Old Dominion University

Department of Electrical and Computer Engineering

“Little Blue”

2025 IGVC Design Report

Submitted: 05/15/25

Faculty Advisor: Dr. Lee Belfore

Senior Team Captain: Tyler Cason – tcaso001@odu.edu

Junior Team Captain: Seth Huthmaker – shuth001@odu.edu

Senior Team Members:
Jared Cochran – jcoch004@odu.edu
Jarub Ellenwood – jelle004@odu.edu
Christopher Griffin – cgrif015@odu.edu
Julia Miller – jmill090@odu.edu

Graduate Team Member:
Raphael Pamie-George – rpami001@odu.edu

Junior Team Members:
Dan Furnary – dfurn002@odu.edu
Zach Gamble – zgamb002@odu.edu
Stefon Mclean-Burrell – tmcle012@odu.edu
Yasmany Paucar Chambe – ypauc001@odu.edu
Thuy Nguyen – tnguy080@odu.edu
Lynnet Rich – lrich023@odu.edu

ODU Auto-Nav entry

I certify that the design and engineering of the vehicle of the current student team has` been significant and equivalent to what is awarded credit in ODU's senior design course sequence.

5-15-25

Lee A. Belfore II
Associate Professor
Old Dominion University

Table of Contents

1. INTRODUCTION

This project aims to enhance the "Little Blue" autonomous vehicle to meet the 2025 Intelligent Ground Vehicle Competition (IGVC) standards. Currently, the vehicle faces challenges, including software compatibility issues that hinder the integration of new components, such as cameras. These problems delay progress and require extensive debugging. The machine learning system, critical for decision-making, is underperforming and needs refinement, while the data collection process for training the neural network is inefficient. Additionally, simulating performance in Gazebo proves difficult, limiting pre-test validation. By incorporating advanced devices and optimizing machine learning algorithms, we seek to improve data processing and decision-making accuracy. This effort demands expertise in electrical engineering, computer engineering, and machine learning to deliver a highly autonomous vehicle that excels in the competition. Designed to comply with traffic regulations, Little Blue will navigate obstacles, stop at signs, ensure pedestrian safety, and maintain speeds of 1–5 mph in autonomous mode. In manual mode, it achieves a maximum speed of 5 mph with full 360-degree maneuverability for tight spaces. Safety remains paramount, with local and remote emergency stop options ensuring reliability in both competition and real-world scenarios. Equipped with LiDAR, cameras, IMUs, and GPS, the vehicle gathers real-time data to navigate dynamically, adapting to obstacles and light rain with precision.

2. DESIGN CONSIDERATIONS AND TEAM COMPOSITION

2.1 Team Composition

The team is comprised of Department of Electrical and Computer Engineering (ECE) students from the Frank Batten College of Engineering and Technology. On the team there are the recent graduates, a current graduate student, and multiple undergraduate students. The recent graduates comprise of Jarub Ellenwood and Christopher Griffin, who most recently participated in Fall 2024, and Tyler Cason, Jared Cochran, and Julia Miller, who all are fresh graduates from Spring 2025. The active graduate student is Raphael Pamie-George. Finally, the current undergraduates (junior team) consist of Seth Huthmaker, Dan Furnary, Zach Gamble, Stefon Mclean-Burrell, Yasmany Paucar Chambe, Thuy Nguyen, and Lynnet Rich. All of the current graduate and undergraduate students were most recently working on the vehicle in Spring 2025 and will form the core to take over the project for Fall 2025. Tyler Cason serves as the Senior Team Captain, with Seth Huthmaker serving as the Junior Team Captain. All junior team members are seniors. In total, the teams contributed around 550-600 hours of time to this project.

2.2 Design Considerations

2.2.1 Design Assumptions

The development team of Little Blue approached the project with the following information kept in mind:

1. The vehicle could not drop below a 1 MPH speed, nor could it go faster than 5 MPH
2. The vehicle must be developed not to leave the marked course, and must not hit any obstacles
3. The vehicle must be able to drive independent of human control
4. The vehicle must be able to handle a 20-pound payload.
5. Our software must be developed with optimization and development by future teams in mind – it must have good documentation, and should not hinder future codebase evolution
6. What can be made weatherproof should be made weatherproof. As the competition will be outside, and potential operational weather may include light rain, our vehicle must be able to handle such an environment.

By following such, the team kept competition rules in mind while also ensuring an attempt to follow best practices in software development.

2.2.2 Design Process

Overall, the design goals of Little Blue were set up with two points in mind. One is the IGVC rules, defining important design constraints such as minimum and maximum speed. The other is that the main control system will be driven by a neural network. All the software and design goals were created with the purpose of supporting this model as best as possible. The software was built in a modular fashion to facilitate ease of transportation of new sensors or datapaths that may be implemented in the future. The software design process was centered around test driven development to ensure all functionality and reliability was being met exactly as needed. When designing the control system for the motors, care was taken to ensure that the IGVC competition rules for minimum and maximum speed were being met, while allowing for safe speedup and slowdowns. Overall, Little Blue takes advantage of the iterative nature of the project with improvements from previous years being expanded upon and with new features and additions to support a better functioning design.

3. LITTLE BLUE SYSTEM ARCHITECTURE

3.1 Identification of System Components

3.1.1 Significant Mechanical, Power and Electronic Components

The mechanical components of Little Blue lie mainly with the wheelchair base of the vehicle. Specifically, the motors which drive Little Blue are the main mechanical components.

The main power components lie in the batteries that power the vehicle, and an inverter to allow for a stable 120V AC option. The two batteries that power the vehicle are 12V and 24V, and these power the electronics suite and the motors, respectively. The 120V option provided by the inverter helps run the necessary computer accessory equipment, which includes the display monitor and USB hub. The network switch also provides 48V POE to other electronics, such as the Raspberry Pi 4.

The electronic components Little Blue contains are mostly sensors, but there are non-sensor components (see parts lists for specific models). The list of sensors includes an IMU, LiDAR, a GPS antenna, cameras, and odometer pods. Also located on Little Blue is a network switch which handles all ethernet communication, the Nvidia Jetson, Arduino and Raspberry Pi for controllability and autonomy, and the motor drivers.

3.1.2 Safety Devices

The safety devices included on Little Blue can be divided into emergency stops (e-stop) and manual motor releases. Both devices allow for the team to work and/or operate Little Blue without fear of rogue controller inputs or controller input buffer delays. Beginning with the e-stops, as dictated by competition rules, Little Blue has a manual e-stop within reachable distance for anyone near Little Blue. Likewise, there is also a wireless e-stop with handheld remotes for stopping Little Blue from a distance. Switching to the manual motor release, as our vehicle is built on a wheelchair base, the two motor release handles remain. They are located under the front of the vehicle and are used when working underneath or near the vehicle during development work.

3.1.3 Software Modules

Little Blue is run on top of the Robot Operating System (ROS). ROS is a software framework which allows for easy and efficient development of robot platforms. It features many built-in libraries and functions for common sensors and general mathematics. ROS provides mechanisms for parallelizing reading/writing of data and moving messages between these nodes very easily. Little Blue additionally features a neural network to help maintain the course between bounds. It begins by selecting the processing device through CUDA or MPS and defaulting to CPU if neither option is available. It then prepares a dataset, incorporating steering and throttle data alongside training images. The model architecture consists of convolutional layers for feature extraction and fully connected layers for prediction. Training occurs over 20 epochs, utilizing the Adam optimizer and MSELoss for weight adjustments. After each epoch, the model is saved, allowing for iterative refinement and improved accuracy in future training cycles.

4. EFFECTIVE INNOVATIONS OF LITTLE BLUE

Little Blue utilizes a neural network to navigate the competition course. Training a neural network to many efficient runs of the competition course, along with many variations, allows Little Blue to adapt to varied challenges and chooses the most efficient method of navigating them regardless of what is in front of it. Little Blue’s neural network is trained utilizing behavioral cloning to mimic the driving of the input data given to it. This year it was trained on a variety of drivers to allow Little Blue to develop a unique driving style based on each of the different drivers it was trained on.

An additional innovation from last year’s version is a switch from Python to C++ in most of the ROS code. The main advantage of switching to C++ is the additional performance benefits gained from the added control over memory, and from directly running code on the CPU rather than through an interpreter like Python. In data-intensive tasks this has improved the runtime significantly while also utilizing fewer overall resources and memory, leading to a more efficient design. Another added benefit is the compile-time error checking and structured error handling that C++ provides. This allows for much more reliable design over Python in critical tasks.

5. DESCRIPTION OF MECHANICAL DESIGN

5.1 Overview of Mechanical Design

Little Blue utilizes a powered wheelchair base as its platform, with an enclosed structure added onto it to support the encased electronics. Also built onto Little Blue is a platform at the rear of the vehicle which supports the controls for development, training, and competition access. Included on this platform is the manual E-stop, putting the vehicle in compliance with the IGVC E-stop height requirement.

Returning to the enclosed structure below the platform, the encasing material also supports the location for the required competition payload. The location of this holder is intentional, as it keeps the payload from interfering with the cameras, which are mounted onto the platform. On this level of the vehicle, access to the batteries and internals of the wheelchair base is made possible with two hydraulic struts, which lift the entire enclosed electronics structure out of the way. This access is only capable when there is no payload, as the terminal placement of the encased structure is right below the edge of the platform.

5.2 Description of Structure and Framing

Revisiting section 5.1, the structure of the vehicle is built on top of a powered wheelchair base. At that same height level, the battery bay for the vehicle is also located. Moving upwards from there, the vehicle next has an aluminum cap to the motors and battery bay. This aluminum cap can be moved out of the way by lifting it up from the front of the vehicle. This is kept in the air with the support of two hydraulic struts. The vehicle's electronics suite is largely located in a box structure on top of the cap. It has a removable shell, which allows easy access to the electronic components of the vehicle. On top of this shell, a holder for the competition payload is present. Moving back to the base, a structure for a platform is built from the base to about two feet above the base. The platform is where the human-usable components are located, i.e. the manual emergency stop, monitor, keyboard and mouse, power switches, etc. This is also where the cameras for the vehicle are mounted.

5.3 Description of Weather Proofing

Little Blue has multiple types of weather proofing to provide protection from any potential rain during competition. First, all the electronics are covered by the shell, which has weather-proofed gaskets on the outside to provide crucial protection to the power delivery and data communication systems. As mentioned earlier, all on-board controls and electronics are also protected by this shell. Second, certain equipment procured for Little Blue was picked with weather resistance built in. The monitor used is a capacitive touch Point of Service monitor with significant weather resistance and luminance to provide quality daytime usage, along with protection from weather.

6. DESCRIPTION OF ELECTRONIC AND POWER DESIGN

6.1 Overview of Electronic and Power Design and Component Descriptions

The electronics on board are part of two different power sources, one powered by a 12V DC battery, the other powered by a 24V DC battery. For those electronics not powered by 24V power, a power inverter converts it into 120V AC power source to be converted into the required voltages internally. Included is a high-level diagram to better explain the interconnection of Little Blue's electronics with power.

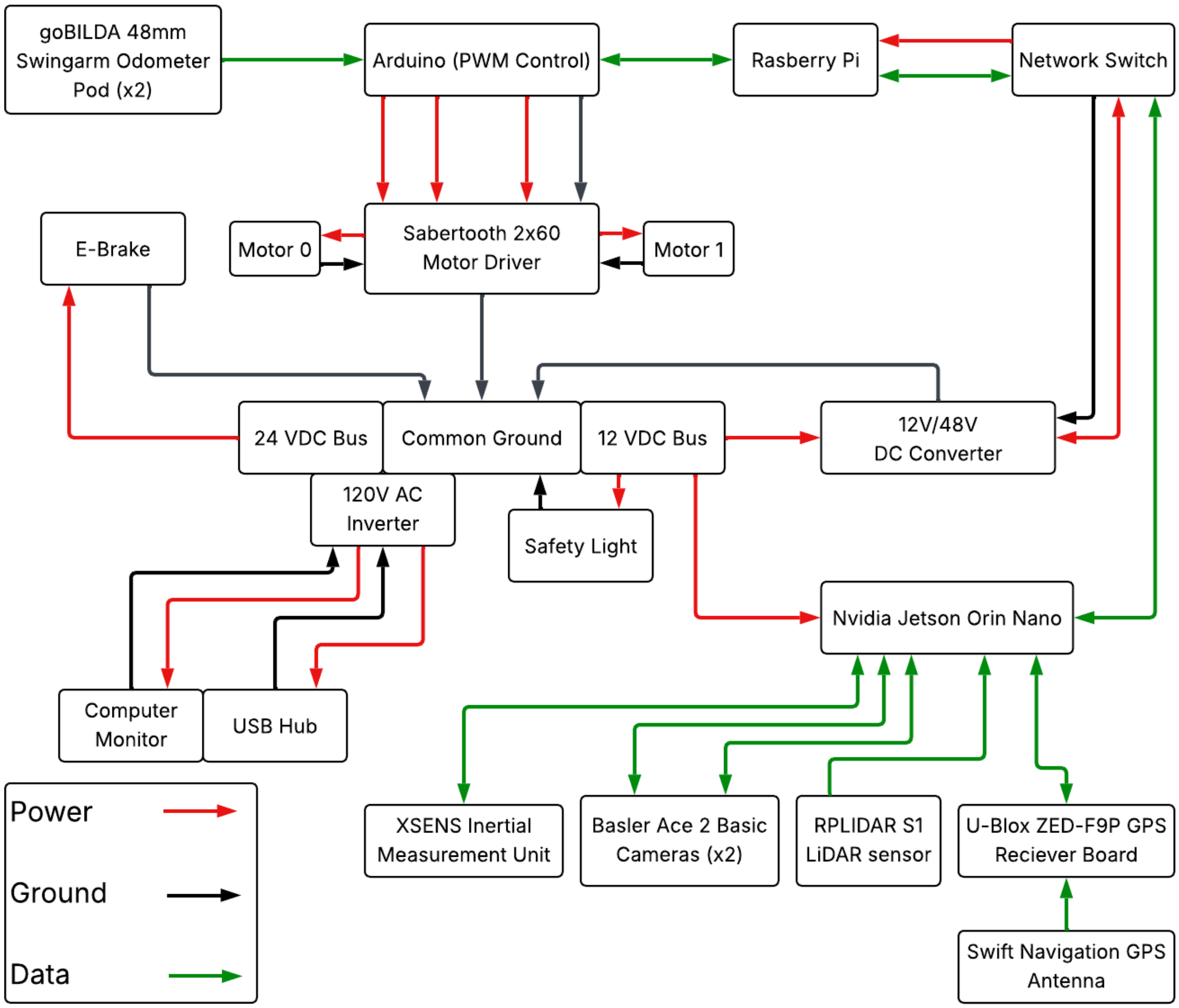

Figure 1: High Level Connection Diagram

Above shows the interconnections between all the control electronics on-board Little Blue. Not shown is the 24 VDC Bus being connected to an inverter to power other miscellaneous items like the monitor. The 12 VDC Bus powers most of the electronics on board, powering the Jetson Nano which runs much of the computation on board Little Blue. It is responsible for running the neural network and interfacing with all the sensors as can be seen at the bottom. All the sensors on board are powered by the Jetson.

6.2 Characteristics of the Power Distribution System

The battery life has been tested to last for at least a full day of functionality during previous testing days. This entails at least 8 hours of functionality, but it is likely extended to at least 10 hours. This is due to the typical usage of the battery not going above 100W, which means with most of our equipment running off a 12V battery with a capacity of at least 1300

Wh, the duration is easily 10 hours while accounting for spikes from relatively infrequently used pieces of equipment, such as the motors. The 24V battery goes to the inverter, and as such is not as frequently depleted as the 12V battery.

As partially mentioned in the last paragraph, the capacities of the two batteries are a minimum of 1300Wh for the 12V battery and 2000 Wh for the 24V battery. Through those capacities, and the average charging capabilities of their respective chargers (90W and 116W), it would take around 14.4 hours and 17.2 hours respectively to recharge the batteries from a theoretical 0Wh.

Additionally, it is important to note the fuse panel on the outside of the shell. This fuse panel allows us to ensure the connected systems will not have a catastrophic failure which cascades through all of our systems, and instead ends with the one fuse blown, and potentially the loss of one component. This helps us keep our vehicle and ourselves safe.

6.3 Electronics Suite

6.3.1 Sensors

The electronics suite sensors include the GPS antenna, LiDAR, an IMU, two cameras, and two odometer pods. These each provide a different version of support to Little Blue, but that support can best be divided into two categories. Those categories are obstacle detection and speed compliance. The LiDAR and cameras are used to feed into the neural network to provide obstacle detection to allow the network to sense its surroundings and decide to maneuver around the obstacles. The IMU and odometer pods are used to ensure speed compliance by both providing an acceleration-based velocity and a direct wheel measurement for velocity. These are used to ensure Little Blue is both above the minimum speed requirement and below the maximum speed requirement.

6.3.2 Computers

The computing on-board Little Blue is run by 3 main systems. An NVIDIA Jetson Nano, a Raspberry Pi, and an Arduino. The Arduino is responsible for communicating to the motor controller and allowing Little Blue to move. The Raspberry Pi facilitates communication between the Jetson and the Arduino by responding to game controller inputs or reading from the neural network’s control inputs. The Jetson is what reads from each of the sensors and feeds them into the neural network accordingly. It is responsible for making each inference and passing that information along to the Raspberry Pi to move Little Blue.

6.4 Mechanical and Wireless Emergency Stop(E-Stop) Systems

Both the wireless and mechanical E-Stop systems on Little Blue work by de-energizing the motor driver on board Little Blue, stopping all power to the motors. The mechanical E-Stop utilizes the 24V rail on the battery to a relay which opens the normally closed contacts and removes power from the relay. The wireless E-Stop works by sending a 433.92 MHz radio signal to a receiver on board, similarly, opening the normally closed contacts upon receiving the signal and stopping all power to the motor driver. Both contacts will be open until both E-Stop signals are set to their “off” position.

7. DESCRIPTION OF SOFTWARE SYSTEM

7.1 Breakdown of Sensor Usage and Data Processing

Little Blue utilizes multiple different sensors in its neural network to process its surroundings. The main sensors are two cameras which are combined to provide Little Blue with a stereoscopic image of the world. This is combined with LiDAR data to provide accurate ranging information and IMU data to provide velocity information to Little Blue. Little Blue’s internal software stack utilizes the Robot Operating System (ROS) to ingest the sensor information. Either manufacturers provided or standard ROS nodes were used to bring the sensor data directly into ROS, whilst custom nodes are utilized to clean the data coming in and feed it into the neural network.

7.2 Operating Modes

Little Blue has three operating modes. The first is the remote-controlled version which allows a game controller to provide steering and throttle inputs to Little Blue without any autonomous functionality. The second allows Little Blue to drive on its own according to the decisions generated by its neural net model. Finally, the third mode is Little Blue’s training mode, which saves all sensor and control data to a directory for later use.

7.3 Decision Making Process

As per the design of Little Blue, all decisions are made directly by the neural network and executed by the rest of the system. Little Blue is trained on data collected by doing many test runs of portions of the competition course, or varied sections and obstacle courses. The purpose of this training is to provide the neural network with enough training data to allow it to make informed decisions about how it should tackle an obstacle. This includes determining the best path around an obstacle and taking into consideration future obstacles.

8. CYBER SECURITY ANALYSIS USING RMF

The NIST Risk Management Framework (RMF) is a framework for integrating good cybersecurity practices into development. It provides 7 main steps to prepare a product for a cyber attack. These steps are, Prepare, Categorize, Select, Implement, Assess, Authorize, and Monitor. Although all the steps are of equal importance, the main ones that pertain to this document are the Select, Implement, and Assess steps. The Select step is about choosing the specific types of controls laid out by the NIST that should be implemented to protect valuable information. The Implement step is quite intuitive, it consists of implementing the controls identified in the Select Step. Finally, the Assess step is about ensuring that the controls implemented are operating correctly. In the case of Little Blue, the main privacy concern is with network credentials and git credentials. As the software is hosted on an external repository with restricted access, an authorized user must put in their credentials to clone the repository to Little Blue. If an attacker were to get ahold of these credentials, they would be able to view and modify the contents of the repository or access the same networks that Little Blue is able to access.

9. ANALYSIS OF COMPLETE VEHICLE

9.1 Lessons Learned During Construction and Software Development

During software development, a large issue that often came up was the ability to reliably test the software and deploy it properly. Integration testing a large complex piece of software that requires access to external hardware is always a large pain point for many projects. The solution to this problem was to create mock hardware nodes and set up a consistent hardware interface. This way the rest of the system could function exactly as intended while providing synthetic inputs designed to test the system to its capabilities. This would help find edge cases, fix crashes, and determine the overall reliability of the system. Deployment was solved by Docker and GitLab. Utilizing Docker allows for all the software and its necessary environment to be reliably set up the exact way every time. GitLab allowed for proper versioning of the software itself, and a convenient place to store all dependencies and changes to the Docker. This way an environment could be built just by cloning a single repository.

9.2 Major Hardware Considerations and Redundancies

As with any autonomous vehicle, hardware is extremely important to the overall performance of the vehicle. The major considerations considered when picking out the hardware for the project were how weather proofed, or proof-ability, of the item and how much overall data bandwidth the hardware would require. As with most sensors, the higher resolution produces diminishing returns on the usefulness of each additional data point. In addition to the diminishing usefulness, the more data that needs to be processed requires additional time spent processing it. For Little Blue, the more inferences the neural network can create allows for smoother operation and nuanced movement to produce an efficient path through the course.

9.3 Safety, Reliability, and Durability Considerations

When working with Little Blue, safety was a large concern as a wheelchair can produce a large amount of torque to carry hundreds of pounds. Always ensuring both the mechanical and wireless E-Stops were functioning correctly helped reduce the risk of injury. In addition, procedures were put in place to ensure that two people at minimum would be required to operate Little Blue, with both users holding a wireless E-Stop in case of emergency.

Reliability was tested in several ways. The software was testing through full-scale integration testing utilizing mock hardware. The mock hardware allows for synthetic input data to be provided to the rest of the system to test specific functionality. This synthetic data can also be manipulated to push the system to its limits to ensure that under extreme circumstances the systems will not fail. In addition to software reliability testing, during data collection the system is effectively fully end-to-end tested numerous times for hours on end. In this scenario, anything that goes wrong can be fully documented and controlled to be fixed later down the line.

9.4 Competition Day Key Failure Points and Redundancies/Resolutions

As Little Blue is driven entirely by a neural network, all sensors must be functional for the neural network to function properly. At the very least, it must be receiving the exact number of inputs it expects. If one of the cameras or the LiDAR goes out, that is a large portion of the input data completely removed from the network. This is handled by providing the neural network with blank/repeated inputs if something goes awry. Although the neural network may perform much worse than with all sensors active, letting the bot completely fail if one sensor fails is not an option.

Environment reproducibility is a huge burden for these types of projects. If one directory gets moved, or an environment variable is set up incorrectly, then it is entirely possible that the software may just fail. This is handled by Docker to containerize the software of the project. Although the outer environment still needs some setup, most of the configuration and environment is handled through Docker itself to reproduce the exact same setup every time. This eliminates a large failure point if the environment would need to be rebuilt.

9.5 Predicted Performance of Little Blue

The performance of Little Blue is entirely dependent on two main factors. The quality of the training of the model, and the number of inferences that can be done per second. The quality of the training is dependent on properly cleaning and preparing the data, as well as having a large amount of sample data. Little Blue is currently limited to making inferences based on the slowest sensor as no data interpolation is currently implemented. This limits inferencing to around 10 Hz.

With an inference rate around 10 Hz, combined with the max speed of 5 mph, allows for an inference to be made every 23 centimeters moved at max speed. This allows Little Blue plenty of time to recognize and develop a strategy to navigate around an obstacle, especially considering that Little Blue will not always be at max speed. Even with last year's model, Little Blue can currently navigate around traffic barrels on its own without any intervention. With the additional training and sensor inputs, the hope is that Little Blue can be more efficient and reliable in its movement.

10. UNIQUE SOFTWARE, SENSORS AND CONTROLS TAILORED FOR AUTO NAVIGATION OR SELF DRIVE

Little Blue is comprised of 3 main sensors to enable autonomous movement. These would be the two cameras for stereoscopic vision, the IMU to allow for velocity information to be processed, and the LiDAR for accurate ranging of all objects in front of Little Blue. These sensors are ingested into ROS for processing. The two camera images are transformed into grayscale, the top and bottom portions of the images are cropped to focus on the valuable section of the image, and then z-score normalized to enhance the edges of the image for better recognition. The LiDAR data is fed directly into the neural network as is, along with the IMU linear acceleration and integrated velocity.

Once the data is ingested by the neural network, an inference is made which provides steering and throttle input to the Raspberry Pi which communicates the information to the Arduino. The Arduino utilizes a feedback control loop to smooth out the instantaneous torque from the electric motors to ensure smooth starts and stops to prevent unnecessary jostling of Little Blue.

11. INITIAL PERFORMANCE ASSESSMENTS

During the last autonomous testing training session, the vehicle managed to successfully, albeit slowly, navigate its way through a basic traffic barrel course without exiting the road lines given. Initial testing of the software system outside of Little Blue shows that the switch to C++ was a well informed one and will improve Little Blue's efficiency.

APPENDIX

Appendix A – Parts List

Physical Components

- Sabertooth 2x60 V1.00 Motor Driver (x1)
- DC Motor (x2)
- Electromechanical Relay (x1)
- Emergency Stop Button (x1)
- Wireless Emergency Stop Transmitter (Remote Control Key Fob) (x1)
- Wireless Emergency Stop Receiver (x1)
- Raspberry Pi 4(x1)
- Arduino Due (x1)
- 12V Battery (x1)
- 24V Battery (x1)
- Converter (24V to 12V 10A) (x1)
- Nvidia Jetson Orin Nano (x1)
- Basler Ace 2 Basic (x2)
- RPLIDAR LiDAR Sensor (x1)
- 12V/48V DC Converter (x1)
- EnGenius ECS2510FP Network Switch (x1)
- U-blox ZED-F9P-02B-00 GNSS Module (x1)
- Swift Navigation GPS500 Antenna (x1)
- Basler 2200000573 50mm Lens (x2)
- TK-1560/C Lilliput 15.6”1000 nit Capacitive Monitor