AUTONOMOUS INTELLIGENT GROUND VEHICLE

by

Jonathan Campbell (EE)

Duncan Clark (CpE)

Jacob Michalick (CpE)

Victor Olabiyi (EE/CpE)

Jarub Ellenwood (EE)

Christopher Griffin (EE)

Faculty Advisor:

Dr. Lee Belfore

OLD DOMINION UNIVERSITY

Department of Electrical and Computer Engineering

Spring 2024

Approved by:

____

Dr. Lee Belfore (Advisor):

ABSTRACT

The purpose of this design project is to build and develop an intelligent autonomous vehicle that will qualify and compete at the 31st Intelligent Ground Vehicle Competition (IGVC). In this competition, the designed vehicle must complete the obstacle course within 6 minutes. During this run, the vehicle must maintain an average of at least 1 mph and must not exceed 5 mph. The emergency braking system must be operational via a push button on the vehicle and a handheld remote switch with a range of at least 100 feet. While completing the course, a judge will place an obstacle in the vehicle's path, and the vehicle must avoid it. After the course is finished, the judges will send a GPS signal to a location and the vehicle must navigate within one meter of the signal. This project aims to incorporate the emergency stop system, the controller code, more accurate and precise GPS coordination, and the neural network model which will meet the competition requirements. These systems will interface with multiple Lidar sensors, inertial measurement units (IMU), global positioning system (GPS), and stereo cameras. These sensors, systems, cameras, and units are essential for design implementation and competition qualification. With the integration of the designs for this project, the vehicle will achieve safe and secure autonomy.

Table of Contents

List of Figures

I. Introduction

I.1 Design Problem

This project's primary goal is to have a completely autonomous car that can participate in the 31st Intelligent Ground Vehicle Competition (IGVC). Eligibility in the competition requires the vehicle must adhere to the IGVC rules and regulations. The vehicle must maneuver through a 450-foot track with hazards without assistance from humans in order to complete the competition course. Developing such a vehicle requires many engineering disciplines including electrical, mechanical, and computer. Strong skills in software development are also imperative. The goal of this design team is to enhance the preexisting Little Blue robotics platform with autonomous capabilities that are in adherence to the IGVC rules.

I.2 Design Function

Overall design functionality is aimed at having a safe, reliable, autonomous vehicle. Presently there are two platforms under ownership by the team, Little Blue and Monarch. For the time allotment of this team, tackling the autonomy for Little Blue presented a much more attainable goal. The platform is currently (at the time of writing) outfitted with two 2MP UVC cameras. Utilizing a Linux-based operating system called Robot Operating System 2 (ROS2) to collect and manage data captured from the cameras that are fed into a low-cost Neural Network to operate the vehicle.

I.3 Performance Objectives

As stated previously, the overall objective is to have a fully autonomous system onboard the Little Blue platform. This shall be capable of fully controlling the vehicle at a maximum speed of 5 MPH. Furthermore, this system shall navigate and maneuver between, around, and even over obstacles in accordance with the IGVC guidelines. Exact performance specifications pertaining to various sub-systems can be found in the following section(s) below.

II. Design Approach

II.1.1 Manufacturability

Incorporating manufacturability into the design requires a deliberate balance between innovation and practical implementation. By focusing on standardization, accessible fabrication techniques, and scalability, the team focuses on developing a competitive vehicle that meets the rigorous demands of the IGVC competition, while remaining within the source limitation. The team's design not only improves the likelihood of success in the competition but also enables the vehicle to be further improved by upcoming teams. The team has been working with the MAE, as well as the current ECE 486 team in terms of how to incorporate the design for future use.

II.1.2 Sustainability

Sustainability for a robotics platform of this nature has two scopes. One is the physical aspect such as the frame, wheels, motors, etc… While the second is the software aspect. This team is particularly focused more so with the software aspect. Given that scope, generally permitting no new hardware to be added later on, or no hardware is replaced the update platform should be able to retain its usability. The main point of concern in the platform's life cycle is software updates to address given bugs, or even to make the software even more future-proof. Simply put once the platform has been updated only small updates need to be pushed to the platform from a software perspective.

II.2 Alternative Design

Our current vehicle design is for a 2-wheel drive system with a 24V and 12V battery. For an alternative design, the vehicle could use a 24V power system and an all-wheel drive. Currently, the power system uses 2 batteries which increases the footprint of the vehicle. The power system is designed to where a single 24V battery would be sufficient. This would also make troubleshooting the system more time-efficient. On top of decreasing the footprint of the vehicle, it would be better to use an all-wheel drive system. With an all-wheel drive, the weight is distributed evenly throughout the vehicle; this would be beneficial to the frame and the motors on the vehicle in the long run. All-wheel drives are also better than 2-wheel drives on uneven terrain. In competition applications, this would be beneficial since there will be ramps included in the course. All-wheel drives will have better acceleration on these ramps than a 2-wheel drive which also is another advantage for the competition given there is a minimum average speed, 1mph, requirement.

Currently, the implemented design approach for the autonomy of our platform follows the behavioral cloning model. This approach is the less common approach for designing an autonomous platform. As it replicates the behavior of the driver, therefore, never allowing the platform to be better than the driver itself. The alternative design more commonly used is Simultaneous Localization and Mapping, or SLAM for short. With this approach, this would require a stack of robust sensors and therefore more extensive processing capabilities to effectively maneuver the vehicle. Efforts were previously made with this approach on previous iterations of the Little Blue platform. This alternative design is better suited for a vast suite of autonomous platforms rather than a singular solution. This uses data from its sensor stack such as LiDAR, Radar, Stereo Cameras, GPS, and more to map any vehicle's surroundings. While this does achieve the desired results, behavioral cloning allows for a more straightforward and cost-effective solution.

The vehicle used for this project does not have any mechanical brakes such as a drum or disc brake. However, for future designs, the mechanical braking system was integrated into the vehicle. This approach alters how the emergency braking system will be designed. Therefore, an alternative design for a mechanical emergency stop system is to use a linear solenoid which will be activated upon a single pulse from a capacitor bank. Once the handheld receiver or pushbutton is activated, power then flows through the voltage regulator to give a smooth, excitation signal to the solenoid. Once activated the solenoid will release a holding pin that allows the main spring to release its tension. As the main spring is released, the mechanical brakes are engaged. The figures below depict this design.

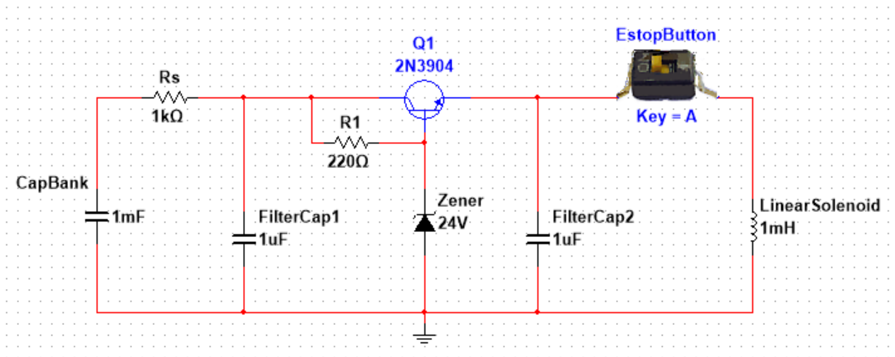

Figure 1: Emergency Stop Voltage Regulator Circuit

For our vehicle, the PID controller is executed through voltage sensing. For future models, an alternative design would be to use a Hall effect sensor. Encoders add to the cost of the design and they have to be properly integrated onto the wheels for accurate readings. On the other hand, there are motor controllers that incorporate Hall effect sensors into the drive. These sensors apply a magnetic field to the current-carrying conductors which gives an output voltage. This is the feedback signal that is used for the PID controller which is sent to the external PWM signals on the motor driver. This approach is more cost-effective and it optimizes the footprint of the vehicle.

III. Project deliverables

III.1 Project Management

Project organization has remained constant and consistent throughout the year of development of this undergraduate project. Independent team meetings have stayed at a constant frequency of two meetings weekly, one with and without Dr. Belfore. Given the formation of a sister team in the Mechanical Engineering department, bi-weekly meetings are also occurring to keep the entire IGVC group in the same bearing.

IV. Design Specifications

IV.1 Details of the Engineering Design

This project utilizes a robotics platform outfitted with three primary subsystems: sensor input, actuator output, and compute. Further, the platform operates in two modes: autonomous and human-controlled. As for human control, a simple wireless controller is connected to the vehicle via Bluetooth. Interfacing with a Raspberry Pi, a ROS2 node is used to read controller input that then serially outputs data to an embedded Arduino. This serial communication controls the required PWM signals needed to operate the motor controller. Descriptions for the autonomous operations are detailed below.

IV.1.1 Power and Emergency Stop System Design

As stated, the Little Blue platform obtained this semester was passed from a previous project team. To start the design of the current power system, the main components of the previous power system were tested and integrated into the system; this includes the 24V and 12V Lithium Iron Phosphate batteries, the Sabertooth 2x60 Motor Driver, the wireless transmitter and receiver, and the 12V to 48V DC-DC Converter. As project development progressed, the Arduino, Raspberry Pi, and KV260 FPGA were integrated into the system for motor control, remote control, and autonomous mode operation. The network switch was integrated for power over ethernet for the Raspberry Pi; this approach decreased the footprint of the power system.

Once the power system was designed, it was imperative that the emergency braking system was incorporated into the design. Since this vehicle does not have a mechanical brake onboard, the quickest way to stop the vehicle is to disconnect the power to the motors. As seen below, the 24V source is connected to both the normally closed (NC) push button and the normally open (NO) wireless switch. This design approach always complies with the IGVC and is power efficient. The 24V battery powers the motor driver via the NC connection on the relay, and when a situation occurs when the E-stop is needed, the push button will disconnect the power to the motor or the wireless switch will switch the power flow from the motor to the open contact. How the wireless switch approach works is when activated it will send a positive signal to the receiver onboard the vehicle which will output a positive 24V signal that triggers the coil inside the relay which flips the internal switch of the relay to the NO position.

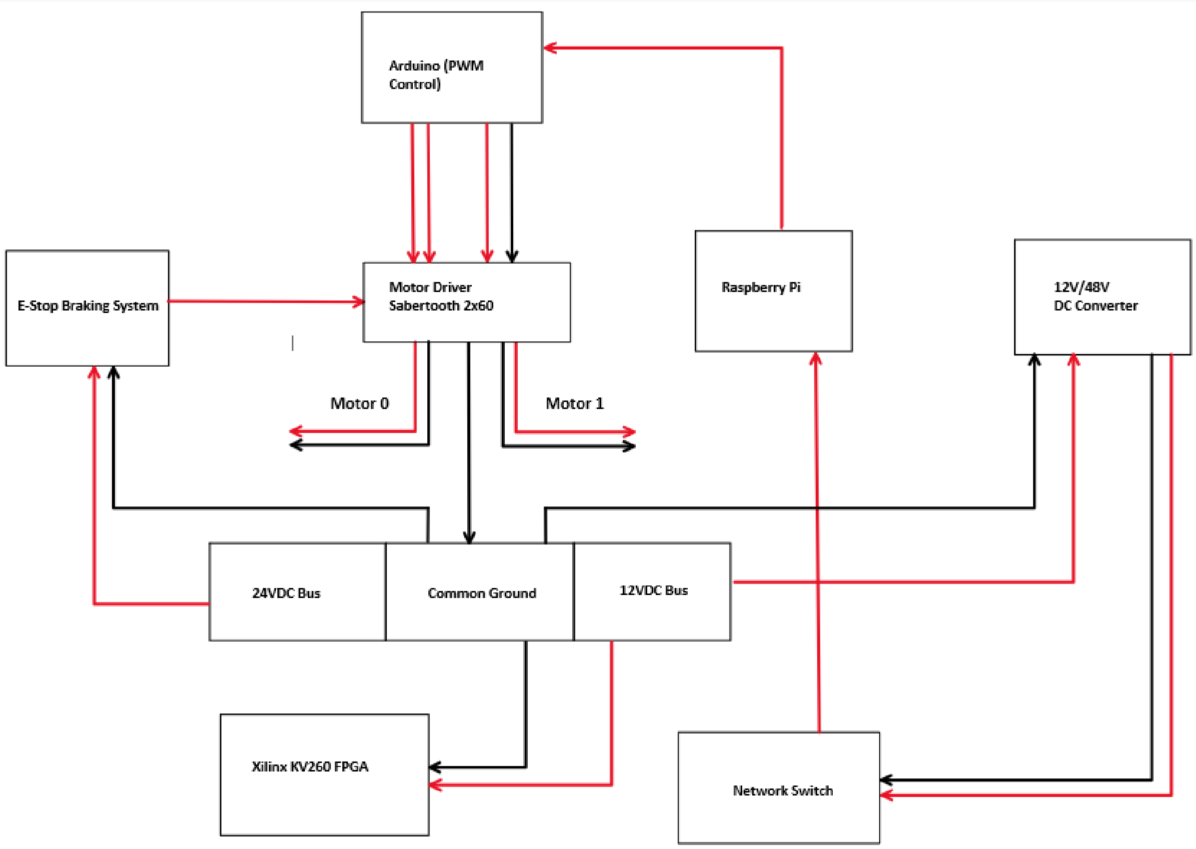

Figure 2: Power System Design

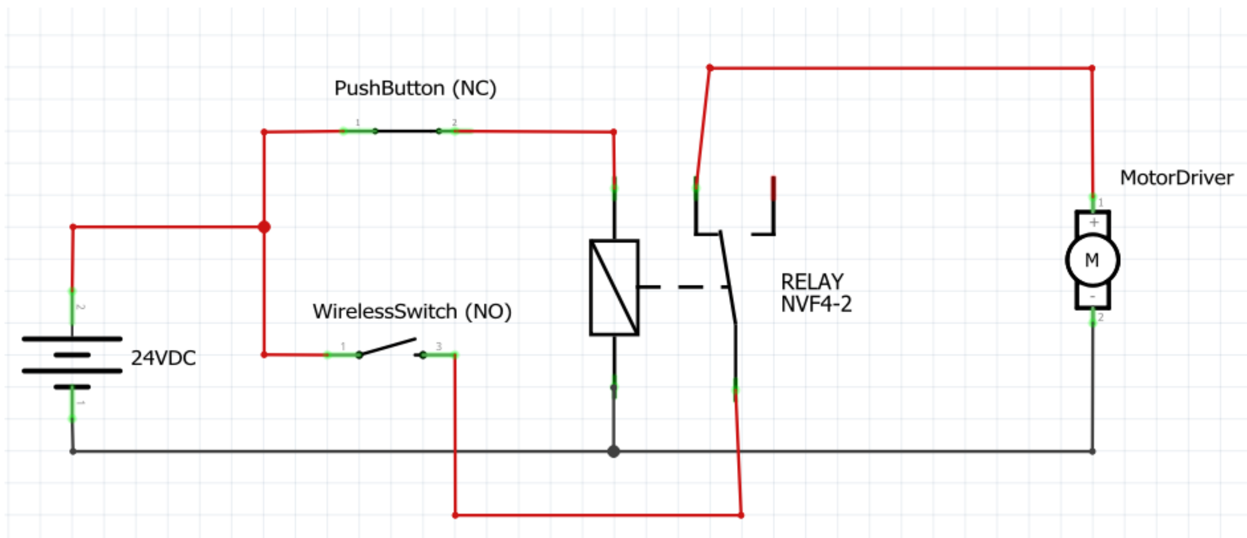

Figure 3: E-Stop Circuit Design

IV.1.1 Neural Network

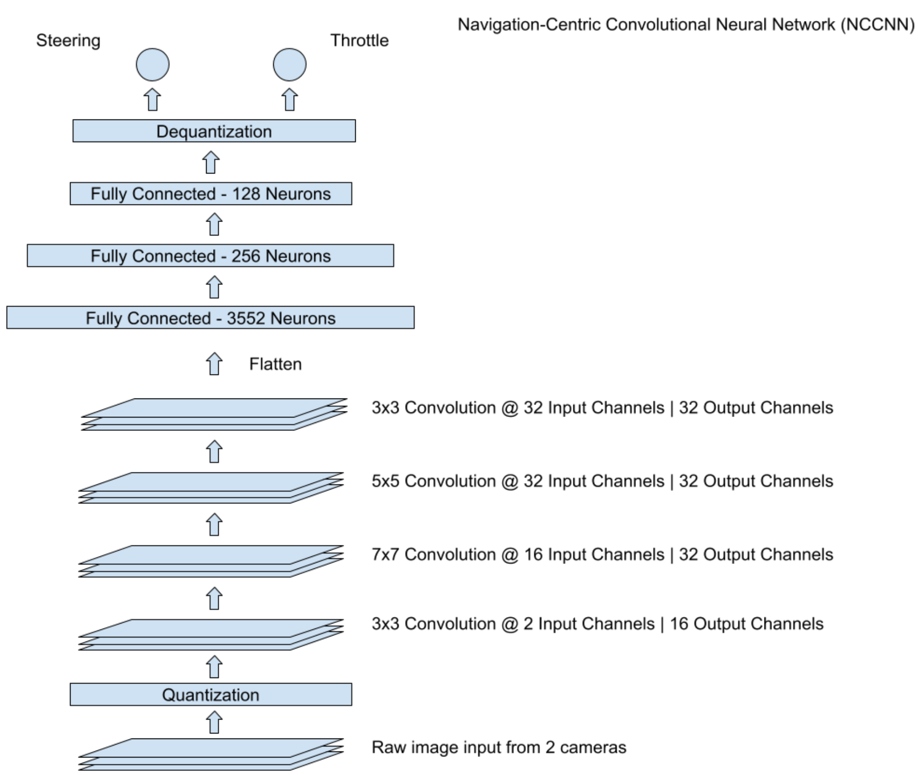

Figure 4: Navigation-Centric Convolutional Neural Network (NCCNN), first network design

Autonomy is achieved by way of a machine-learning technique called Behavioral Cloning. This is a derivation of Imitation Learning and focuses primarily on mapping the actions of a known expert to the sensor input. With this technique, we teach a neural network to mimic the behavior of a human driver.

Our current neural network design exists solely as a convolutional neural network. This system is called the Navigation-Centric Convolutional Neural Network. Its purpose is to maintain a course within a given track. In our case, this track is bound by lane lines.

The next revision of our neural network will see the addition of LiDAR input data. Lidar is exceptional at creating high-resolution maps of the local environment. This will be important for implementing obstacle detection. The idea is to structure the network in such a way that the relationship between the lidar and camera data is retained, and this data collectively corresponds to the actuator outputs. We expect the result to be an autonomous system we call the Joint Navigation and Obstacle Avoidance Neural Network.

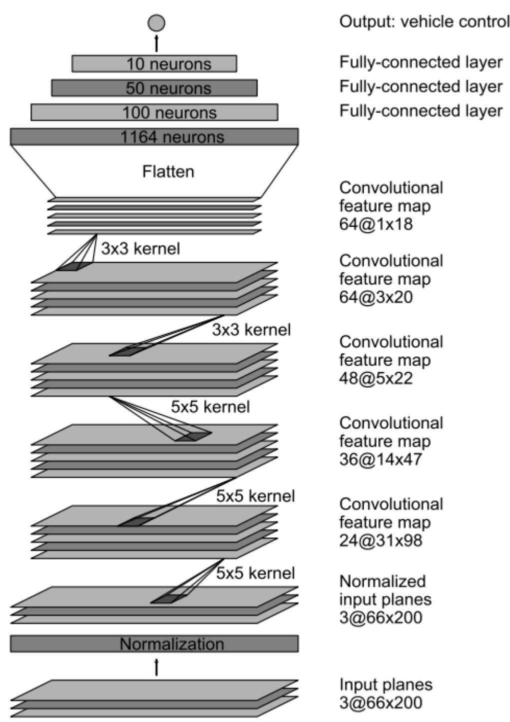

Figure 5: NVIDIA's Neural Network

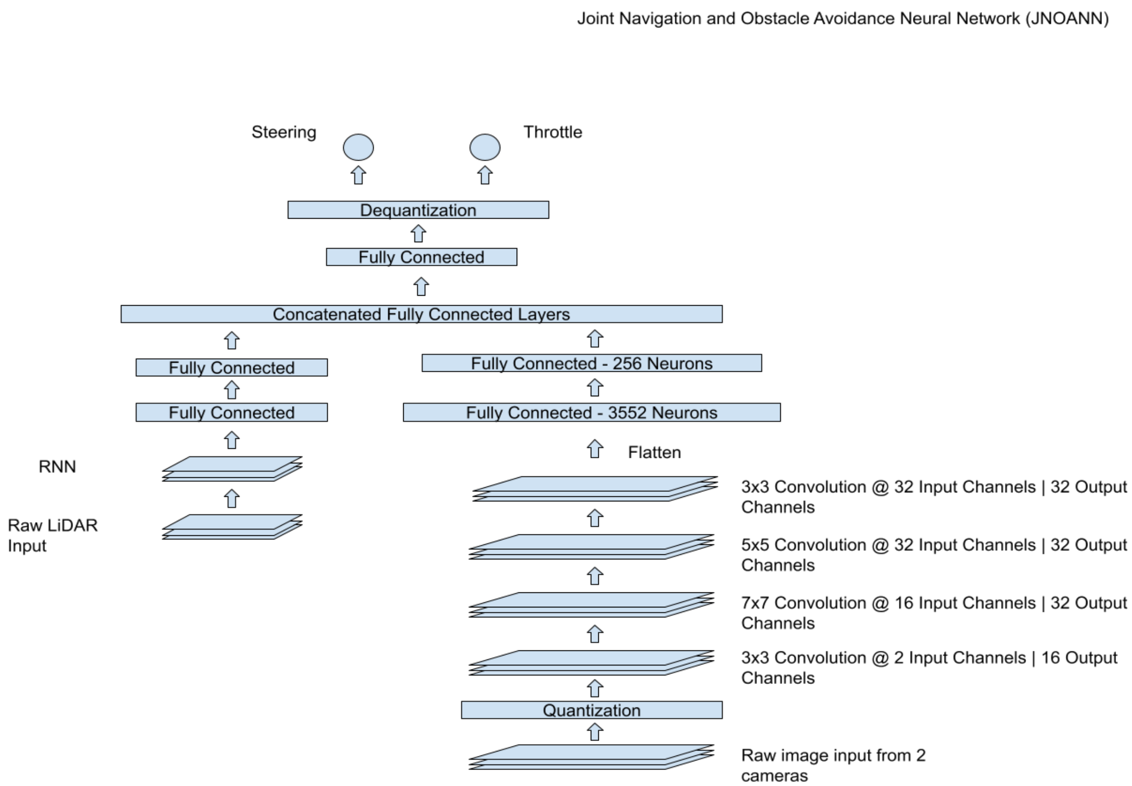

Figure 6: Joint Navigation and Obstacle Avoidance Neural Network (JNOANN)

IV.1.2 Quantization Aware Training

Quantization Aware Training (QAT) models the effects of quantization during training allowing for higher accuracy compared to other quantization methods [5]. Quantization in machine learning is a technique for storing tensors at lower bit-widths than floating point precision [5]. PyTorch supports int8 quantization, enabling a theoretical 4x increase in performance and 4x reduction in memory bandwidth when compared to fixed point 32. In practice, performance gains much higher than this were observed.

Inferencing was roughly 1.8 Hz prior to QAT implementation. This proved to be too slow for the vehicle to make decisions, thus a performance increase was required. A normalized performance test was created to measure against. This test ran an inference loop on a set size of data. The design could then be altered to improve performance.

IV.1.3 Motor Control Equations

$$a = Throttle \mid -1.0 : 1.0$$

$$b = Steering \mid -1.0 : 1.0$$

$$M_0 = Motor\ 0 \mid 0.0 : 1.0$$

$$M_1 = Motor\ 1 \mid 0.0 : 1.0$$

$$M_0 = -\frac{ab}{4}(b + \sqrt{b^2}) + \frac{a+1}{2}$$

Equation 1: Motor 0 control equation

$$M_1 = -\frac{ab}{4}(b - \sqrt{b^2}) + \frac{a+1}{2}$$

Equation 2: Motor 1 control equation

IV.1.4 Custom ROS2 Message Type

To reliably transmit data throughout the system, a custom ROS2 datatype called “Data” was required. It consisted of the following elements:

<Message Type> <Name>

sensor_msgs/Joy gamepad

sensor_msgs/Image cam0

sensor_msgs/Image cam1

bool data_active

bool auto_active

float32 steering

float32 throttle

With this custom message type, each element must be populated to transmit via ROS2. In doing so, the synchronization of the elements is more reliable.

Data synchronization is essential in maintaining dataset integrity. This is because the Neural Net requires the incoming data samples to be for a given moment in time. If the data samples were taken sequentially, they would be out of sync by the amount of time it took to take the previous data sample.

To combat this, ROS2 is leveraged to make parallel data collection easy to implement. The ROS2 framework provides all the tools required to make this possible.

IV.1.5 Data Buffering for Improved Performance

Given the lack of performance in the KV260 Processing System (PS), it initially failed to reach a sufficient sampling rate for data collection. This was in large part due to the latency in writing the data to disk storage. To overcome the performance limitations, data is cached prior to the write sequence.

As data is collected, it is stored in a queue, chosen for the purpose of maintaining order in our dataset with a FIFO strategy. Once the queue is filled, data collection is halted and the queue is dumped into storage. This approach trades a higher latency in data writing for a higher sampling rate, since the overhead of formatting and writing the data is postponed to a dedicated write cycle. This trade-off is further justified as real-time data formatting is not considered essential.

A more significant trade-off occurs when key events are experienced during the write cycle. Given the length of time required to complete, it's not uncommon for key events to occur while the write cycle is executing. Given that the write cycle halts data collection, this means these key events are ultimately missed. This is addressed with our data collection strategy.

It's understood that a majority of data should be collected while the vehicle turns to avoid a bias towards driving straight [5]. This is advantageous as long straight sections provide a good opportunity to dump data. Given this, the tradeoff is rarely experienced in practice.

IV.1.6 GPS Base Station

Global Positioning Systems (GPS) are satellite-based navigation systems that were first developed by the United States Department of Defense for military use [6]. Its applications and use cases have continued to grow since then, ranging from civilian to the commercial sectors. In the commercial sector, there are still many different forms of GPS modifications that are used, such as Assisted-GPS (AGPS), Simultaneous-GPS (SGPS), Differential-GPS (DGPS), and Non-Differential GPS to name a few [6]. Each type comes with its pros and cons depending on the particular use case for it and what degree of precision is necessary. The accuracy of most geographic locations provided by GPS servers range from 1 to 100 meters with military-grade devices improving this accuracy to less than 1 meter [6]. The accuracy of the GPS coordinates received by a receiver depends on the specific use case.

In the commercial sector, specifically cellular companies and shipping services like Amazon, use very imprecise GPS systems which are typically accurate within 4.9 meters of the actual position [7]. This is due to the fact that for many of these providers, this level of accuracy is enough to provide the necessary information to the user. For example, when using Google Maps to navigate to a location, the mobile device being used uses the cellular network's GPS system to locate the position of the vehicle [6]. Since it is inferred that the vehicle will most likely be on a paved road while driving and these roads are known, the estimated location of the vehicle can be determined within a decent margin of error as long as the general location and direction are provided well enough to get to a specific destination, there is no need for a higher percentage of accuracy [6]. However, some systems do need more accuracy in order to provide better navigation to specific waypoints, such as this project. The good news is that systems do exist that enable the user to do just that.

Differential-GPS is one of these types of very accurate types of coordinate systems that allow the user to get an accuracy down to the range of centimeters (cm) [6]. The main purpose of this type of GPS is to increase the overall accuracy of positioning coordinates. It works by using a single or networked system of fixed-ground stations that broadcast the calculated difference between the known fixed point and the satellites' locations [6]. This is incredibly important for this project because the IGVC requires the autonomous vehicle to navigate to specific GPS coordinates in a very small and confined space. Implementing such a system will allow the team to meet these requirements with a high degree of accuracy which could be the difference between winning and losing in the competition. To achieve this degree of accuracy, the team will be implementing a device called a GPS base station which will remain in one location on the ODU campus.

The goal of the GPS base station system is to provide the autonomous vehicle with very precise and accurate coordinate information that cannot be achieved with a standard off-the-shelf consumer-grade GPS device, as discussed previously. These devices can have very poor coordinate positions since they are mainly used to navigate using paved roads and thus can get close enough to the correct coordinates and determine where the vehicle position is on the road. However, for the IGVC, there are precise coordinate locations that the autonomous vehicle must drive to that are not on paved roads. The standard GPS accuracy of 10 meters far exceeds the allowable limit of error when the project goes to competition and would likely fail to meet the guideline requirements. To overcome this, a previous team that worked on the project in the past developed a stationary GPS base station system that utilized an antenna connected to a UBLOX ZED-F9P GPS Surveyor (depicted in Figure 11) receiver via coax, which then sent the data to a Raspberry Pi where it could be compiled and processed to send out corrective data wirelessly and with a ten-kilometer range to another receiver mounted to Little Blue. This design would, theoretically, allow for an improved coordinate accuracy of one centimeter. Based on the documentation provided by the previous team, they were successfully able to get the device to work, however, misunderstood the design requirements, and ultimately were unable to implement the base station into the design. Therefore, the team picked up where they left off and finalized the design to complete the implementation process.

Figure 7: ZED-F9P GPS Surveyor Receiver

IV.2 Engineering Standards

This section depicts the standards followed during the development of the autonomous ground vehicle. Let it be noted that additional standards may be appended by the end of the project planning phase.

ISO 26262 Road Vehicles Functional Safety[2]**.** This talks about the safety-related systems that include one or more electrical or electronic systems that are installed in road vehicles, which relates to the Autonomous Ground vehicle the team is working on. This standard pertains to the functional safety of electrical and electronic systems within road vehicles.

IEEE 1164 Standard Multivalue Logic System for VHDL Model Interoperability [1]. This standard is consolidated into a package that, "provides a standard datatype system for the declaration of ports and signals in VHDL"[1]. Given our project will pertain to hardware acceleration of artificial intelligence for autonomous ground vehicles, this standard will be required. Furthermore, development of this hardware accelerated model will be done in VHDL providing more necessity for this standard.

IEEE Standard for Robot Map Data Representation for Navigation [3**]**. This standard outlines the requirements for generating mapping data for use in autonomous driving robots. It involves the use of SLAM mapping, guidance, and control. This is important for our project because we intend to use SLAM within our autonomous navigation stack.

ISO 13850 Safety of Machinery - Emergency Stop Function [4]**.** This standard defines the emergency stop function as, "intended to avert arising or reducing existing hazards to persons, damage to machinery or to work in progress and be initiated by a single human action" [4]. This standard did limit the reach of its scope to handheld push buttons. While the autonomous vehicle will have a handheld button, the push button installed on the vehicle will require the use of this standard. This standard will specify the necessary functions of an emergency stop, and it will add context to situations where the emergency button needs to be activated.

V. Design Performance

V.1 Quantization Aware Training Results

Baseline Test Results: 6.98 seconds @ 0.0302 Average Loss

QAT Test Results: 0.62 seconds @ 0.0575 Average Loss

The result of this change yielded a 10x performance increase in our benchmark test. The tradeoff in turn is a lower accuracy for the model. We hope to combat this with a better dataset and longer training cycles. With these changes, we don't expect any potential loss in accuracy to translate to real-world operation.

V.2 Results of Buffered Data Collection

Original Max Sample Rate: 1.6 Hz

Original Write Latency: 0.6 seconds

Cached Data Max Frequency (tested): 10 Hz

Cached Data Write Latency: 6.6 seconds

V.3 Network Optimization

When training our first network design, it was surprising to find the size of the weights file to surpass 2.5GB. This immediately necessitated an improved design to reduce the size. Refining the design resulted in an optimized weights file of 1MB in size.

V.4 GPS Base Station Results

The team has been in the process of implementing the base station this semester and has had some compelling results. Using the SparkFun RTK Surveyor in conjunction with a TOP106 GNSS antenna mount on top of a Bosch BT160 surveying tripod, we collected 24 hours of GPS coordinate data that was saved in an observation file format (.OBS). This file was then sent to a third-party company for analysis to return accurate GPS coordinates of the antenna with an estimated error in the millimeter range. Once we received the precise coordinates from the third-party service, we were able to set the RTK Surveyor to “Fixed” mode by inputting the precise coordinates into the command user interface (CUI) of the RTKLIB. Once that was complete, we used a Raspberry Pi to connect to an NTRIP casting server over the internet to cast the differential GPS coordinates to rtk2go.com under the Mount Point name ODUIGVC. This will allow anyone with an internet connection to connect to the NTRIP caster as a client to receive the GPS correction data. This is where the ZED-F9P u-blox GPS module and Tallysman TW4721 antenna come into play. These devices were mounted to the vehicle and connected to a Raspberry Pi that was already on board. Using the same RTKLIB library on the Raspberry Pi, we can connect to the ODUIGVC NTRIP caster as a client and receive the correction data. During our driving tests, we were able to obtain an estimated coordinate accuracy of 1 to 3 centimeters in “Fixed” mode.

VII. Summary

In all, the power system design has performed well. The equations derived for the motors were essential in the control code for the vehicle. Most importantly, the behavioral cloning model was successful. As we drove the car around the test environment, the data was able to be collected in training the model. After a few laps around the track, the vehicle was tested in autonomous mode, and it was able to recognize the turns and begin executing. This was the main proof of concept desired for the project timeline. As we continue the project, we plan to integrate the power consumption system and design safety features for the vehicle, and we intend to collect more data to fully train the behavioral cloning model.

References

[1] "IEEE Standard Multivalue Logic System for VHDL Model Interoperability (Std_logic_1164)," in IEEE Std 1164-1993, vol., no., pp.1-24, 26 May 1993, doi: 10.1109/IEEESTD.1993.115571.

[2] "ISO 26262-1:2018." ISO. (2023, October 15). https://www.iso.org/standard/68383.html

[3] W. Yu and F. Amigoni, "IEEE standard for Robot MAP data representation for navigation," IEEE Standards Association, 2015. doi:10.1109/ieeestd.2015.7300355

[4] "ISO 13850:2015 Safety of Machinery Emergency stop function Principles for design," ISO 13850:2015, https://www.iso.org/standard/59970.html (accessed Oct. 19, 2023).

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to End Learning for Self-Driving Cars," arXiv preprint arXiv:1604.07316, 2016.

[6] C. BasuMallick, "What Is GPS (Global Positioning System)? Meaning, Types, Working, Examples, and Applications," spiceworks, 3 June 2022. [Online]. Available: https://www.spiceworks.com/tech/iot/articles/what-is-gps/. [Accessed 17 April 2024].

[7] National Coordination Office for Space-Based Positioning, Navigation, and Timing, "GPS Accuracy," U.S. Space Force, March 2022. [Online]. Available: https://www.gps.gov/systems/gps/performance/accuracy/. [Accessed 17 April 2024].