Paradigm Engineering

F.R.O.G.

May 15th, 2022
29th Intelligent Ground Vehicle Competition
Memorial University of Newfoundland & Labrador
and College of the North Atlantic

Paradigm

Gavin Burden (Team Lead)
gavin@paradigmhyperloop.com

Aidan Clark
amhclark@mun.ca
Colton Smith
cnsmith@mun.ca
Ian Carroll
icarroll@mun.ca
Patrick Cleary
pcleary@mun.ca

Andrew Nash
anash@mun.ca
Dawson Mercer
dwem52@mun.ca
Jumanah Babar
jbabar@mun.ca
Ryan Carey
ryan.carey52@ed.cna.nl.ca

Cole Matthews
cmatthews20@mun.ca
Francis Walsh
fkw155@mun.ca
Mohammad Emad
memad@mun.ca
Tinashe Goora
togoora@mun.ca

Dr. Oscar De Silva (Faculty Advisor)

I certify that the design and engineering of the vehicle by the current student team has been significant and equivalent to what might be awarded credit in a senior design course.

17 May 2022

Table of Contents

1 Introduction

Paradigm Engineering is happy to be representing Memorial University of Newfoundland & Labrador (MUNL) and the College of the North Atlantic (CNA) at the 29th annual Intelligent Ground Vehicle Competition. Paradigm has a standing history of design competitions having competed historically in the SpaceX Hyperloop Competitions and the first Not-a-Boring competition. F.R.O.G. is Paradigm's competition vehicle that runs on three independently powered omnidirectional wheels to allow a high degree of maneuverability. A Stereolabs ZED 2 Camera is utilized for the stereo vision system that F.R.O.G. relies on to navigate the course.

2 Team Organization

Paradigm's team structure is largely flat with subdiscipline leads aided by a team lead. The team is broken up into four sub teams: Business/Marketing, Electrical, Mechanical, and Software. Table 2-1 shows a complete roster of the design team and respective departments.

Table 2-1 - Technical Team Members and Departments.

Aidan Clark	MUNL Mechanical Engineering
Andrew Nash	MUNL Computer Science (Graduate)
Cole Matthews	MUNL Computer Engineering
Colton Smith	MUNL Computer Engineering
Dawson Mercer	CNA Software Development
Francis Walsh	MUNL Electrical Engineering
Ian Carroll	MUNL Computer Engineering
Jumanah Babar	MUNL Electrical Engineering
Mohammad Emad	MUNL Mechanical Engineering
Patrick Cleary	MUNL Mechanical Engineering
Ryan Carey	CNA Mechanical Engineering Technology
Tinashe Goora	MUNL Mechanical Engineering

A combined total of approximately 2000+ man hrs have gone into the vehicle development at the time of submission on this report. Total vehicle component cost is approximately $8000 CAD.

3 Design Assumptions and Process

The criteria for design outlined in the rules document was the large driving force behind design decisions. The information outlined in the rules was extracted and a set of criteria was established to direct the design efforts. The criteria were used to assess concepts brainstormed by the team, drawing from existing solutions and novel ideas to establish the final design. All design work was peer reviewed by fellow team members and checked against the design criteria.

4 Innovations in Vehicle Design

A key innovation in the vehicle design is the holonomic drivetrain system, made of a three omni wheel setup, designed to have complete freedom of movement. The level of dexterity afforded by the omni wheel topology will allow F.R.O.G. to maneuver through the course efficiently. The vehicle's navigation software utilizes state-of-the-art machine learning research from recent publications, for semantic segmentation and bird's eye view mapping of the environment.

Another key innovation/improvement that the team made (when considering past performance in other student engineering competitions) was increasing the simplicity of the vehicle's electrical architecture. Since there is only one custom circuit board that houses all components, connectors and sensor interfaces, the wiring harness design was simplified since there are less points of connections. Moreover, the connectors on the custom circuit board were selected to be the exact same as the downstream connectors they interface with, thus, OTS pre-crimped, 1:1 wire harnesses could be used - increasing reliability and ease of manufacturing.

5 Mechanical System

This section outlines the mechanical systems design and addresses the validation methods employed to ensure the vehicle will perform as anticipated.

5.1 Design Overview

Table 5-1 - Vehicle Specifications.

Total Vehicle Mass	80 kg
Vehicle Height	1130 mm
Vehicle Width/Length	880 mm
Minimum Ground Clearance	69 mm

5.2 Drivetrain

All three VEX Robotics omni wheels are directly driven from independent motors. The motor is keyed into a custom driveshaft fitted into the OTS omni wheel hubs. Each shaft is supported by double row angular contact bearings in custom mounts. Figure 5-1 shows a cross section of a wheel module with description of sub components in Table 5-2. Each omni wheel is able to transmit power in its tangential axis while sliding freely along the axial axis. The motors were selected based on an assumption of full speed on the 15% grade hill to ensure adequate power capabilities. Motor speed is limited through the ESC firmware.

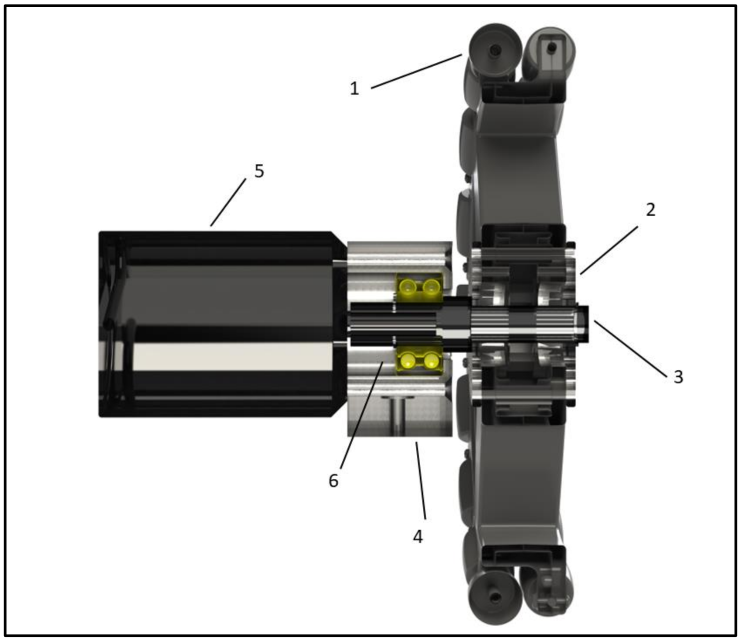


Figure 5-1 - Cross-section schematic of the wheel module.

Table 5-2 - Wheel Module part list.

Number	Part
1	Omni-Wheel
2	Wheel Hub
3	Drive Shaft
4	Motor Mount/Bearing Housing
5	Flipsky Motor
6	Double Row Angular Contact Bearing

The critical shaft load case consists of three primary loads: motor torque, normal force due to vehicle weight and an axial reaction force due to forward acceleration. The shaft in made from 1045 Steel achieving a factor of safety above 10. A double row angular contact bearing supports the driveshaft and prevents loads from being transmitted to the motor. The motor mount is a dual-purpose component, it secures the motors and seats the angular contact bearing.

5.3 Chassis

The vehicle chassis is designed to house and protect the electronics, battery pack and payload. The chassis design ensures ease of access for maintenance while also providing a multitude of layout options for the onboard electronics. The chassis is composed of three-levels, each constructed from ⅜" thick Aluminum 6061 each connected with stand-offs in the 6 corners. Figure 5-2 shows the vehicle chassis.

Figure 5-2 - Vehicle Chassis.

To ensure that the chassis was suitable to handle the loads inflicted by the robotic components - SolidWorks Finite Element Analysis was used to simulate the loading that is experienced by the fully assembled vehicle. Suspension was deemed to be unnecessary given the consistent terrain provided by the asphalt and was eliminated to avoid unjustified system complexity. The external shell is comprised of 16-Gauge 6061 Aluminum to protect the electronics and provides the space for sponsoring logos to be displayed. Specific panels can be removed to access hardware within the vehicle.

5.4 Weatherproofing

Both the Control Laptop and Node board are housed in weatherproof enclosures to protect from water ingress. The motors and ESC board are weather rated and all connections to these will be sealed from water ingress. The design of the chassis itself affords protection from weather to the electronics housed inside the vehicle.

6 Electrical and Power Systems

6.1 Design Overview

The vehicle's electrical system contains two distinct subsystems: compute and control. The compute portion of the electrical system is responsible for navigation and high-level state control. The control section is responsible for controlling/monitoring the motors and battery system. A high-level diagram displaying the vehicle architecture can be seen below in Figure 6-1.

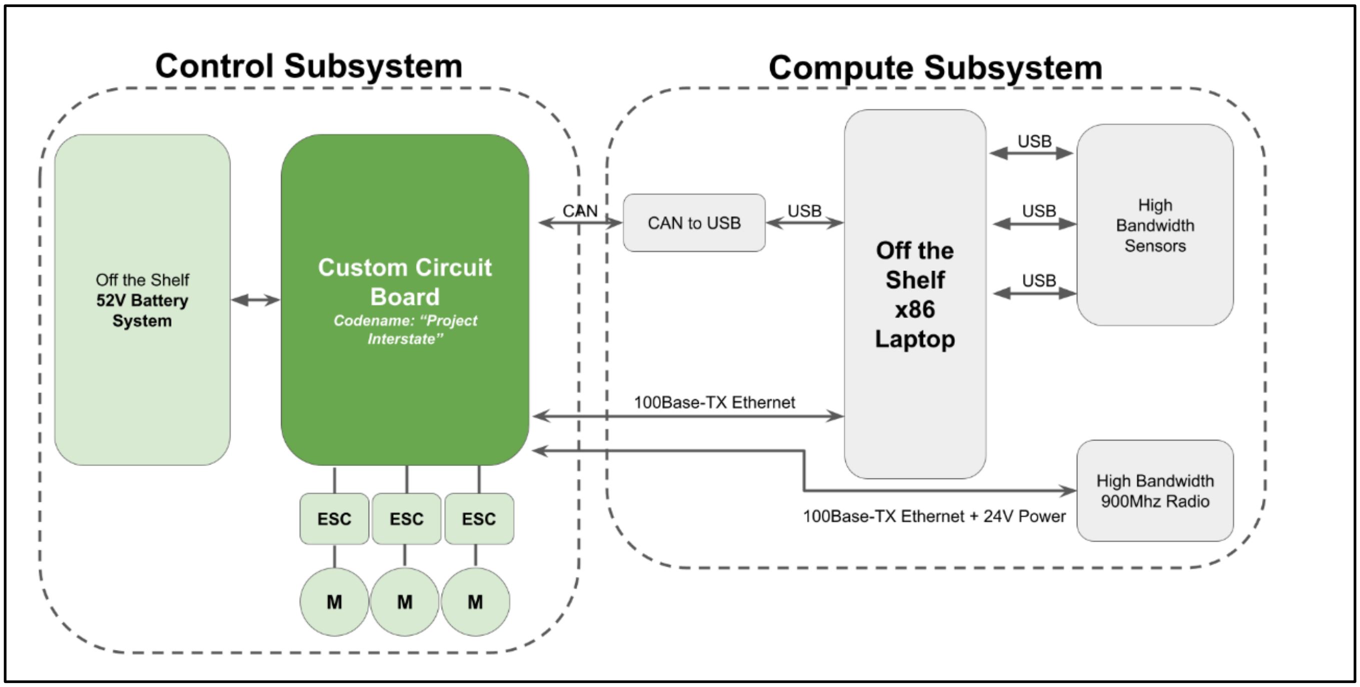


Figure 6-1 - Schematic of Vehicle Electrical System Architecture.

6.2 Control Subsystem

The control subsystem consists of:

- 3x OTS Brushless Direct Current (BLDC) motors
- 3x OTS BLDC motor controllers (or Electronic Speed Controllers (ESCs))
- 1x OTS Lithium-ion e-bike battery and battery management system
- 1x custom circuit board designed by Paradigm's electrical team
- Custom wire harnessing

The lithium battery chemistry was selected over others (such as lead acid, nickel cadmium, etc.) due to its high energy/power density. The reduced power/energy to weight ratio reduced constraints for the mechanical/structural design of the vehicle. An OTS battery system was selected since the team did not have the resources to design and fabricate a lithium-ion battery, battery management system and charging system. Table 6-1 Shows the OTS e-bike battery specifications.

Table 6-1 - E-bike Battery Specifications

Specification	Value	Unit
Battery Chemistry	Lithium Ion	N/A
Working Voltage	36 - 54.6	Volts
Charge Time	4-6	Hours
Battery Capacity	20	Ah
BMS Overcurrent	30	Amps

Given that the vehicle's nominal power draw (while driving) is expected to be around 300 W, the 20 Ah rating of the battery will provide a 3.5 hour run time for the vehicle (and even longer when in standby mode).

The custom controls circuit Node board was designed to perform the functions listed below:

- Provide an interface for the compute subsystem to the control subsystem (I.E control the motors)
- Distribute 52V power
- Regulate 52V power to other voltage rails
- House circuitry to perform mandatory IGVC safety functions (I.E: wireless/wired e-stop, safety lights etc.)
- Act as a single point for signals/wiring, to simplify harness design
- Provide Power over Ethernet (PoE) injection for the high bandwidth 900MHz radio

A render of the board can be seen in Figure 6-2. The Node board schematic can be seen in Appendix A.

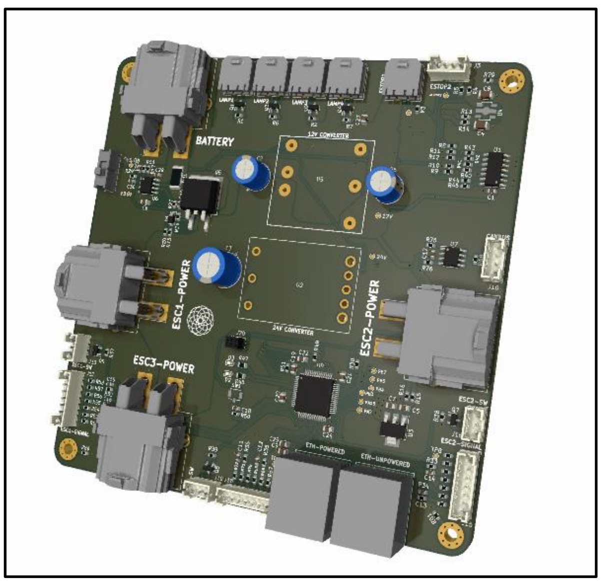


Figure 6-2 - Custom Node Board.

Power that enters the custom circuit board meets a hot-swap controller to prevent contact arcing and brown-out from inrush currents. This 52 V power is then sent directly to the ESCs and switching/linear power converters. 1:1 wiring harnesses are used to interface the ESCs with the custom circuit board.

6.3 Compute Subsystem

Compute subsystem hardware was selected based on availability and compute requirements. The Control Laptop is an x86 platform with a 6-core Intel Core i7-10875H CPU. Machine learning computations are offloaded onto an Nvidia RTX 3070 external GPU (eGPU). Preliminary testing has shown that this CPU and GPU combination provides sufficient compute for the navigation calculations and machine learning pipeline. Alternative AI-oriented embedded platforms were explored, such as Nvidia Jetson and Google Coral. Based on cost, availability, and simplicity, an x86 laptop and desktop GPU were selected.

The vehicle's vision system utilizes a Stereolabs ZED 2 stereo camera, which is commonly employed in autonomous robotics applications. The ZED 2 has an integrated inertial navigation system for positional tracking and a robust software development kit facilitating rapid prototyping. An ardusimple RTK GPS is used in conjunction with the ZED 2 for localization and waypoint-based pathfinding.

6.4 Safety Devices

The mechanical e-stop is a momentary, normally open, push button that's located on the center rear of the vehicle. The button interfaces with the custom circuit board and produces an active high signal when pressed.

The wireless e-stop uses two ESP32 LoRa OLED development boards and two U.FL IPX to SMA Antenna in the 915 MHz frequency range. The wireless e-stop is controlled from a handheld remote-control transmitter containing one of the developmental boards and an antenna. The transmitter remote has a push button which when pressed transmits a signal to the receiver. The 915 MHz receiver is interfaced with the custom circuit board and generates an active high signal when an e-stop request is sent.

The e-stop circuits were designed to be completely analog, reducing complexity that is incurred with firmware. The full circuit can be seen in Figure 6-3. The 915 MHz wireless e-stop module and physical e-stop both generate active high signals. These signals are interfaced to open-drain output comparators (the LM359 from Texas Instruments). These comparators are connected in a logical OR configuration. When an e-stop signal goes high, the 'ON' pin of the hot swap controller (the LT1641-1 from Analog devices) is 'servoed' to ground (normally this pin is just used to detect a low voltage scenario). A sliding toggle switch was also interfaced in the logical OR configuration to also 'servo' the ON pin to ground.

When the ON pin is lower than 1.233 V on the hot swap controller, the gate to the bootstrapped NMOS pass transistor is turned off, turning off 52 V power to the ESCs, power converters and subsequently, the onboard microcontroller. When the e-stop signals disappear the NMOS pass transistor is turned back on allowing the system to boot back to a safe state (See Appendix B for Control Subsystem State Machine).

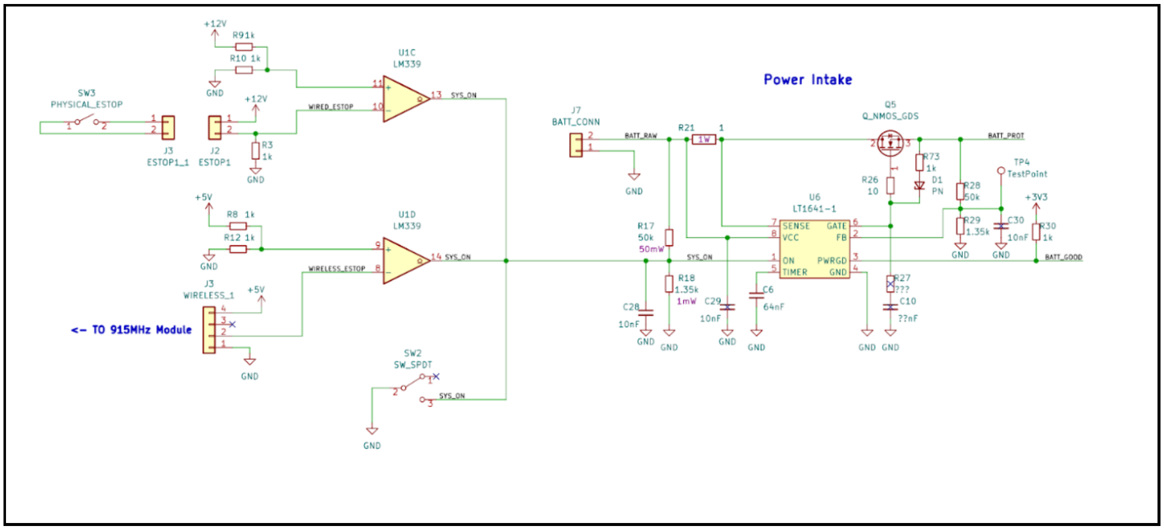


Figure 6-3 - Node board E-stop circuitry.

6.5 Safety Lights

The vehicle has solid indicator red LED surface mount safety lights which are connected to the custom circuit board and are turned on whenever the vehicle power is turned on. They go from solid to flashing when the vehicle is in autonomous mode and back to solid when it comes out of autonomous mode. These lights are toggled by the microcontroller onboard the custom controls circuit board via low side NMOS transistor switches.

7 Software Strategy and Mapping Techniques

7.1 Design Overview

Utilizing state-of-the-art machine learning approaches, the software team designed a perception system that relies solely upon a camera to segment and map the environment around it. Detailed below is a full overview of the software navigation pipeline and architecture.

7.2 Obstacle Detection and Avoidance

Since a vision system is required to detect obstacles, lanes and (simulated) potholes, the team decided to solely rely upon it as the only perception sensor; avoiding additional complexity and uncertainty associated with sensor fusion approaches. The detection system follows a simple two step machine learning pipeline: first an image from the vehicle's camera is semantically segmented; then this image is converted into a bird's eye (top down) view of the environment. The detection system utilizes E. Xie *et al*'s [1] SegFormer model for semantic segmentation of images and the model proposed in Reiher *et al*'s [2] uNetXST to obtain the bird's eye view. These machine learning models were selected because of their state-of-the-art performance on the CityScapes dataset [3], but are subject to change as additional models are benchmarked on simulator testing data. The team obtains training data for machine learning models from CARLA based simulation, Figure 7-1 demonstrates the extraction of ground truth labels for semantic segmentation from the simulator.

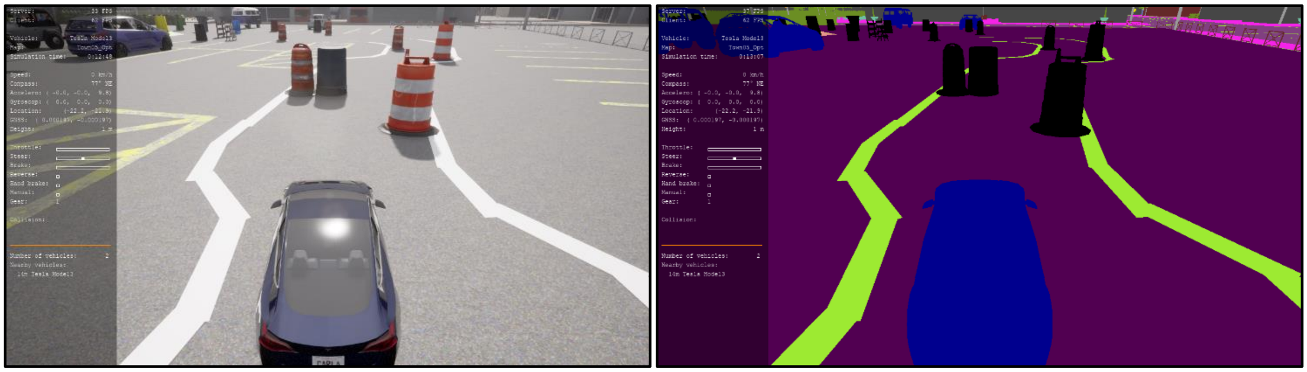


Figure 7-1 - CARLA simulation of vehicle traversing the track: left RGB image, right semantic segmentation ground truth.

7.3 Software Strategy and Path Planning

D* Lite is used to maintain the planned path to the next waypoint as new obstacles are discovered once the vehicle has obtained a bird's eye view of the environment. Waypoints are obtained using an ardusimple RTK GPS system and plotted onto an environment cost map as the path finding goal. Using previously generated bird's eye view plots, a short-term cache of obstacles around the vehicle is maintained. A full overview of the navigation strategy can be seen in Figure 7-2.

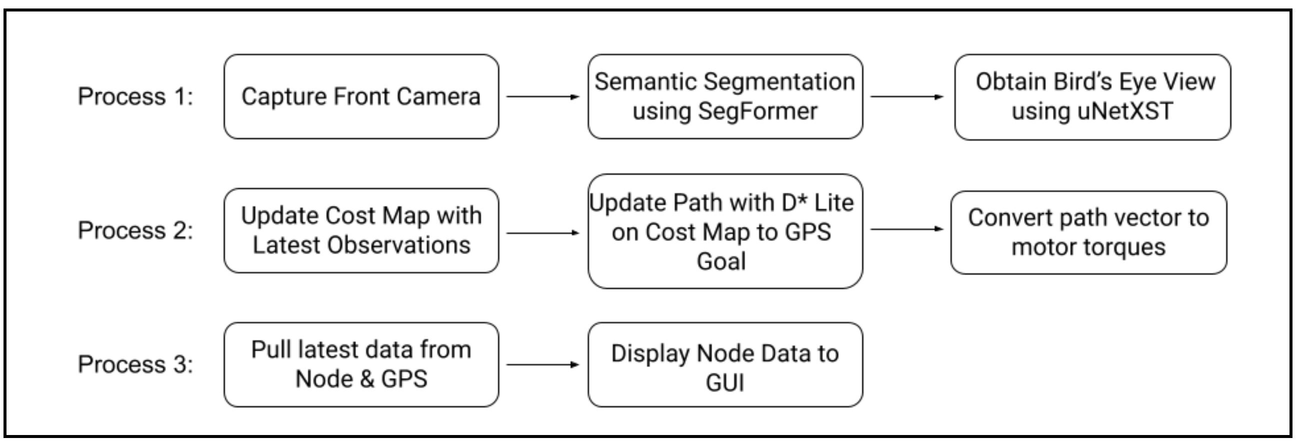


Figure 7-2 - Control Laptop Processes.

The Control Laptop has a dedicated process for the machine learning pipeline, this computation is offloaded onto the eGPU. The path planning operates on its own process using the latest information posted by the machine learning and data intake processes. The Control Laptop interfaces with the Node board via CAN bus. The Control Laptop is responsible for all navigation computation and the Node board is responsible for battery data intake and motor control. The control software onboard the Node maintains an internal state machine which ensures that the vehicle boots into a safe state with no motor output upon receiving an e-stop trigger. Further details of this state machine can be found in Appendix B: Control Subsystem State Machine. The approximate latency from image capture to motor torque updates is 50 ms.

7.4 Map Generation

Multiple bird's eye view segments are combined into a cost map of the environment; segments like obstacles and lane lines are considered high-cost values, the ramp and drivable concrete are low-cost values. Using the cost map, the path planner can effectively stay between lane lines and avoid obstacles. Obstacles are detected at a minimum depth of field of 5 m and often greater.

7.5 Goal Selection and Path Generation

Goals are selected as the next chronological waypoint, preliminary testing has shown the use of intermediary, calculated waypoints can improve navigation. The vehicle is capable of navigating without GPS waypoints to preprogrammed area displacements.

8 Failure Identification and Mitigation

8.1 Vehicle Failure Modes and Resolutions

Potential failure modes were analyzed as a team and addressed throughout the design process. Table 8-1 demonstrates the critical failure paths addressed in the system design.

Table 8-1 - Failure Modes and Resolutions.

Failure mode	Repercussions	Resolution
Tip-over	Loss of maneuverability	• Vehicle geometry has been selected to mitigate tipping risks • In the event vehicle tips over, Wireless E-stop will be used to power down the vehicle
Chassis hang-up on ramp	Loss of maneuverability	• The geometry of the vehicle is such that there is sufficient ground clearance for the maximum angle encountered during the course
Machine vision impairments from weather or lighting	Inaccurate readings of environment in bird's eye view and cost map	• Data set generated from the simulator contains a variety of environment lighting and weather conditions.
Inability to pass between obstacles	Loss of maneuverability	• Omni wheel system allows for full motive control with no perceived configuration where the vehicle will be unable to maneuver between obstacles.
Loss of power	Inability to continue with the course run	• Battery capacity sized for excess run time to allow for multiple runs • Vehicle charging will be taking place as frequently as possible.
Object becomes lodged in wheels	Loss of maneuverability	• Wheels have been given sufficient clearance to ensure any expected debris (small rocks, etc.) can clear the vehicle chassis

8.2 Vehicle Failure Points and Resolutions

Careful consideration has been given to address system failure points. Table 8-2 summarizes the major concerns addressed in the vehicle design.

Table 8-2 - Vehicle Failure Points and Mitigation.

Point of Failure	Repercussions	Mitigation Technique
Battery harnessing breaks or is intermittent	• Possible arcing • Possible rail transients • Loss of total power to the controls system or signal motor	• Hot swap controller integrated into design • TVS diodes added to 52V rails • Firmware detection moves to safe state
Signal harnessing breaks or is intermittent	• Some telemetry (that may be critical to navigation) may give poor values	• Integrated firmware detects and moves vehicle to safe state
Loss of signal to wireless e-stop	• Unsafe situation in the event the vehicle needs to be stopped	• Ensure 915 MHz antenna placements on ground station and vehicle are optimal
Payload becomes dislodged	• Damage to internal components	• Dual redundant methods of securing the payload

8.3 Battery Thermal Event Prevention

There is a battery management system integrated into the OTS e-bike battery. This system monitors the current of the pack and limits it to 30 A. Additionally, there are thermal sensors inside the pack which monitor the temperature of the pack, preventing thermal runaway. The battery is also integrated in the vehicle in a way such that direct sunlight will not heat the battery. Finally, there will be a 30 A, automotive, fast blow fuse integrated in the harness connected to the positive terminal of the battery output - to protect for a thermal event should the BMS fail.

9 Simulation

9.1 Simulations in Virtual Environment

CARLA, an open-source simulator for autonomous driving research built on top of Unreal Engine, is used for all simulations of the vehicle. The main use of the simulator is collecting labeled data for the machine learning pipeline, specifically, of various camera angles of matching RGB images and semantic segmentation ground truth images. Using the Unreal Engine editor in combination with CARLA APIs, the vehicle can be trained and tested in high fidelity environments nearly identical to the competition, this can be seen below in Figure 9-1.

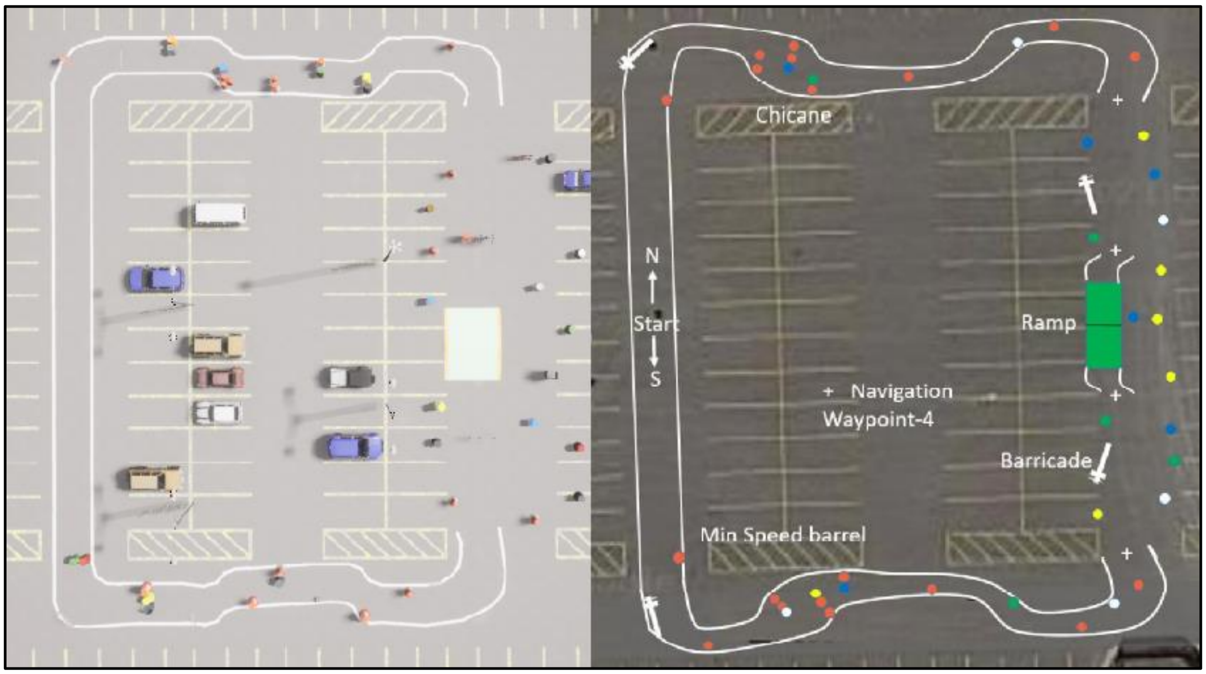


Figure 9-1 - Left recreation of competition track using CARLA, right real competition track.

9.2 Theoretical Concepts in Simulation

Machine learning models are trained on a wide variety of data including different competition arrangements of lane lines and complex obstacle arrangements to aid in sim-to-real transfer. Gathered datasets contain many weather, lighting, and obstacle texture variations.

10 Testing

Due to supply chain constraints, components needed to construct the custom circuit board were late to arrive, thus, full electrical systems testing has not yet been completed. The passive PoE injection used for the Ubiquiti Rocket M900 radio was previously designed and tested by the team for a different student engineering competition. The ESCs selected for the vehicle have been connected to the BLDC motors, calibrated, configured, and tested.

Software testing is performed on two levels - unit and integration. Unit testing evaluates individual software components and integration testing ensures that components are correctly coupled, and subsystem interactions behave as expected.

There were concerns raised regarding the durability of the omni wheels on asphalt surfaces. To alleviate this concern, the team 3D printed wheels for a skateboard made from TPU. It was discovered that the TPU wheel would not hold up to the asphalt surface, so the decision was made to switch to a wheel that used rollers made of SBR rubber.

11 Performance

Preliminary benchmarking has demonstrated the machine learning pipeline has met our internal requirement of greater than 20 FPS on the selected Control Laptop and eGPU.

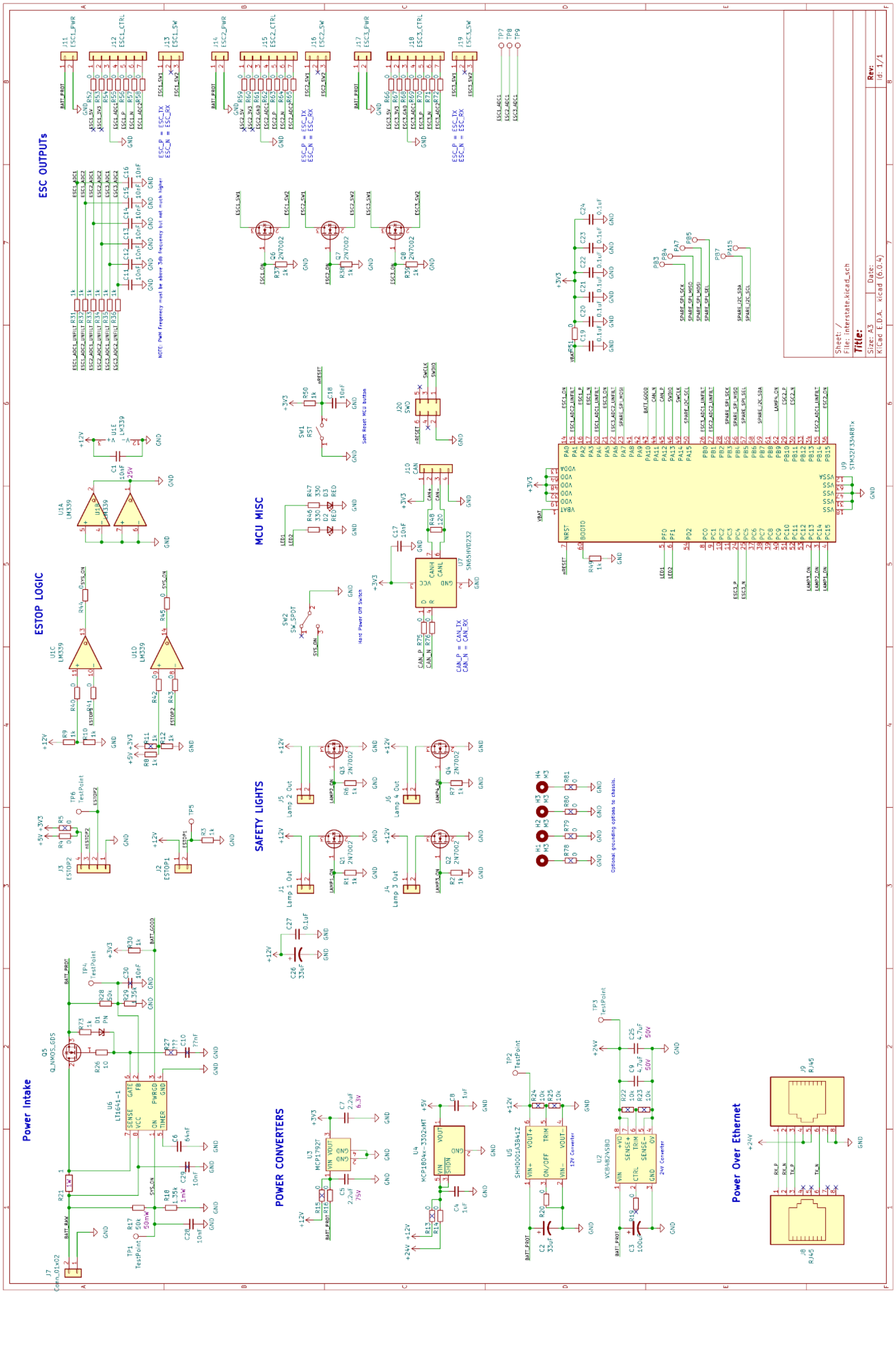
References

[1] E. Xei, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez and P. Luo, "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers," arXiv, 2021. [Online].

[2] L. Reiher, B. Lampe and L. Eckstein, "A Sim2Real Deep Learning Approach for the Transformation of Images from Multiple Vehicle-Mounted Cameras to a Semantically Segmented Image in Bird's Eye View," arXiv, 2020. [Online].

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Appendix A – Node Board Schematic



Appendix B – Control Subsystem State Machine