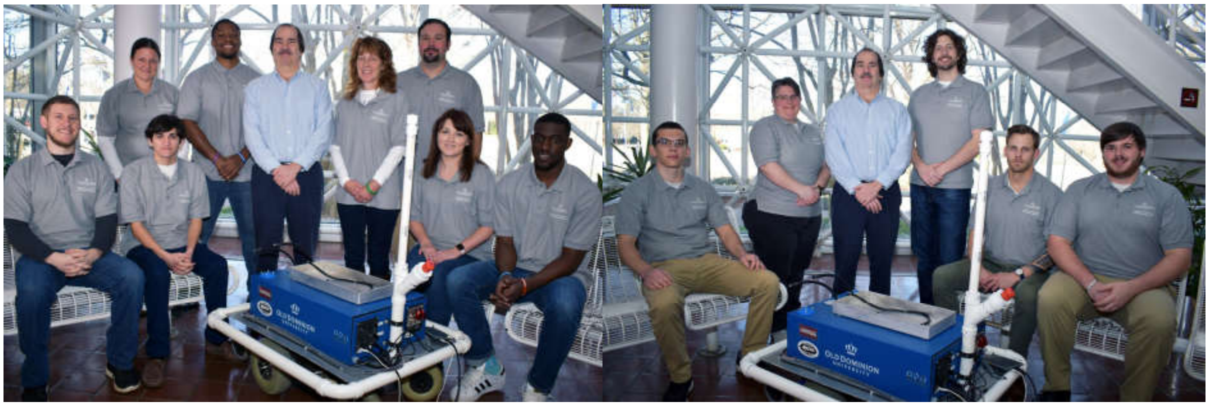
Old Dominion University

Batten College of Engineering & Technology

Intelligent Ground Operating Robot "IGOR"



Date Submitted: May 15, 2019

Team Captains:

Bonnie Lee Miley (bmil001@odu.edu)
Dana Pelland (dpell001@odu.edu)

Team Members:

Mark Boyd (mboyd002@odu.edu)
Craig Earls (cearl001@odu.edu)
Jackie Edmiston (jedmi001@odu.edu)
Jason Felton (jfelt013@odu.edu)
Susana Long (slong001@odu.edu)
Carlos Martinez (cmart022@odu.edu)
David Osafo (dosaf001@odu.edu)
Preslav Ivanov (pivan001@odu.edu)
Ryan Miller (rmill001@odu.edu)
Taylor Roy (troyx001@odu.edu)
Justin Rush (jrush004@odu.edu)
Chase Vogler (cvosl001@odu.edu)

Graduate Adviser: Adam Seay (aseay001@odu.edu)

Intern: Ryan Redmon (rredm002@odu.edu)

We, the students of Old Dominion University, aspire to be honest and forthright in our academic endeavors. Therefore, we will practice honesty and integrity and be guided by the tenets of the Monarch Creed. We will meet the challenges to be beyond reproach in our actions and our words. We will conduct ourselves in a manner that commands the dignity and respect that we also give to others.

In addition, and as the faculty advisor, I hereby certify that the design and engineering of the vehicle (original or changes) by the current student team has been significant and equivalent to what might be awarded credit in a senior design course.

Faculty Adviser: Dr. Lee A. Belfore (lbelfore@odu.edu)

Conduct of design process, team identification and team organization

Introduction

This paper represents the conceptual design of our vehicle, IGOR, and its components. The autonomous vehicle, IGOR, must navigate a course through lane detection and waypoint capabilities. While much of the functionality was stated to exist, verification and validation was necessary as was the safety and integration of the software. The quantitative measure of success will be the implementation of all of the qualifying requirements of the Auto-Nav competition.

Given the current implementation, the design methodology implemented is Continuous Process Improvement, specifically the DMAIC (Design, Measure, Analyze, Improve, and Control) principle. The DMAIC principle is considered a Lean Six Sigma (LSS) method looked upon favorably by government agencies because it focuses on the *reduction of waste* and *simplification* of processes. The letters within the principle itself are the abbreviations for the five phases of the Six Sigma Improvement [1]. By using this process, we started out by defining the existing project, developing a plan built around improving that project, defining the improvement process within the plan, and evaluating the existing project's progress [1]. From there, we needed to measure and collect all current data and analyze it [1]. Lastly, we need to move to improve the different items and systems on the project and organize ways to continuously measure and evaluate the progress accomplished within the project [1]. This methodology has allowed us to stay focused on the core design requirements of the competition. Once met, optimization of the vehicle's functionality related to the core design requirements will be addressed.

Organization

This project is the culmination of predecessor teams, operating independently and often with complete redesigns. This year we have moved towards vertical integration by incorporating the preparatory design team into the current design team and creating a unique collaborative effort that utilizes a Lean Six Sigma (LSS) process of gap analysis and Continuous Process Improvement (CPI) to ensure the teams' success. The competition team is interdisciplinary in nature and comprised of two distinct Electrical and Computer Engineering Senior Design II undergraduate engineering students and has primarily prepared not only to compete in the IGVC Design component, but also qualify for the Auto-Nav challenge.

Given the University's upward trend in the Auto-Nav challenge, Ms. Miley's teams' primary task is to ensure qualification for the 2019 IGVC. Ms. Pelland's team is focused on qualifying for the IOP and Cyber challenges, which is new for Old Dominion University this year.

The competition team has been split into four sub-teams, which are focused on documentation, safety, hardware, and software. Several team members share duties spanning several sub-teams, while some members have taken on more singular roles.

Name	Major	Primary	Secondary
Bonnie Lee Miley Dana Pelland	CpE[1]	Team Captain	Scheduling/Quality Control/Lab Tours/Fundraising

[1] Computer Engineering

	ECE[2]		
Susana Long Taylor Roy	CpE ECE	Project Lead	Documentation/Assignments
Craig Earls Ryan Miller	EE[3] EE	Hardware Lead	Safety Support
Jackie Edmiston Justin Rush	EE ECE	Hardware Support	Engineering Standards Finance
Carlos Martinez	CpE	Hardware Support	Requirements
Jason Felton Chase Vosler	ECE CpE	Software Lead	Finance
Mark Boyd Preslav Ivanov	CpE ECE	Software Support	Lab Tour Support
David Osafo	CpE	Safety Lead	Realistic Constraints

Table 1 – Tasks and Responsibilities

Design assumptions and design process

Our assumption in the design was to have an autonomous vehicle that would not only qualify but compete in the Auto-Nav competition. In conjunction, our design process involved integrating hardware and software together to tackle line detection and image processing. Furthermore, a replacement of the LiDAR and enclosure to prevent damage to the component was part of the design process. Safety lights were to be fixated on the vehicle as well as safety modifications.

The design expectations and process are in accordance with the 2019 IGVC Rules[4]. Our main focus is fulfilling the four requirements, detailed in this section, which were not met during the 2018 competition as well as ensuring there are no changes to those specifications met via regression testing.

Safety Lights

The vehicle is currently equipped with Light Emitting Diodes (LEDs) on the exterior of the vehicle chassis. These will be used to indicate the current state of the vehicle, which will be either *Standby* mode, *Autonomous* mode, or *Error* mode. The colors used to indicate these different modes are expected to be green, blue, and red.

Lane Detection and Following

The vehicle is expected to determine a lane in front of it. Once a lane is distinguished, the vehicle is to maneuver within that lane.

---

[2] Electrical Engineering and Computer Engineering (Dual Major)
[3] Electrical Engineering
[4] http://www.igvc.org/2019rules.pdf

Obstacle Detection and Avoidance

The vehicle currently detects objects with the use of the A3 LiDAR. After further software development, the vehicle is expected to move around obstacles that are detected by the A3 LiDAR while in a lane.

Waypoint Navigation

The vehicle must be able to detect different GPS waypoints with the help of the newly acquired cellular LTE. Once the waypoints are detected, the vehicle is expected to navigate to the waypoint and avoid any obstacles along the way.

Effective innovations considered in your vehicle design

There are several aspects of IGOR that could be improved upon related to both hardware and software. These improvements do not have direct bearing on the ability to qualify at the competition, but they do include both hardware and software implementations/improvements.

Implementing additional object detection hardware: The Microsoft Kinect® is an existing hardware already designed for 3D object detection and tracking. We could use this in addition to our LiDAR to assist with the detection and distancing functionality. In addition, this would be a safety enhancement. The LiDAR is top-mounted and we cannot detect distance of objects that are below it, creating a potential safety hazard for the vehicle, small children, plants and animals.

Utilizing license-free software alternatives: Since MATLAB[5] is a licensed product, an attempt to translate our implementation into a marketable product would be require substantial capital. An open-source alternative to MATLAB would remove that hurdle to a profitable venture.

Innovative technology applied to your vehicle

Replacing batteries: The car batteries currently being used are heavy, pose a fire hazard when charging, and current flow decreases as they discharge. These batteries can be immediately replaced with any alternative 12-volt battery and immediately provide better performance for the weight and, given the characteristics of lead-acid batteries, spend much less time charging.

Reducing footprint of electrical/electronic components: IGOR has a Standard-ATX motherboard installed which could be replaced by using a smaller motherboard, Next Unit of Computing (NUC), or a Raspberry Pi 3 Model B+ to reduce size of the electrical components. While it is possible for us to reduce the size of the electrical hardware, we cannot reduce the overall size of our vehicle due to design specifications of the competition. If we choose to reduce this footprint it could prove beneficial for uses outside the competition and future implementations.

Replacing drive motors: A smaller and more economical motor would allow IGOR to run for longer periods of time, thus increasing efficiency. IGOR currently operates with a 24-volt motor, and if we chose

---

[5] MATLAB is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks

to replace drive motors, we could easily reduce the number of batteries. We are in the process of procuring additional information related to the current motor and once available, we will determine the best way to move forward to ensure efficiency related to current draw. Innovative concept(s) from other vehicles were not designed into our vehicle.

Description of mechanical design

Overview

The current implementation is the same motorized wheelchair-based vehicle that participated in the 2018 IGVC in which predecessors successfully met four of the seven criteria for qualification. Navigation was done by creating a costmap which is populated using hits from the Light Detection and Ranging (LiDAR) sensor. A Mini-Dome security camera was used for edge and pothole detection. Maneuverability algorithms were used to process the edge detection, to ensure that the vehicle does not go out of bounds.

Decision on frame structure, housing, structure design

Suspension and weatherproofing

The current design is comprised of a wheelchair base, with a waterproof/weatherproof case, 2 Arduino(s), 2 Raspberry Pi(s), DC/DC boost and buck converter(s), Sabretooth Motor Controller, and a U-blox ZED-F9P GPS module. The existing case was designed by a predecessor team and is a waterproof case.

Software Development and Strategy

Previous teams began development on a Convolutional Neural Network (CNN) for object detection which we have deemed non-essential for competition due to time constraints, this will be revisited post-competition as an optimization to our *obstacle detection/avoidance* algorithm.

Implementation of edge detection was in place at the time we inherited the design. A large software development effort was necessary related to the core design requirements.

Outside of safety issues, the highest priority this team had was related to software development. With the predecessor team, the Regional Convolutional Neural Network (R-CNN) had been successfully trained to identify orange pylons kept within our lab. The R-CNN was to be utilized in many different aspects within the software. However, after analyzing resources and calculating the amount of man hours needed to implement it, the R-CNN was deemed too strenuous to continue as the main design for the software development team.

Instead, the software development team is developing four separate programs that are required to execute independently for qualification and must be integrated for competition. These programs encompass the functionality that satisfies the core design requirements: *lane detection/following, obstacle detection/avoidance, waypoint navigation* and *status indicators* (using the LED strip) from a software perspective.

Integration of these programs is necessary as the competition encompasses numerous challenges which will test all functionalities at varying different times while traversing the course. To ensure seamless integration, the software is being created with specific ROS variables, thus allowing each program the ability to communicate with each other.

ROS has different libraries that directly support development in C++ and OpenCV programming languages. The software development team has already designed and implemented programs related to the core design requirement, *lane detection/following* and *status indicators.* Several tests have been conducted and testing remains ongoing in an effort to determine the efficiency. The software development team is actively working on the two remaining core design requirements *obstacle detection/avoidance* and *waypoint navigation* to ensure proper testing and software integration to prove vehicle control during performance events.

The existing design incorporates Robot Operating System (ROS), C++, Python and OpenCV.

Description of electronic and power design

Overview

The vehicle uses the base of a motorized wheelchair for locomotive purposes and has a weather-resistant case for housing the computers and electronics mounted on top of the base. The primary sensor is an A3 LiDAR (*upgraded model*) which uses light from a laser as a radar system which is used for our primary *obstacle detection* which creates a 2D environment map in the ROS software. Secondary sensors include the Mini-Dome Network Camera which will also be used as a secondary obstruction detection sensor and a thermal imaging camera. An LED strip surrounding the vehicle will alert users and observers alike to the status of the vehicle, and an electrically actuated mechanical relay will cut power to the motors when triggered by the wireless or wired emergency stop (E-stop) buttons.

Power distribution system (capacity, max. run time, recharge rate, additional innovative concepts)

A new hardware bundle has been designed and built. The new bundle uses two (2) Raspberry Pi B+ instead of a PC motherboard for data processing. This amounts to a 120 square inch footprint reduction. This smaller footprint will allow for additional multiple processors as required in the future. The new bundle also includes a new GPS device which uses correction data over a cellular internet connection to increase our location precision from over 1.5 meters to 10 millimeters. This is in direct response to our problem of not qualifying in last year's competition due partially on a failed GPS *waypoint navigation* test. A new 60 amp per channel motor controller replaces a 20 amp per channel one which will allow our wheel motors to run at full speed.

To allow for the gathering of GPS correction data we have added a cellular Long Term Evolution (LTE) hat to one of our microprocessors. A Power over Ethernet (POE) switch has been added to minimize the amount of wires traversing our case. The POE switch allows us to remove the power cables to our cameras and the Raspberry Pi. Using the POE switch and Pi allows us to remove a 58 square inch power supply and replace it with two (2) 10 square inch DC/DC converters. These changes clearly reflect a design that is an efficient use of power, material and space.

Electronics suite description including CPU and sensors system integration/feedback concepts

A proper wiring diagram was designed prior to working on this project and implementing changes.

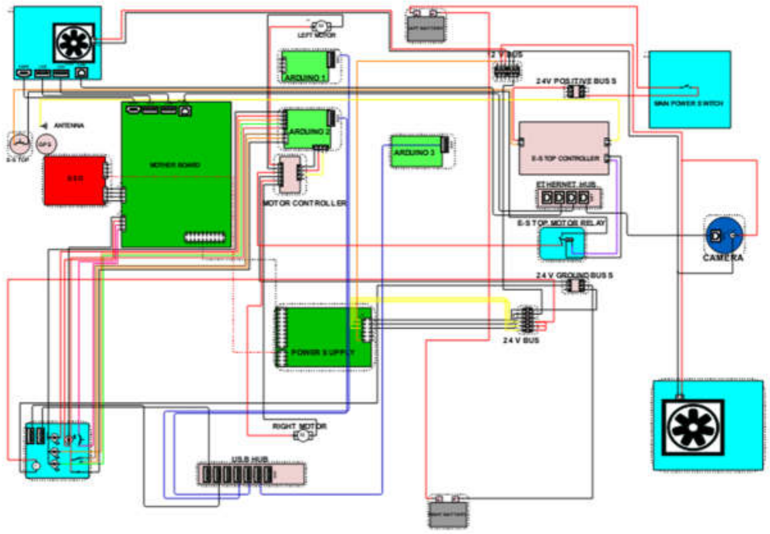


Figure 1 – Wiring Diagram

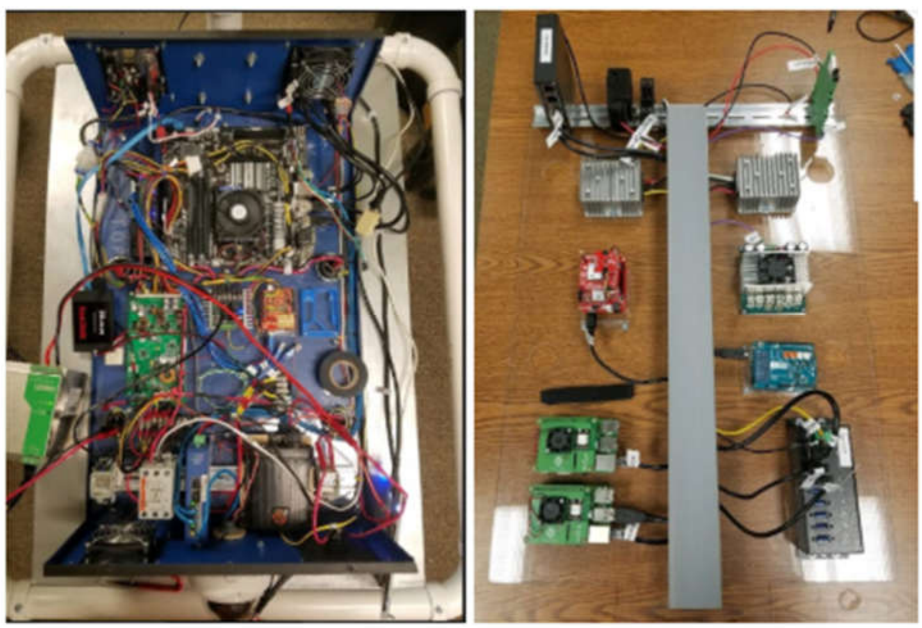


Figure 2 - Before and After Component Layout

Safety devices and their integration into your system

One of the first issues addressed was the damaged LiDAR. The equipment itself has been upgraded and the primary concern was to ensure the new LiDAR was adequately protected going forward. A new mount satisfied our equipment safety concern. This mount and a screw on cover was designed and 3D printed by our Hardware Lead. The cover will provide additional protection while not in use. Integration testing of the LiDAR with the vehicle is in process.

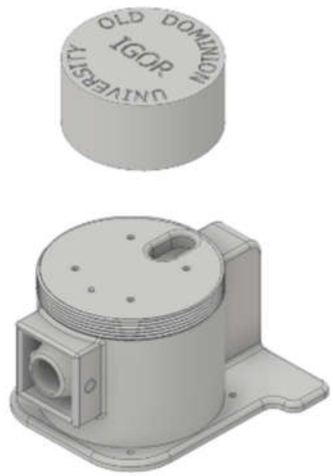


Figure 3 - New LiDAR Mount and Cover

Previous use of an active camera was deemed inefficient and exploration into the cost effectiveness of passive night vision solutions (not requiring IR or other illumination sources) was done. Additional funding, outside of our major sponsor, was obtained through the College Projects Committee and we have procured an adequate passive camera. Image intensification was discussed, but given the priority of the core design requirements, was passed to our successor team for implementation.

The Pixy cameras previously used for lane following and pothole detection were also damaged, deemed inefficient, and removed, the 4MP IR Mini-Dome Network Camera used in conjunction with the Pixy cameras will be the primary means for *lane detection/following* in order to meet the core design requirements in a timely fashion.

Description of software strategy and mapping techniques Systems Integration

Overview

ROS is the software previously used for device management, sensor data, transfer of messages between publishers and subscribers, and in conjunction with the Mini-Dome Network Camera allowed for the publishing of images using OpenCV. The vehicle utilizes ROS, C++, and OpenCV within a program which allows the vehicle to detect a lane in front of it. Once detected, the vehicle travels through the lane. This is done by transmitting values to the motor controller and the not only propel the vehicle forward within the lane, but also help maintain the vehicle's position around the center of the lane. In addition, an A3 LiDAR has been integrated with ROS to detect objects around the vehicle.

Obstacle detection and avoidance

The vehicle is expected to determine a lane in front of it. Once a lane is distinguished, the vehicle is to maneuver within that lane. The vehicle currently detects objects with the use of the A3 LiDAR. After further software development, the vehicle is expected to move around obstacles that are detected by the A3 LiDAR while in a lane.

Software strategy and path planning, map generation and goal selection and path generation

A GPS is required for *waypoint navigation*. However, with an accuracy of 1.5 meters, the previous GPS lacked the accuracy to complete the competition and the waypoints were not being translated properly. The GPS needed to either be replaced or an advanced algorithm developed that could process additional data to meet the core design requirement. Given the software development team needed to redefine the *lane detection/following* algorithm, a new GPS was procured.

The vehicle must be able to detect different GPS waypoints with the help of the newly acquired cellular LTE. Once the waypoints are detected, the vehicle is expected to navigate to the waypoint and avoid any obstacles along the way. Additional creative concepts have been placed on hold due to the importance of qualifying for competition.

Description of failure modes, failure points and resolutions

Vehicle failure modes (software, mapping, etc.) and resolutions

Safety Lights

The vehicle is currently equipped with Light Emitting Diodes (LEDs) on the exterior of the vehicle chassis. Requirements dictate a status indicator on the vehicle. These LEDs operate in that capacity and we have utilized the following colors and corresponding modes: green indicates Standby mode, blue indicates Autonomous mode, and red indicates Error mode. Standby mode correlates to IGOR operating as expected and/or awaiting input from internal command or the user. Autonomous mode will be situated around all times where IGOR is moving free of user control. This will encompass times where IGOR is solely using software to accomplish different tasks within the competition. Any time an error is encountered, either while completing a task, in Standby or Autonomous mode, the red lights will engage, indicating the shift into Error mode.

Within the software, different checks will be in place to ensure proper switching between the different colors, thus helping the team identify vehicle failure points (electronic, electrical, mechanical, structural, etc.) and aid in the resolutions. This means that several programs will be talking directly to one program, which will handle the changing of the LEDs colors. This means that checks will be placed within these different programs to tell the main program what color needs to be present on the LEDs. If any running program encounters an error, the LED program will switch to the red indicator, which will indicate “Error Mode”. If the vehicle has one or several autonomous programs running, the LED program will switch the LEDs to blue, which will signal “Autonomous Mode”. If the vehicle is on, but there are no autonomous programs running, the LED program will know to display the green indicator, which indicates “Standby Mode”.

This is a component that will be tested later in the design phase. It will be tested after the different programs are created and contain different checks, which will be used to indicate the vehicle’s current state.

Further software development is still being conducted, where future results will be obtained from proper testing.

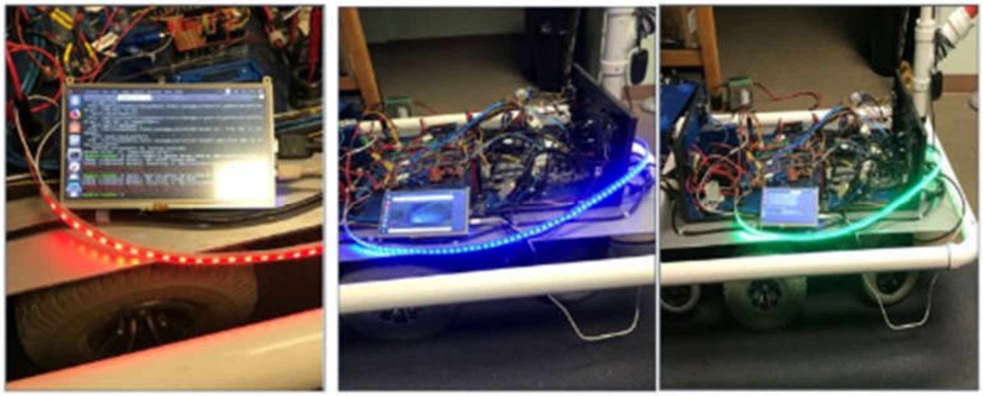


Figure 4 - Safety Lights

All failure prevention strategy, testing (mechanical, electronic, simulations, in lab, real world, etc.) and vehicle safety design concepts are addressed elsewhere in this document.

Simulations employed

Simulations in virtual environment

Physical testing has begun but it is often costly and time consuming. Given this, we are looking into options related to modeling and simulation solely with RVIZ.

Theoretical concepts in simulations

There currently exists a program for LEDs which will be utilized as a reference point for further software development related to using these LEDs for debugging. The GPS has been procured for waypoint navigation and we are projecting this will be the last action item that will need to be tasked and tested prior to competition.

Performance Testing to Date

Component testing, system and subsystem testing, etc.

Lane Detection and Following

The vehicle previously was unable to detect and maneuver within a lane. However, two developed programs allow for IGOR to detect a lane and maintain movement within the lane. The base code for this program was obtained from GitHub, Inc., [5] which contains open source code dealing with lane detection. This code was manipulated and developed further to function within ROS and work directly with a developed PID controller for IGOR's motor controller in order to satisfy lane detection and following.

Within creating lane detection code independent of the open source code, the software was able to produce lines that would outline a lane in front of the vehicle. However, the lines were unable to properly fit to the lane when the vehicle encountered curves or the video feed contained dimly lit areas from shade or an overcast day. However, with the open source code, these problems were taken into account. This lane detection software detects a specific lane and then manually flips the images, as if someone is looking at the lane from an aerial view. After being flipped, certain manipulations are conducted on the image to ensure that a dimly lit image will not provide problems to the lane detection software. In addition, the software had curve fitting functions to follow a curved lane.

Once this software was manipulated and developed further to work with ROS, a PID controller was created to help provide movement to the vehicle, while it is detecting a lane. The input for the PID controller was the value that shows how far off the center of IGOR was from the center of the lane. The PID controller then takes this value and performs calculations to produce values which will be sent to the motor controller. These values will directly alter the vehicle's movement, which will keep it within the middle of the lane. With the lane detection and PID controller working in conjunction with each other, the vehicle will be able to continuously check if it is in the center of a lane and correct its position if the check results in an off-centered vehicle.

Currently, preliminary tests have been conducted on the lane detection code. These tests have been conducted inside a confined place, with broken and solid white lines. These early tests have produced both favorable and unfavorable results, where methods to fix the unfavorable results have been formulated and are being developed. In later tests, the vehicle will be moved outside in larger areas, where it will experience well-lit and dimly lit areas.

The early tests conducted produced very promising results. The lane detection algorithm does need more proper tuning to fit properly to any size lane. Proof of lane detection software detecting a lane, whether it is made up of solid or bold lines.

As noted in the Figure 5 and Figure 6 there is a value placed within the top left portion of the lane detection program. This is indicating how far the center of the vehicle is from the middle of the lane. This value will also indicate if it is off on the left or the right side of the middle of the lane. After several test runs, the car showed some unfavorable results that showed the car move through the lane, but heavily favored the right side of the lane.

Figure 5 – Lane Detection with Adequate Curve Fitting for Solid Lines

Figure 6 – Lane Detection with Broken Lines

Results were small excerpts from big logs of data that showed, the general readings from two different test runs, where the test lane was approximately 1.25 meters wide.

0.26 meters to the Right
0.2575 meters to the Right
0.245 meters to the Right
0.315 meters to the Right
0.315 meters to the Right
0.315 meters to the Right

Table 2 – Test Run 1

0.0175 meter to the Right
0.0175 meter to the Right
0.015 meter to the Right
0.015 meter to the Right
0.0175 meter to the Right
0.0175 meter to the Right

Table 3 – Test Run 2

At first, the software development team hypothesized that the problem was due to improper tuning within the PID controller. Improper tuning can produce these type of results (heavily favoring either the left or right side of a lane). However, after further inspection, the software development team understood that the car was losing the lane every time the left or the right lane was hidden from the vehicle's camera view. This placed the vehicle in a state where it is no longer following a lane, but just constantly moving until either the user stops the vehicle or the vehicle detects a new lane to follow. Further software is being developed to retain an image of the detected lane as reference, even if the camera loses one side of the lane it is maneuvering within.

Obstacle Detection and Avoidance

The vehicle currently detects objects, however, it cannot properly navigate around them. However, by utilizing the new A3 LiDAR, the different objects detected are being converted to data points. These points will be used as input into the Navigation Stack within ROS. This will produce a program which tells IGOR that obstacles are in its path and plot a course to avoid those obstacles.

When the vehicle has plotted a new course, different values will be given to the motor controller. These values will deal directly to different movements the vehicle will utilize to properly avoid obstacles. These movements include going straight, turning moderately and sharply left, turning moderately and sharply right, going backwards, and stopping. Furthermore, the obstacle avoidance software will be developed to stop the vehicle when it reaches a certain obstacle. This is done to allocate the vehicle adequate time to plot its next moves to avoid an obstacle. However, as the software becomes more developed and utilizes more real-time analysis methods in the obstacle avoidance, the stopping phase will be removed.

With the use of the A3 LiDAR, different objects within a confined area can be detected. When the obstacle is detected, this information is placed into data points, which can be used by different programs. However, further software needs to be developed to utilize these points, before any testing can be conducted on the obstacle avoidance program.

Further software development is still being conducted, where future results will be obtained from proper testing.

Waypoint Navigation

The vehicle previously was able to receive a waypoint coordinate using a GPS. The GPS previously used was however not able to provide an accurate enough location to navigate to. A newer GPS was purchased to be able to increase the accuracy of the waypoint location so that IGOR can navigate to the waypoint, or a minimal distance from it. When the GPS provides a waypoint for IGOR, a course or general direction will be established to navigate. This path will work in unison with the lane detection and obstacle avoidance and will not be directly providing motor controller instructions.

The new component still needs to be installed on the vehicle. This hardware installation and further software development needs to be conducted before any testing can be conducted.

Further hardware installation and software development is still being conducted, where future results will be obtained from proper testing.

Initial Performance Assessments

To date, everything is working in sync except for waypoint navigation. Ms. Miley's team is currently focused on integrating waypoint navigation into the current system, while Ms. Pelland's team continues to focus on meeting the requirements of the IOP and Cyber challenges.

Acknowledgement

Without the ongoing funding of the Combat Capabilities Development Command C5ISR Center (*formerly known as CERDEC)*[6], project sponsors including Protocase[7], Synthetic Solutions USA - AMSOIL®[8], the College Projects Committee, project alumni as well as the continued support of our advisors, and mentors (*faculty, industry and graduate student*) this project would not be possible.

6 https://www.cerdec.army.mil
7 https://www.protocase.com/ - Custom Electronic Enclosures
8 http://syntheticsolutionsusa.com/

References

[1] “Continuous Process Improvement (CPI) and Lean Six Sigma (LSS).” Continuous Process Improvement (CPI) and Lean Six Sigma (LSS), www.dau.mil/acquipedia/Pages/ArticleDetails.aspx?aid=3ac9f6c4-bede-4b63-9aac-7f4ba3c37162.

[2] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, “A review of semantic segmentation using deep neural networks,” International Journal of Multimedia Information Retrieval, vol. 7, no. 2, pp. 87–93, 2017.

[3] J. Dai, K. He and J. Sun, "BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation", 2015 IEEE International Conference on Computer Vision (ICCV), 2015.

[4] A. Khoreva, R. Benenson, J. Hosang, M. Hein and B. Schiele, "Simple Does It: Weakly Supervised Instance and Semantic Segmentation", 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[5] https://github.com/HsucheChiang/Advanced_Lane_Detection