Intelligent Ground Vehicle Competition

Technical Design Report

Department of 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 LTC Kathryn Pegues

Team Members:

(Team Captain) Tianna Johnson: Tianna.Johnson@westpoint.edu

Will Born: William.Born@westpoint.edu

Brian Degori: Brian.Degori@westpoint.edu

Jared Ells: Jared.Ells@westpoint.edu

Taylor Sharpsten: Taylor.Sharpsten@westpoint.edu

Date Submitted: 15 May 2019

Statement of Integrity: Reference “Advisor Statement Memorandum”

Table of Contents

1. INTRODUCTION

The Intelligent Ground Vehicle Competition (IGVC) is an annual robotics competition sponsored by the United States (US) Combat Capabilities Development Command – Ground Vehicle Systems Center (GVSC). The competition allows students to leverage current technologies to create solutions that are applicable in the field of autonomous navigation. A sprint-based agile design approach and integration of cutting-edge technology were integral to the United States Military Academy's (USMA) team's ability to design and develop an autonomous ground vehicle. The USMA-designed robot, named Intelligent Vehicle of Autonomous Navigation (IVAN), utilizes multiple sensors and computing hardware to achieve autonomous navigation. Sensor outputs from a camera and laser rangefinder are integrated and processed by a neural network for determining the best navigational path. IVAN is a powerful, reproducible, and applicable solution to the Army's desire to modernize using the next generation of autonomous robots capable of self-navigation in challenging environments.

1.1 Problem Statement

Build a fully autonomous unmanned ground robotic vehicle that can navigate an outdoor obstacle course under a prescribed time while remaining within its designated lane and detecting obstacles with adequate reaction time to avoid them on the course for the IGVC on 07JUN2019.

1.2 Team Organization

The interdisciplinary team consists of five undergraduate students from the United States Military Academy. The faculty advisors include LTC Christopher Lowrance and Mr. Dominic Larkin from the Department of Electrical Engineering and Computer Science (EECS) and LTC Kathryn Pegues from the Department of Systems Engineering (DSE). The team leader, Tianna Johnson, is a computer science major, Will Born and Taylor Sharpsten are electrical engineering majors, Brian De Gori is a systems engineering major, and Jared Ells is an engineering management major.

1.3 Design Process

While designing the robot, IVAN, the USMA team used the Agile Development Process from the Scrum Model. 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 for the Agile process. GitHub repositories were used to incorporate branches in our GitFlow in order to build on new feature development.

1.4 Design Overview

The USMA team decided to use the base design from the 2018 USMA robot and make small physical adjustments while completely revamping the components used on the robot. Reducing the bulk of previous designs by removing the keyboards, monitor screens, and lowering the center of gravity, the design was simplified for easier use and less physical volume. The component box was moved from a vertical position to lying horizontally on the platform of the robot. The use of the aforementioned weather-resistant box, the bumper system around the edges of the robot, and the tracked drivetrain platform were maintained. These solutions were adequate for this year's submission and effort to redesign these areas would not have made as significant a contribution to improving the capabilities of the robot. IVAN uses two tracks with rubber tread, and is powered by up to four military-grade lithium-ion batteries (model: BB-2590) located between the treads on both sides of the vehicle. IVAN's platform and battery system was improved to include an additional Lithium Polymer battery for powering the computing hardware independent of the robot's motor. New sensors and computers replaced legacy hardware in the 2018 robot's design. The additional components include two Odroid single-board computers, a CNS-5000 GPS unit, two LattePanda single-board computers, a MSI GeForce GTX 1050 Ti, and a small SICK LIDAR mounted on the front of the robot's chassis.

Item	Quantity	Unit Cost ($)	Cost ($)	Notes
LattePanda Alpha	2	358.00	716.00	2 per robot
Amcrest 8 Port PoE Switch	1	59.98	59.98	1 per robot, 1 extra (or to test sensors on lab bench), 3 total purchased
Amcrest ProHD 1080P PoE Camera	1	84.89	84.89	1 per robot, 1 extra, 3 total purchased
MSI 1050 ti	1	182.99	182.99	1 per robot, 2 total purchased
M.2 to PCIE 4x (2 Pack)	1	9.99	9.99	2 pack, one per gpu
CNS-5000 TerraStar Correction Service	1	1200.00	1200.00	1 Year Subscription
LiPo Batteries	1	150.00	150.00	1 per robot, 5 total purchased
Total Cost of Upgrades			2403.85	

Table 1. 2019 USMA Component Cost Summary Per Robot

2. INNOVATIONS

2.1 Overview

The innovation this year was focused on innovating and advancing the abilities of the prototype instead of restarting the design process (Johnson et al., 2018). The team decided to use the design from last year's model as a starting point and make small physical adjustments while completely revamping the navigational, computing, and obstacle avoidance components. The four major areas of innovation for the 2019 design included the removal of unnecessary components, a track modification to increase traction, and the inclusion of machine learning to improve navigation. The bulkiness of the previous design was reduced by removing the keyboard and monitor screens. The component box was moved from a vertical position to lying horizontally on the platform of the robot. The adjustment lowered the robot's center of gravity. The final 2019 USMA design, in comparison to the 2018 USMA design, was sleek, physically smaller, and provided improved stability. During the 2018 competition, the USMA team noted that the robot experience slippage while on the course. To counter this issue plastic pegs were produced and inserted between the wheels and the track on the robot. In addition to the physical modifications the team made software updates to improve accuracy and speed of navigation. The team was able to better utilize the lidar sensor on the robot by implementing the ARL Open-Path algorithm. Finally, machine learning was implemented this year in the form of a convolutional neural network (CNN). The neural network is used to implement imitation learning to aid with navigation.

2.2 Innovative Concepts Learned from Others

Upon researching previous competition entries during the 2017 and 2018, a design trend of placing the camera further back on the vehicle in an elevated position was noted. This adjustment increases the field of view and also produces a better view of the lane lines. Using this innovation, the team moved the camera from the main vertical mast to the front of the platform. The adjustment reduced the height of the camera's perspective which enabled the camera to see the ground in front of the robot. The change eliminated a “blind spot.” The location and orientation of the component box and location of the payload location was adjusted in order to adjust the location of the center of mass toward the front of the robot and lower to the ground. These innovations improved the stability of the robot and its ability to sense the lane lines and obstacles by providing a less obstructed view to the sensors.

2.3 Innovative Technology Integrated into IVAN

The main improvement to the software solution was the introduction of machine learning. The Keras python module was used with a TensorFlow backend to train a CNN to classify input images as left, right, or straight. The input to the network is a combination image of the camera and lidar sensors. The data from both sensors is combined into one side-by-side image so that a single network can process both at once. The network is trained on human input and the resulting network can navigate the robot by learning to imitate the human input.

2.4 Frame Design

A custom designed frame was constructed to house the necessary components. The designed frame meets the payload specification by using 80/20 framing on top of the robot's base so that the structure could hold and position the sensors while being able to carry the payload required for IGVC. Additionally, a foam bumper system encompasses the chassis in order to ensure the safety of both the robot and other objects or people in the vicinity.

3. MECHANICAL DESIGN

3.1 Overview

The physical framework of IVAN, as shown in Figure 1, includes 80/20 aluminum framing to house the payload, computing components, and mounted sensors needed for obstacle detection and avoidance. The robot contains an external LiPo battery, separate from the GVR-bot's batteries, to power the components. This allows the computing components and sensors to be run separate from the system powering the tracks' motors. The 80/20 utilized in the IVAN prototype provides the encasing system to house and position the components providing weather resistance protection and support to the computers and sensors.

3.2 Decision on Frame Structure, Housing, and Structure Design

The current frame design allows the system to be easily attached to and removed from the existing GVR-bot platform. This makes it very easy to replace the robot platform when problems occur and also makes the system easy to reproduce and distribute. The frame structure used is very strong and the improvements this year to the weight distribution and center-of-mass make the robot more stable and maneuverable. The housing for the compute module is weather-resistant and also easily detachable and replaceable. Overall, the hardware redesign this year has increased the capabilities of the system, both for navigation and power consumption.

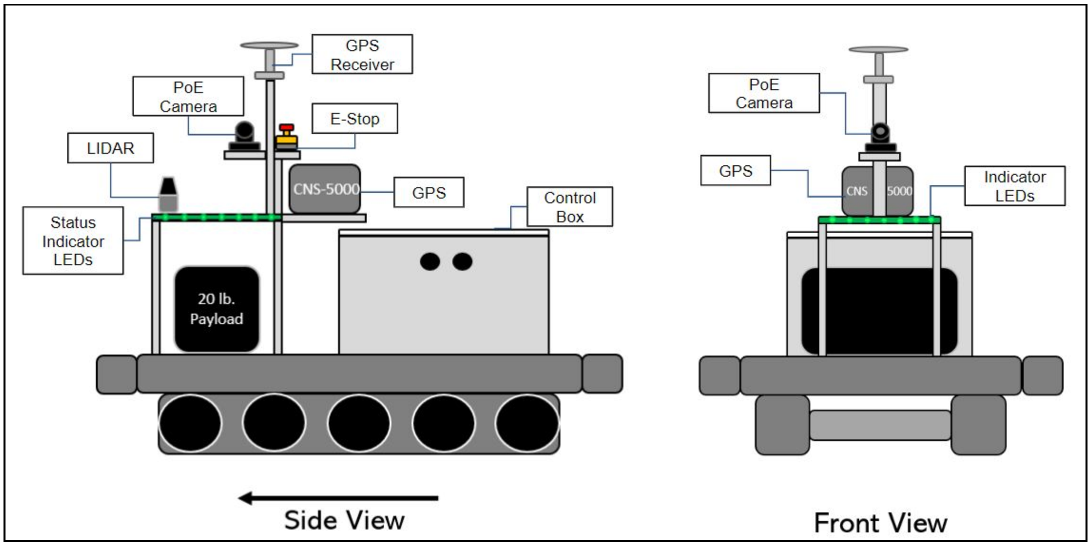


Figure 1. Hardware Diagram

3.3 Suspension

The tracked system is supported by two wheels inside each of the tracks. The suspension system in the GVR-bot includes suspending brackets in the inner side of the track which are adjustable to different types of terrain. For example, the tracks for use on grass require deeper treads than on sand or dirt. Since the competition is located on grass testing was done on grass to replicate the competition's conditions. 3D printed pegs were added to the tracks' wheels to improve traction on wet or long grass, to mitigate slippage during suboptimal conditions.

3.4 Weather Resistance

IVAN is designed to operate in adverse weather conditions. The chassis is water resistant. The hardware components identified as susceptible to water damage are enclosed in a clear housing unit fixed in the rear of the vehicle. The use of a clear box additionally provides the team visibility of hardware.

4. ELECTRICAL AND POWER DESIGN

4.1 Overview

IVAN is powered by four BB-2590/U rechargeable lithium-ion batteries running in 24V mode. These batteries will power the motor of the GVR-bot base. A 20,000 mAh Lithium Polymer (LiPo) battery is used to power the hardware components such as the two latte pandas for computing the obstacle avoidance code.

4.2 Power Distribution System

IVAN is powered by four BB-2590/U rechargeable lithium-ion batteries with a 27.2 Amp-hour capacity, and one lithium polymer (LiPo) battery providing 14.8 V at a 20,000 mAh capacity. The team decided to add the LiPo battery in addition to lithium-ion batteries to extend that operational life of the prototype by allotting power capacity to the motor units and the computing components separately. The BB-2590/U batteries power the robot chassis' motors. The chassis physical design allows for easy access to the interchangeable batteries to simplify their replacement. The LiPo battery provides power to the various electronics within the component box, distributed on a printed circuit board. The 14.8 V input power is routed through multiple 12V and 5V regulators to power components directly mounted on the board, such as both Odroid computers and an E-stop receiver, as well as components separately mounted within the box, such as the single-board Odroid's and switches. A custom-designed, printed circuit board was created to distribute power and minimize wiring within the box. This was done to help shift the effort this year from troubleshooting on software rather than hardware. The GVR-bot's design allows for easy access to the interchangeable batteries in order to simplify their replacement. Sets of charged batteries are maintained on standby for when the battery life runs low. Additionally, an LED strip is wrapped around the robot to show the status of the robot's condition, either autonomous or manually operated. The lights remain solid until the robot enters autonomous mode, at which point they switch to flashing. The sensor power ratings of each component were considered in implementing the design in order to accomodate necessary power requirements for IVAN's proper function.

4.3 Electronics Suite Description

The robot's compute module is powered through a high-capacity LiPo battery and a custom-designed printed circuit board (PCB). The PCB distributes the battery voltage to six voltage regulators. These regulators create four 12 volt rails and two 5 volt rails. Each computer and graphical processing unit (GPU) gets its own regulated 12 or 5 volt source and the rest of the components share the remaining 12 volt supply. All data connections are made over ethernet or universal serial bus (USB).

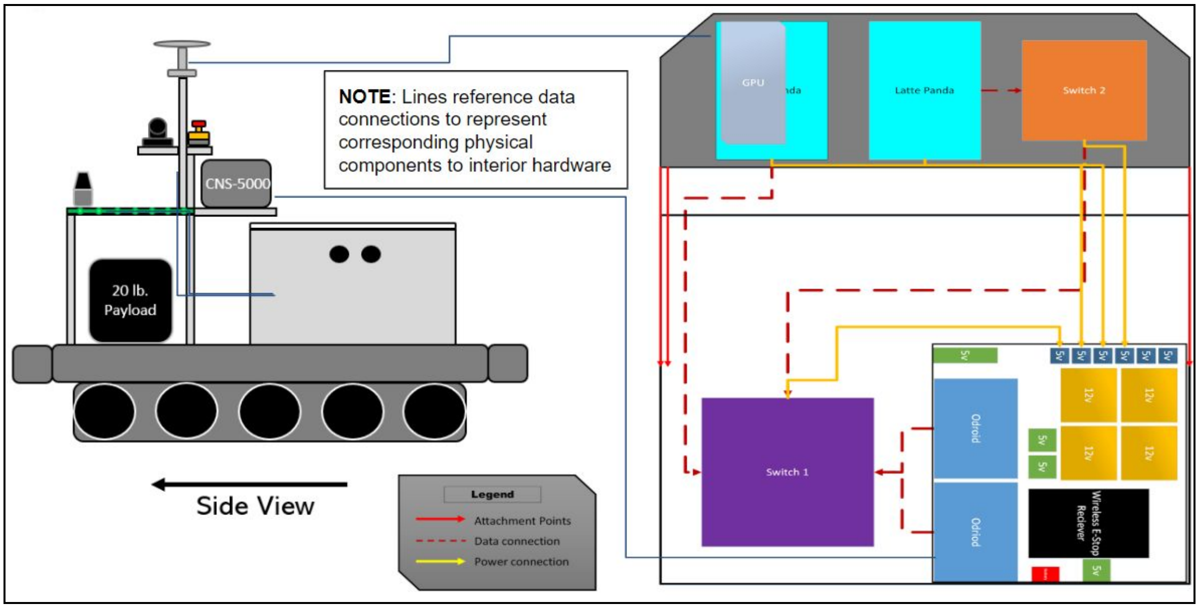


Figure 2. Component Box Electronics Diagram

4.4 Safety Devices

The safety requirements of the IGVC course implemented onto IVAN are the physical and remote emergency stops, specified speed limit, and lights along the top of IVAN that indicate whether the robot is in autonomous or manual control mode. IVAN has physical and remote emergency stops to safely control the robot's actions. The physical emergency stop button is located at the middle of the robot where it can easily be reached by the operator. This button acts to cut the flow of power to the robot, causing it to stop in emergencies. The wireless emergency stop uses a wireless relay to short the power circuit in situations where the robot needs to be stopped from a distance. Both of these emergency stop mechanisms are separate from the robot's computers and not dependent on any software, as they are set in series with the motor's power source that is disabled when triggered. There are lights along the top of IVAN that indicate whether the robot is in autonomous or manual control mode. Additionally, LCD screens are located on each of the GVR-bot's batteries that indicate the state of charge on IVAN's batteries, this acts as a secondary safety system.

5. SOFTWARE DESIGN

5.1 Overview

IVAN's software design leverages ROS to expedite software development by allowing team members to work on individual subsystems simultaneously in a variety of programming languages while seamlessly targeting a modular design. The ROS infrastructure uses nodes that can concurrently run independent processes. By using

ROS, the team benefited from abstraction and did not have to write new, multi-threaded code. ROS also provides various tools and applications to simulate robot behavior. The tools include RViz, a three-dimensional visualization tool, and Gazebo, a simulator. The ROS repository is an additional core component to ROS that proved helpful in successfully integrating various components that publish or subscribe to various topics to receive messages. This function allows better debugging functionality for quicker implementation process. The use of ROS greatly aided in expediting the software development and troubleshooting processes experienced in the creation of IVAN.

5.2 Obstacle Detection and Avoidance

For obstacle detection, IVAN utilizes the Sick TiM 571 LIDAR. The Sick LIDAR has a 270 degree horizontal field view and an ten meter depth scanning range. As IVAN moves throughout the obstacle course, the LIDAR constantly scans for obstacles. The data that is received by the LIDAR is taken in as an input to IVAN's fuzzy controller. Fuzzy control involves taking a value from a sensor node and weighting it through fuzzy logic. Each sensor is given a fuzzy set that corresponds to the output states of the robot. This ultimately allows to signals to be averaged and effectively combined into one signal. Camera and GPS data are also being ingested at this point, and based on the weight values associated with the inputs, the fuzzy controller sends a direction signal to IVAN's drive controller based on which sensors' data gets priority on determining the best path for the robot to travel.

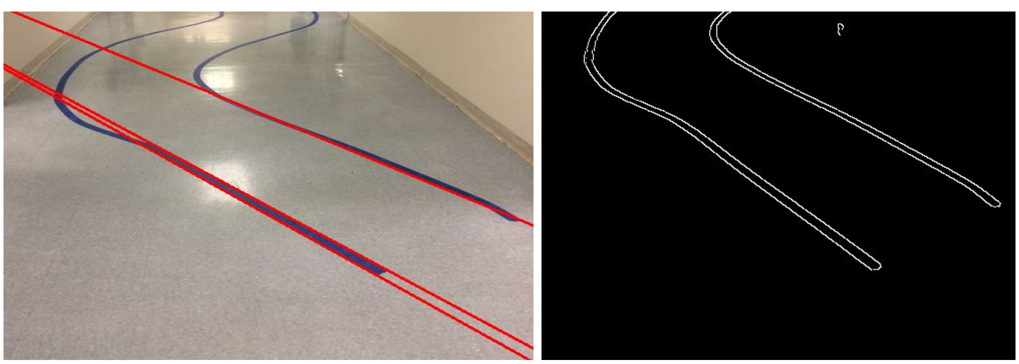


Figure 3. Hough Transformation for Line Detection

5.3 Lane Detection

The algorithm that was used to identify the lines on the course is the Hough Transform. Every line in polar space is defined by unique coordinates 'r' (radius) and 'theta' (its angle). The Hough Transform detects the unique lines of an image by counting the number of pixels that create the same unique coordinate. For each unique coordinate, if the sum of pixels is greater than a chosen threshold, then it will be considered a line. Lines are then broadcasted in an image as white pixels, whereas non-lines pixels are changed to black. This contrasted image was utilized by the team as an input to the fuzzy controller decision process.

5.4 Software Strategy for Path Planning

The main software strategy for the robot is to break up the sensors into different nodes. Each node corresponds to one of three sensors: camera, Lidar, and GPS. Each node takes the raw sensor data and translates it into a command velocity signal which can drive the robot. Each node can independently control the robot through a course with only the obstacle that it is programmed to avoid. For example, running the robot on the LIDAR node will avoid obstacles but not stay in the lane. The signals from each node are ultimately averaged using weights determined by how imminently the robot will reach an obstacle that the node is concerned with. This is done by combining the signals using fuzzy control to determine the percentage that each node should be weighted. The robot does not conduct path planning, but rather produces a control signal based on current sensor data with a loop rate fast enough to navigate effectively.

5.5 Map Generation and Waypoint Navigation

The IGVC course requires the robot to navigate within a one meter radius of four GPS waypoints located on the course. The navigation algorithm receives current GPS latitude and longitude values from the CNS-5000 and the latitude and longitude of the next desired waypoint as inputs. It outputs motor drive commands to the integration controller. Assuming forward motion of the robot, differential GPS is used to calculate the current orientation of the robot as well as the distance to the target GPS waypoint. If the robot is not within 50 centimeters of the target waypoint, the algorithm calculates whether the robot needs to turn clockwise or counter-clockwise. The robot will move forward while turning in the respective direction until it is oriented within a two degree radius of the desired orientation. At this point, the robot will switch to a straight orientation. The algorithm will continuously update the current orientation as IVAN moves, adjusting the drive commands and moving towards the target waypoint until the robot is within 50 centimeters of the target. Upon arrival, the algorithm will read and initiate movement towards the next desired waypoint.

5.6 Goal Selection and Path Generation

During the lane-navigation portion of the competition, the robot navigates by taking the current sensor data and determining the best course of action based solely on present state. The robot’s goal is to move as far as possible without hitting an obstacle or going over a line. Once the robot enters the gps portion of the course, a predetermined list of points is used and the robot navigates to each. During this time, the lidar algorithm still runs to prevent the robot from hitting an obstacle on its way to a gps point. Upon reentry to the lane-navigation portion of the course, the original behavior resumes.

6. FAILURE POINT IDENTIFICATION AND RESOLUTION METHODS

6.1 Vehicle Failure Modes in Software and Resolutions

IVAN’s software is designed to navigate using by continuously moving forward and producing left and right signals to avoid obstacles. However, this may lead to a point at which the robot finds itself in a dead end. This can be detected by the largely different signals being produced by each node. In this case, the robot will conduct an escape algorithm which consists of backing up and then turning itself 45 degrees and reentering the main loop. Due to the nature of the competition, further software failure will likely result in run termination. The team has taken great effort to ensure that the individual nodes are robust, but unforeseen conditions on the test course could require additional patching of the underlying code.

6.2 Vehicle Failure Points in Hardware and Resolutions

In anticipation of hardware failures, we purchased duplicates of our main hardware components. These components include backup Latte Pandas, switches, cameras, and batteries. Our trailer is also equipped with additional wires, tools, and soldering equipment to attend to

6.3 Failure Prevention Strategy

An all encompassing total system failure is not expected to occur, but the team has prepared additional electronic components in case a component overheats or breaks during the competition. All hardware components have duplicates, including batteries. Software will be tested on the competition site immediately to ensure the correct parameters and programming are in place, especially for navigation. Although, in the case that the issues cannot simply be fixed with replacement parts, the complete replica of IVAN will be on standby as a last resort to implement as the team’s submission.

6.4 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 framing on all four sides to protect the robot, objects in its vicinity, and bystanders from injury. The robot’s power data, specifically its battery percentage, is published to and utilized by the code programmed on the Arduino that initiates the LCD screen on the GVR-bot batteries. This shows the battery life at 20% or less, 40%, and 60% and above. These are all used to protect the robot from damage or losing power in dangerous situations (when it may need to move out of the way of another stray robot or an oncoming vehicle).

7. PERFORMANCE TESTING TO DATE

7.1 Component Testing to Date

To date, the Lidar and GPS subsystems have been tested and work as intended. All sensors have been tested and verified to work either within specification or adequate for their application on the robot. The Lidar algorithm has been tested in multiple obstacle configurations and has proven to avoid obstacles. The GPS algorithm has been effectively tested using sample GPS data, but has not been tested yet in an actual navigation scenario.

7.2 Integration Testing to Date

The integration algorithm has not yet been tested on the actual robot as not all subsystems are complete. However, initial testing using multiple command velocity signals has proven the concept of combining multiple weighted inputs.

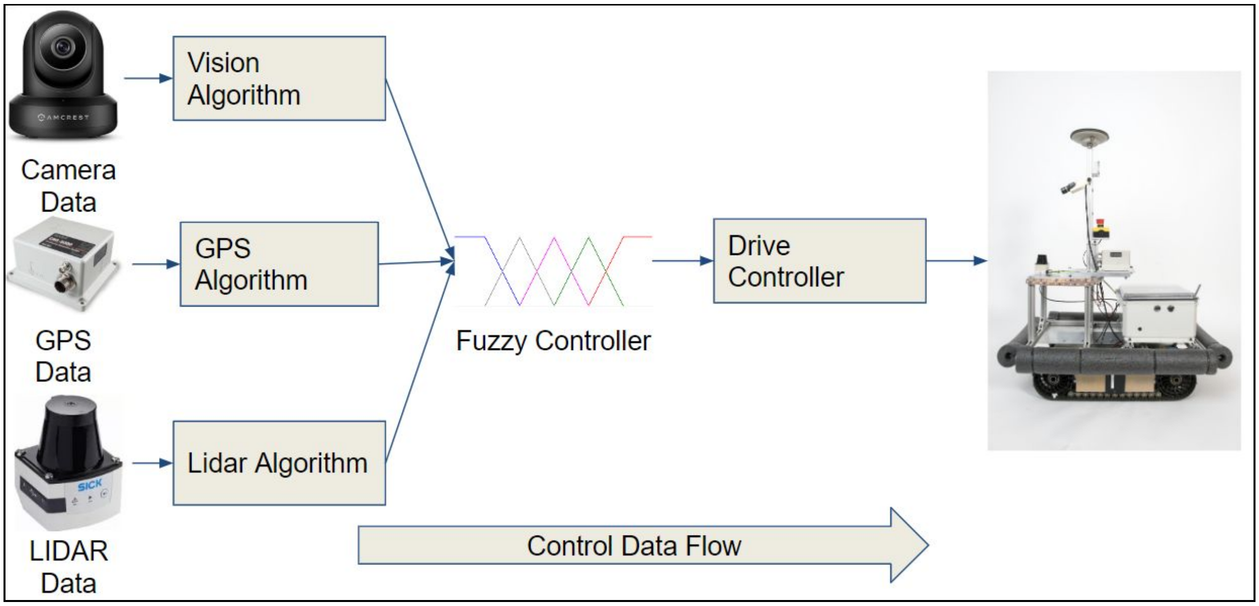
8. INITIAL PERFORMANCE ASSESSMENTS

8.1 Current Vehicle Performance

IVAN's individual nodes perform well, however the work on the integration of these algorithms has not been completed to date. Initial performance assessments on each node have been promising. The lidar node has been especially successful, and the robot can navigate through tight spaces without touching an obstacle. The computer platform has performed well. The PCB has greatly simplified the number of wires in the component box, which has reduced the number of problems while testing the robot. Separating the computer battery power from the main robot platform has increased battery life significantly. Also, the new compute module is several times faster than the old solution. This has enabled much greater flexibility for the software design, and also enabled the use of machine learning. With further development, the robot should surpass the capabilities of the previous year's submission.

APPENDIX A

Software Diagram



Network Diagram