is supervised by the Electrical Engineering and Computer Science (EECS) faculty at USMA. Our robot utilizes the latest technologies in sensory hardware over the modular Robot Operating System (ROS) interface in order to successfully navigate the IGVC obstacle course. In recent years, teams at IGVC have increasingly used Robot Operating System (ROS) as an open source framework to aid in the development of their autonomous vehicles. In August of 2017, our team registered to participate in IGVC as part of our senior design project at USMA. At the USMA Projects Day competition, we received third place overall in the EECS department.

1.1 Problem Statement

Design, build, and test an autonomous robotic platform that meets all requirements and constraints and capable of winning the Intelligent Ground Vehicle Competition (IGVC) on 01JUN18 and compete in USMA's Projects Day competition.

1.2 Team Organization

Our interdisciplinary team consists of five seniors who majored in electrical engineering and computer science. The two faculty advisors from the EECS department are LTC Christopher Lowrance and MAJ Dominic Larkin. The computer scientists include Amy Johnston (team leader) and Geoffrey Stoker. The electrical engineers are Eliza Brownfield, Caroline Harris, and Taylor Sharpsten.

1.3 Design Process

When designing our robot, Izzy, the USMA team used the Agile Development Process from the Scrum Model in Figure 2. We used the basic waterfall design process overseen by the EECS department's faculty advisors. The design process is a five-part process with review or approval at each step: Analysis, Design, Coding, Testing and Operations. This process calls for the incremental building of a product over short periods of time. These periods, known as "sprints," are short-term windows that help subdivide the problem into subtasks in order to facilitate the product's development. Trello and Google Drive significantly helped planning and coordination throughout the team. Trello is a web-based tool to plan using a series of cards as a checklist using the Agile process. We also used GitHub repositories to incorporate branches in our GitFlow in order to work on new feature development. Then we consolidated our work into one fully functional master branch.

Figure 2. Agile Scrum Model

1.4 Design Overview

Our robot, Izzy, meets the mechanical design requirements for the competition height, width, and length. Our team decided to deviate from last year's model and design an entirely new robot. Reducing the bulkiness of the previous model to a simpler, more agile robot, we used a smaller platform and arranged the components in a more compact fashion. We added a clear box on the back of the Izzy to serve as a waterproof housing for most of the robot's electrical components, and added a simple bumper system to protect the robot's framing. The new design is driven by two tracks with rubber tread, and the robot is powered by up to four military-grade lithium-ion batteries (model: BB-2590) located between the treads on both sides of the vehicle. Izzy's platform and battery system require less batteries for powering and a more reliable driving system than last year's robot which required twelve of the same type of battery. When designing our robot, we considered various sensors. The previous robot designed included the best sensors in the industry. This year, we intentionally removed some of the more complex sensors in order to add sensors that were simpler to interface and utilize while still meeting our system's requirements.

2. INNOVATIONS

2.1 Overview

We created a smaller design in order to improve maneuverability and reliability. The robot changed from a completely custom-built robot over to a platform known as the Ground Vehicular Robot (or GVR-bot). The base platform was originally a GVR-bot modified by TARDEC to be an open system architecture and compatible with ROS. We were able to have multiple GVR-bot chasses to help us achieve results quicker by giving us the ability to test and simulate individual sensors and components concurrently without hindering other group members' work.

2.2 Innovative Concepts Learned from Others

Researching previous competing vehicles, we noticed a trend of placing the camera further back on the vehicle in an elevated position. We had initially placed the camera toward the front of the vehicle below the height of the LiDAR. However, we understood the value of increasing the scope of the camera and adopted the popular method by placing the camera behind our LiDAR mounted on a taller 80/20 frame to increase the height. Additionally, we designed our camera mast to be adjustable so that the camera could be set to various heights and angles toward the ground, rather than being fixed in a particular position, as seen in Figure 3. This assist with detecting white lines and minimize light noise from the sun's rays.

2.3 Innovative Technology Integrated into Izzy

Included in our new design built upon a GVR-bot, we added additional sensory components that aid in our vehicle's functionality such as an XSens GPS/IMU and Blackfly Point Grey camera. We also placed our sensors such that if we have the ability to later implement the LiDAR. Our most innovative concept involves the software design in which we used custom ROS nodes that perform image processing using OpenCV. This processing lends aid to our software innovations through enhancing and optimizing the sensory components. Another innovative feature of this year's robot is incorporated into our LED safety design, which visually displays the status of battery life in zones of 20% or less, 40%, and 60% and above, and the lights corresponding to those zones are red, yellow, and green respectively.

2.4 Frame Design

The base GVR-bot does not provide the framing structure necessary for the successful completion of the competition. We designed the payload model necessary to meet specifications by placing 80/20 framing on top of the robot so that the structure could hold the sensors and carry the payload required for IGVC. Additionally, we added a foam bumper system around the chassis in order to ensure the safety for both the robot and other objects or people in the vicinity.

3. MECHANICAL DESIGN

3.1 Overview

The framework of the GVR-bot, shown below in Figure 3, includes additional 80/20 framing to house the payload and additional sensors needed for the competition. It has a durable design and continuous track system that works well on all platforms due to its military-grade tank treads driven by two wheels that are propelled by the motor.

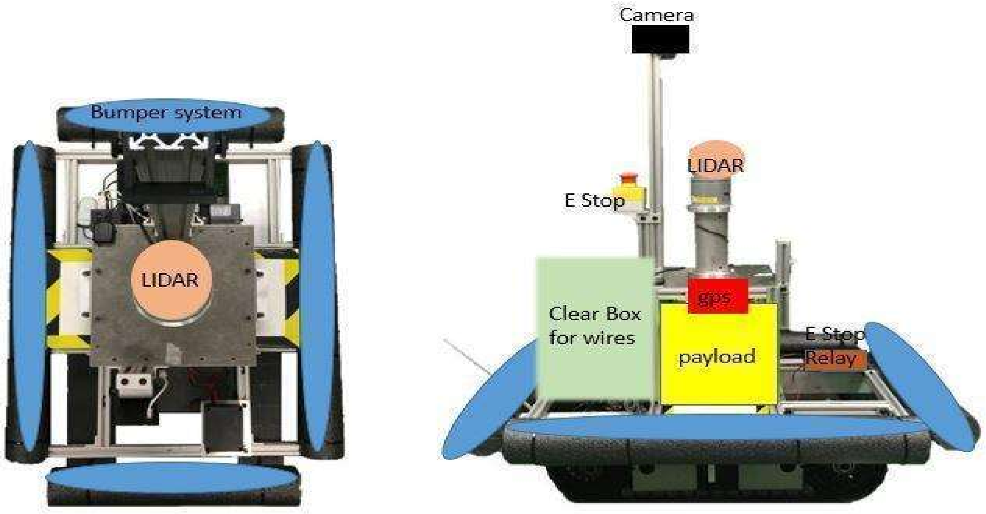


Figure 3. Top-Down View (a) and Side View (b) of Robot.

3.2 Decision on Frame Structure, Housing, and Structure Design

Our decision to make the robot smaller and more compact will guarantee added agility and maneuverability during the competition obstacle course. Housing hardware components in a clear box provides the necessary weather resistance while maintaining the compact design in the rear of the robot. The GVR-bot is a military-grade robot designed to withstand inclement weather and therefore one of the reasons we decided to use this model instead of keeping last year’s design. Because we had multiple GVR-bots on-hand, we were able to use three of them simultaneously for different teams working on the electrical, software, and navigational subsystems.

3.3 Suspension

The tank track is supported by two wheels inside the track system. The suspension system in the GVR-bot includes suspending brackets in the inner side of the track which are adjustable to different grounds. Since the competition is located on grass we typically test and adjust the parameters on a replicated testing ground.

3.4 Weather Resistance

The chassis is water resistant, and the hardware components susceptible to water damage are enclosed in a clear glass housing unit fixed in the rear of the vehicle. This gives the team maximum visibility of hardware around the vehicle for troubleshooting if necessary.

4. ELECTRICAL AND POWER DESIGN

4.1 Overview

The mechanical systems, sensory components, wireless communication system, and safety design features require various power inputs, which are illustrated in Table 1.

4.2 Power Distribution System

Izzy is powered by four BB-2590/U rechargeable lithium-ion batteries running in 24V mode, which has a 27.2 Ah capacity. The GVR-bot's design allows for easy access to the interchangeable batteries in order to aid in their replacement. We maintain sets of charged batteries on standby when the battery life is low. The sensor power ratings of each component are considered in implementing the design.

Component	Power Consumption	Operating Voltage	Sources
Point Grey Blackfly	3 W	8 - 24 V	3.0 USB
Xsens	450 - 600 mW	4.5 - 34 V	2.0 USB
Arduino w/ LED	23.5 mW	7 - 12 V	2.0 USB
Xbox Dongle	2.4 W	5 V	2.0 USB

Table 1: Sensor Power Ratings.

4.3 Electronics Suite Description

The computer is a Brix computer that serves as the control center of the vehicle, as shown in Figure 4. Previous teams determined that neither laptops nor pre-built computers could provide the features or performance required for this project for a reasonable price. At the time of selection, the CPU provided the greatest multi-threaded performance. The computer is connected to a wireless router that enables a remote desktop application to mirror the computer's operation onto an iPad or remote laptop. This allows wireless connection to the computer so that users can make quick adjustments or remove the on-board monitor to reduce weight and power consumption while maintaining a visual display of robot. A more in-depth diagram is provided in Appendix A.

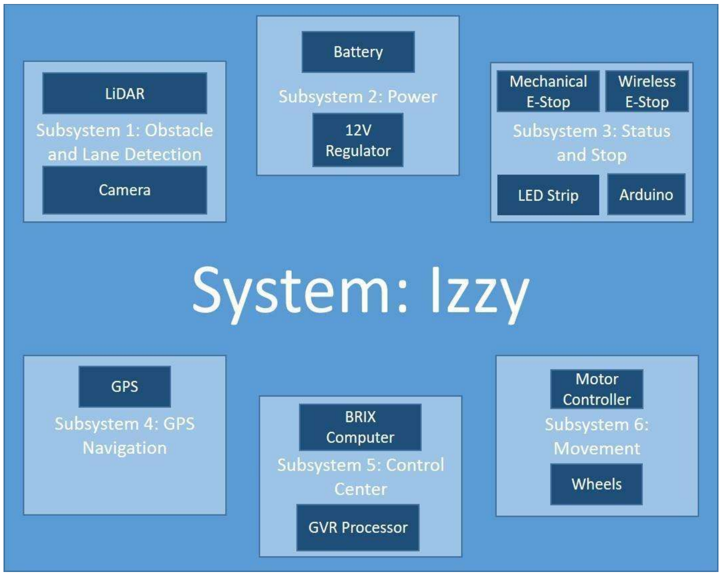


Figure 4. Hardware System-Subsystem Diagram.

4.4 Safety Devices

The safety requirements of the navigation course implemented onto Izzy are the physical and remote emergency stops, speed limit max, and lights along the top of Izzy that indicate whether the robot is in autonomous or manual control mode. Additionally, we mounted LEDs around the top of Izzy that indicates the state of charge on Izzy's batteries, this acts as a secondary safety system when driving Izzy. The battery monitor on Izzy is integrated using an Arduino Uno board with a shield featuring three different colored LEDs. The green, yellow, and red LEDs indicate the state of charge on the batteries inside Izzy. The Arduino board has preloaded code on it that lights up the correct LED with respect to the charge read from the batteries. The battery monitor is a significant improvement to the safety of the vehicle because it allows the user to mitigate the risk of damaging the lithium ion batteries by discharging too much, as well as the risk of the robot suddenly stopping due to no power. The physical emergency stop works by sending an interrupt to the current mode running that stops the vehicle until the button returned to the up position. The wireless emergency stop works in a similar way, by using a wireless relay to short the power circuit. Both of these emergency stop mechanisms are separate from the robot's computers and not dependent on any software, as they are essentially switches in series with the power to the motor that open when triggered.

5. SOFTWARE DESIGN

5.1 Overview

In recent years, teams at IGVC have used ROS with mixed results. The robot design leverages ROS to expedite the development process by allowing team members to work on individual subsystems simultaneously for a modular design. The ROS infrastructure uses nodes that can concurrently run independent processes, and by using ROS, we benefited from abstraction and did not have to write our own multithreaded code. ROS also provides various utilization tools and applications to simulate robot behavior, such as RViz, a three-dimensional visualization tool for ROS, and Gazebo. The ROS repository is an additional core component to ROS that proved most helpful in successfully implements various components that publish or subscribe to various topics to receive messages. This function allows better debugging functionality for quicker implementation process.

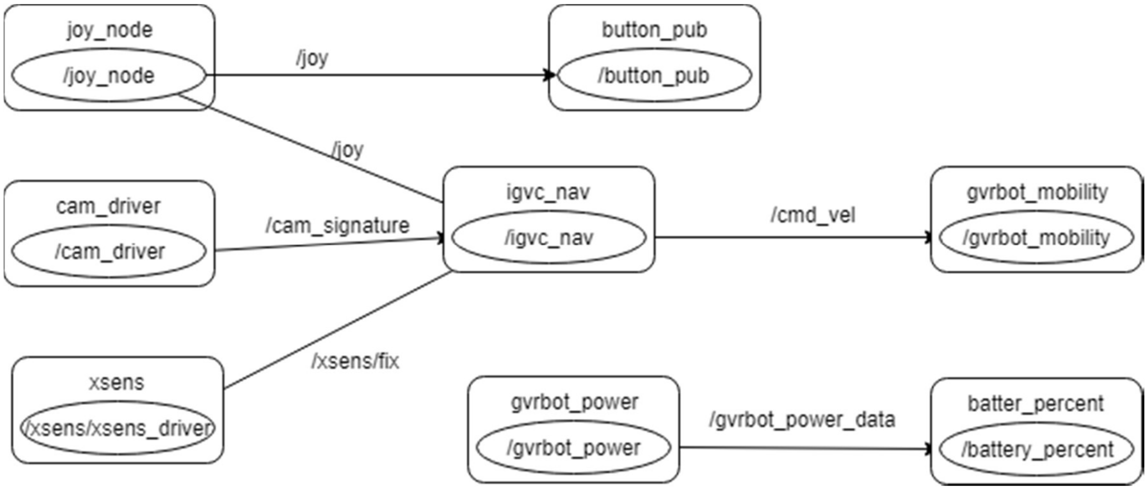


Figure 5. ROS Diagram of Nodes and Packages Implemented in Software Design.

In Figure 5 shown above, the three nodes on the left represent the Xbox 360 controller, camera, and GPS device. These three input devices publish to the main block of code that runs Izzy. Using the received information, the main code decided on a course of action which then publishes that information to the command velocity that is read by the robots motors. This is also reads the *joy_node* and is used to signal the robot. GVR-bot power node reads the power data which is then published and used by the Arduino where the LED lights will initiate.

5.2 Obstacle Detection and Avoidance

Our design encompasses various nodes that continuously talk and listen to each other with given variables. The button publisher publishes whether the state of the robot is in manual or autonomous mode. The *xsens*, which is the GPS/IMU node, the *cam_driver*, which is the camera node, and the *joy_node*, which is the joystick node, all publish to the *igvc_pixy_nav* node, which in turn publishes to the *gvrbot_mobility* node. The previous inputs tell how the robot must act in the /cmd_v and relays the information to the *gvrbot_mobility* whether it is in autonomous or manual mode.

5.3 Lane Following using OpenCV

Our computer vision is accomplished through the Open Source Computer Vision Library (OpenCV). This diagram depicts how the camera and OpenCV work together to collect data and publish it to the ROS framework for the robot to access. After it reads the camera frame, the software will separate the target color from everything else, using a mask. Once the program has the information about the frame it will publish it to one of Izzy's subscribed topics. The robot can now access both the camera image and any data OpenCV derived from it. A more in-depth diagram is provided in Appendix A.

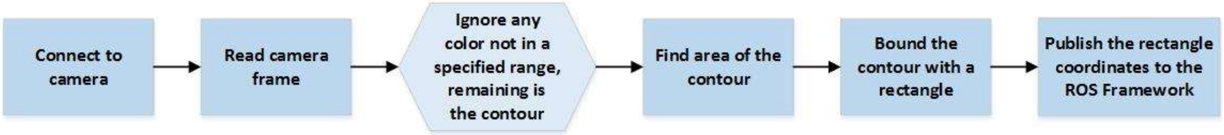


Figure 6. OpenCV Decision Diagram Used for Camera Vision.

5.4 Software Strategy and Path Planning

The software strategy depicted in the Figure 6 above, shows the lane following technique implemented in the robot's software to handrail the lane and avoid various colored obstacles using a camera. Using OpenCV allows the robot to detect a colored line and function autonomously throughout the course, which lead to the numerous successes in the software team's project. We initially started the project using a simple Pixy camera to aid in rapid testing; however, we later transitioned to a more sophisticated camera, the Point Grey camera, and OpenCV processing so that we could optimize our vision-based solution. OpenCV provides the framework necessary to accomplish all of the vision-based requirements. Our code will isolate a single contour and each contour will be bounded by a rectangle, or have a line going through the center of mass. The information is set to the Brix computer to identify each line and/or obstacle described below in Figures 7 and Figure 8.

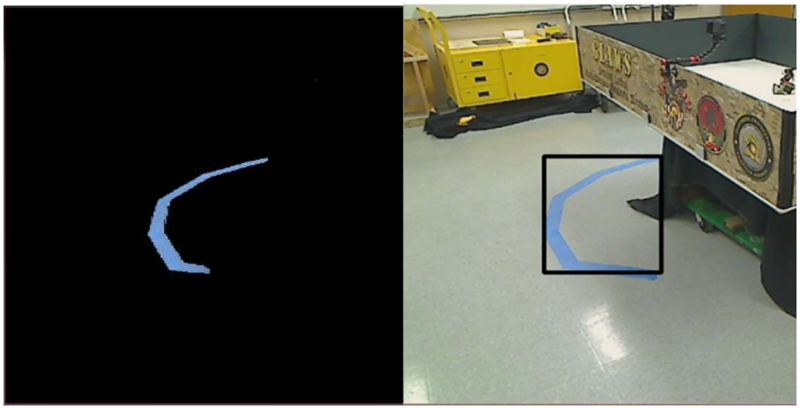


Figure 7. OpenCV Calculates the Size, Shape, and Many Other Aspects of the Identified Contour (i.e. Bounding Box around the Contour in the Image on the Right).

As seen in Figure 8, OpenCV can be used to rotate the rectangle to an optimized position, minimizing the area of the bounding shape while still covering the entire section. Using the rotated rectangle will allow the robot to react more appropriately to the line. OpenCV allows Izzy uses the rectangle edges as target points for lane detection. The next progression after the rotated rectangle was to add a best fit line. This line is calculated by using each shape, or grouping of the target color, to draw a line through the middle. This line will be used to calculate the path of the robot. We used a combination of line and contour approximation. The line approximation will be focused on 'shapes' of at least a minimum size, but still help the contour approximation from missing any gaps in the identification.

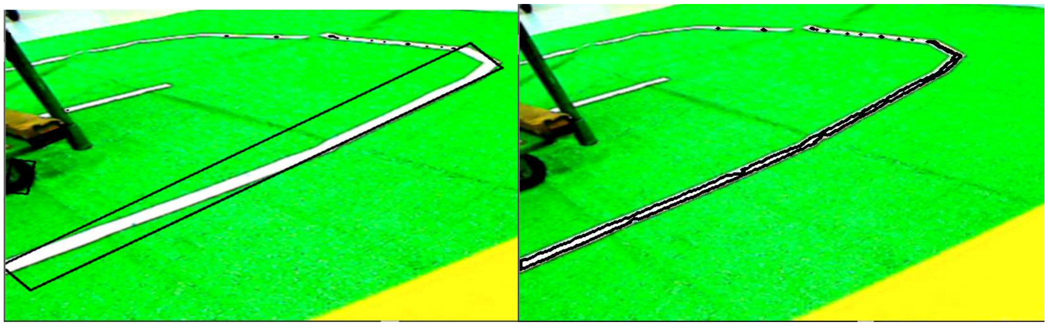


Figure 8. The Right (a) Shows a Current Solution and Left (b) Shows the Optimal Solution

Our current solution takes the largest contiguous grouping of detected pixels and draws a bounding rectangle around those pixels shown in Figure 8a. A future solution we will implement shows bounded groupings of the detected colors. This allows for a more accurate detection of the line, shown in Figure 8b. This better utilizes line approximation and contour approximation to create the best fitted line. OpenCV has proved to be the best solution for lane detection. We plan on integrating the LiDAR for obstacle avoidance where the obstacles are higher off the ground, which is yet another creative concept that will generate success.

5.5 Map Generation and Waypoint Navigation

The combined GPS and IMU sensor made by Xsens, the Xsens Mti-G-710 GPS/IMU, works closely with the path planning and obstacle avoidance. With the GPS coordinates input into the robot, the robot will navigate through the course primarily through the line following code until it navigates within a certain radius of the first waypoint, at which the robot will transition to the waypoint navigation code. The GPS waypoint navigation and map generation uses the Xsens GPS/IMU to coordinate the best path to the waypoint after transitioning from line following and obstacle avoidance. When a waypoint is given, the robot chooses the best path using the software diagram, shown below in Figure 9. The variations in accuracy are due to the satellite and the weather conditions of the day, and we intend to recalibrate the navigation accordingly based on these conditions.

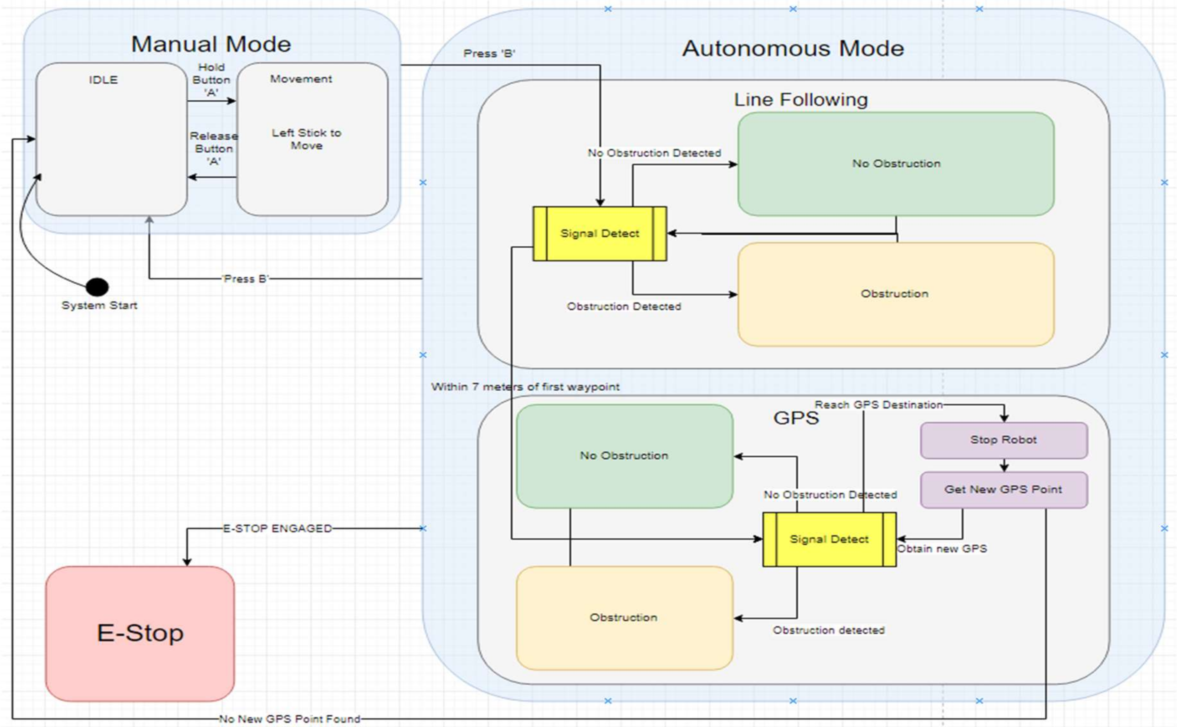


Figure 9. Software Diagram of Software Strategy and Path Planning.

5.6 Goal Selection and Path Generation

The software diagram, shown above in Figure 9, describes the path planning process upon startup configuration. When Izzy first boots-up it is idle in manual mode. Holding the A button will initiate movement by using the left stick to toggle different directions. Releasing button A will stop Izzy. While in line following mode, Izzy will receive location information from the GPS and image data from the camera. While in GPS mode Izzy will inherently go to the next waypoint. When Izzy detects an obstacle, she will avoid it until it is out of view. When Izzy reaches the GPS point, the robot will stop and wait for the next GPS point and enter idle mode. This cycle will repeat until complete or stopped by the emergency stop or until the mission is completed.

5.7 Additional Creative Concepts

Other than integrating OpenCV software, we added a LiDAR to the top of Izzy so that it could better detect and avoid obstacles. OpenCV provides the framework to do all the visual detection by isolating a single color and define it as a ‘contour.’ Each contour can then be bounded by a rectangle and have a line going through the center of mass, this information is sent to the robot’s ROS architecture to identify each line and/or obstacle. The contour analysis and center of mass algorithms working concurrently and publish information to the robot for a more optimal lane following solution.

6. FAILURE POINT IDENTIFICATION AND RESOLUTION METHODS

6.1 Vehicle Failure Modes in Software and Resolutions

We did not have many failures with software and mapping and resolutions. Many of our later work was due to necessary adjustments in the waypoint navigation code due to environmental variations. These environmental adjustments, however, will be more essential in our vision-based code. That is the only factor we are concerned about but can quickly fix it by observing environmental factors that day at the competition site. We do not anticipate this taking long to fix. Lastly, the line following has a relative possibility of failing that day due to environmental factors. If there is a lot of glare on the grass the OpenCV platform will have to adjust to taking out any unnecessary white on the obstacle course. This just means that it ether has to be a cloudy day or configure the algorithm to ignore unnecessary traces of white light. The environmental factors over the course of the competition cannot be ignored as we are well aware of the camera sensitivity. Any software failures and tuning of OpenCV parameters will be handled on site by our software team, and if it does not have an immediate fix we will have to coordinate with another preprogrammed GVR-bot on site, Izzy v2.

6.2 Vehicle Failure Points in Hardware and Resolutions

Our trailer contains additional backup hardware that we made in case of failure and easily interchangeable. These parts include additional electrical hardware including emergency stops, wireless relays, Arduinos, wires, and all the tools necessary to install components. We are also bringing a completed second robot in case of mechanical failure because of the mechanical structure of the GVR-bot. Any hardware issues will be handled by the electrical engineering team. We anticipate hardware failures to occur and we are fully capable of combating such issues with the tool and hardware provided in our trailer/ workstation.

6.3 Failure Prevention Strategy

We do not anticipate an all failure to happen, but we prepared additional electronic components in case a component overheats or breaks during the competition. All hardware component have duplicated, including batteries. Software will be retested on the competition site immediately to ensure we have the correct parameters, especially for navigation. Although if the issues cannot be simply fixed with additional parts. We will also bring a complete replica of Izzy v1 as a last resort.

6.4 Testing All Components in Modular Design

Electronics were replicated and tested on separate GVR-bot to not hinder the software team. The software team tested code and only uploaded working code when tasks were completed for lane following and obstacle avoidance. Waypoint navigation was tested on another GVR-bot and implemented when the software team completed their task. Most components were tested in the lab or replicated competition site depending on weather. Having various GVR-bots to test on, we had the ability to test on various platforms for what each team needed. The Waypoint GVR-bot allowed our electrical engineer to work on configuring the software to find waypoints quickly. The software team worked on obstacle avoidance and lane detection. The Electrical team worked on integrating and testing the hardware components so that they functions properly and consolidated the wiring in the rear of the robot. All groups maintained their separate testing and simulations until fully functional to implement in Izzy.

6.5 Vehicle Safety Design Concepts

The main vehicle safety design includes the bumper system, comprised of black foam pool noodles. They are mounted around the metal pipes on all four sides to protect the robot and objects in the vicinity. The robot's power data, specifically its battery percentage, is published to and utilized by the code programmed on the Arduino that initiates the LED lights on the top of Izzy. This code shows the battery life at 20% or less, 40%, and 60% and above, and the lights corresponding to those values are red, yellow, and green respectively.

7. SIMULATIONS EMPLOYED

7.1 Simulations in Virtual Environment

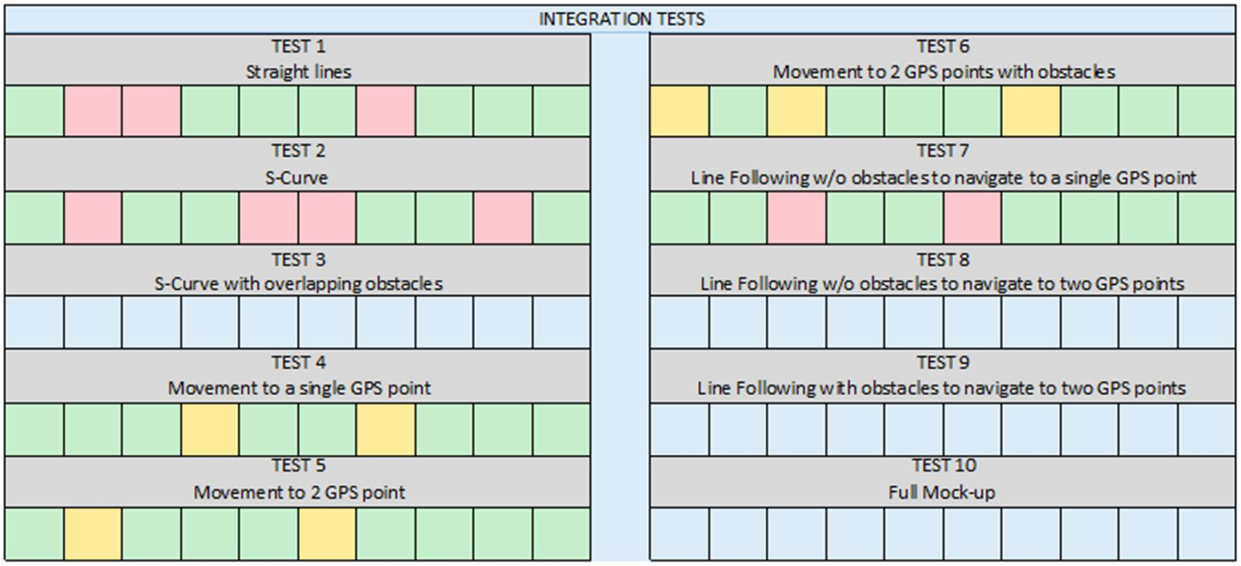
The software team used hardware-in-the-loop (HIL) simulations testing to test complex real-time embedded systems. We continuously tested sensors and software because it was feasible for our team with three robot platforms on-hand. The software team had their own prototype robot to work with to test and simulate OpenCV using the ROS architecture we had previously created to see what the robot could see of the lines and obstacles. More simulations will be done when we begin implementing and enhancing the LiDAR in the near future. The electrical components of Izzy do not need to be simulated to get a viable solution because we worked on a GVR-bot specifically for designing the best way to implement all the electrical components with ease.

7.2 Theoretical Concepts in Simulations

We were leaving for future work building a simulated model of the robot and testing your navigational & obstacle avoidance algorithms using gazebo, but due to time to have a functional system by the end of the year, we elected to perform Hardware-in-the-loop and physical testing. Our focus for this year was to build the basic components and only implement a fundamental avoidance and navigational algorithm. This project is to be built on in future years and with the basic components future teams can focus on more complex algorithms with simulations. Further testing will be conducted to assess the best fit for success. If the LiDAR is not necessary, we may choose to forgo using the LiDAR this year and request that next year's team focus on optimizing LiDAR for obstacle avoidance.

8. PERFORMANCE TESTING TO DATE

Our robot's electrical hardware is fully functional and the LED lights clearly show when the robot is in autonomous or manual mode, as well as the battery life percentage. All electrical components are safely secured in a clear water-resistant box to protect them from inclement elements. The software uploaded in the Brix computer is for lane detection and obstacle avoidance work well together as do the waypoint navigation and can successfully complete a basic course.



INTEGRATION TESTS

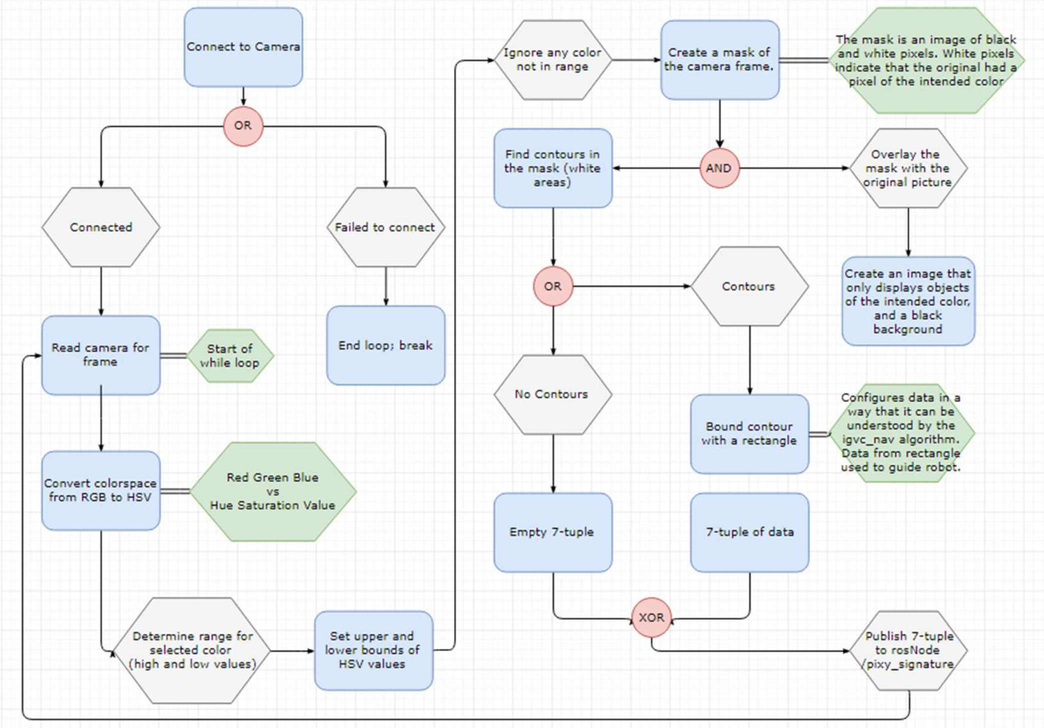
TEST 1 Straight lines	🟩	🟥	🟥	🟩	🟩	🟩	🟥	🟩	🟩	🟩
TEST 2 S-Curve	🟩	🟥	🟩	🟩	🟥	🟥	🟩	🟩	🟥	🟩
TEST 3 S-Curve with overlapping obstacles	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
TEST 4 Movement to a single GPS point	🟩	🟩	🟩	🟨	🟩	🟩	🟨	🟩	🟩	🟩
TEST 5 Movement to 2 GPS point	🟩	🟨	🟩	🟩	🟩	🟨	🟩	🟩	🟩	🟩
TEST 6 Movement to 2 GPS points with obstacles	🟨	🟩	🟨	🟩	🟩	🟩	🟨	🟩	🟩	🟩
TEST 7 Line Following w/o obstacles to navigate to a single GPS point	🟩	🟩	🟥	🟩	🟩	🟥	🟩	🟩	🟩	🟩
TEST 8 Line Following w/o obstacles to navigate to two GPS points	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
TEST 9 Line Following with obstacles to navigate to two GPS points	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
TEST 10 Full Mock-up	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦

Table 2. Integration Testing

Table 2 depicts the various tests performed with our vehicle, with green signifying success, amber signifying partial success, red signifying failure, and blue signifying a pending test. Progressing through the tests, Test 1 entailed the robot navigating successfully through a path delineated by straight lines, Test 2 entailed navigating through an S-shaped lane, and Test 3 entailed added obstacles to the S-shaped lane. Concurrent with conducting these tests, we used another GVR-bot to test the waypoint navigation: in Test 4 we evaluated the robot's ability to navigate to a single waypoint, in Test 5 we evaluated its ability to navigate to 2 waypoints, and in Test 6 we evaluated its ability navigate to 2 waypoints while avoiding obstacles. Our integration testing thus far was conducted through Test 7, in which we evaluated our robot's ability to transition from line following without obstacles to navigate to a single waypoint. We intend to continue testing through conducting Tests 8, 9, and 10 in which we will assess integration further through adding waypoints and obstacles, culminating with trials mirroring competition conditions. We are very proud of all the work have done thus far and know that it will be a foundation for next year's team to build upon.

APPENDIX A

Software Diagram



Hardware Diagram