Embry-Riddle Aeronautical University

A-REX:
Autonomous Robotic Engineered Experience

Team Members:

Date Submitted: 05/17/2021

Parker Tyson	Tony Arrigo	Levi Islam
Justin Bungard	Michael Ahearn	Taylor Johnson
Paul Ryder	Nathaniel Arzola Lilly	Brandon McGannon
Charles Montagnoli		Erasmo Landaeta

Faculty Advisors:

Patrick N. Currier, currierp@erau.edu Charles F. Reinholtz, reinholc@erau.edu Eric Coyle, coylee1@erau.edu,
Christine Walck, daileyc1@erau.edu

Statement of Integrity

I hereby certify that the design and development of this vehicle described in this report is significant and equivalent to what might be awarded credit in a senior design course. This is prepared by the student team under my guidance.

Dr. Charles Reinholtz, Faculty Advisor

1.0 Overview

A Polaris GEM e2 Neighborhood Electric Vehicle equipped with power steering and an extended battery pack was purchased for this project. This vehicle was modified to allow for autonomous and remote control (RC) capabilities. This included developing drive-by-wire control of the steering, brake, and throttle. The converted vehicle is capable of lane following, obstacle avoidance and waypoint navigation under autonomous control. The sensor suite used for perception in autonomous operation includes a Sparton GED-9 AHRS (Attitude and Heading Reference System), front-facing Velodyne Puck 16-Beam LiDAR, Blackfly (formerly FLIR) PGE-2S32 Gigabit Ethernet camera, rear-facing SICK LMS 151 LiDAR and a Hemisphere GPS. A Rocket M5 Radio was also added to allow communication with a remote operator-control unit. Robotic Technology Kernel (RTK) software, provided by U.S Army Ground Vehicle Systems Center (GVSC), formerly TARDEC, was adapted to provide autonomous capabilities. This report describes the development of these systems and the methods used for system integration.

2.0 Conduct of the Design Process

2.1. Introduction

The Embry-Riddle Aeronautical University team is sponsored to perform research under a U.S Army Ground Vehicle Systems Center (GVSC) contract entitled, “Robotic Tool Kit (RTK) Logistics Automation.” Many of the objectives of the IGVC Self-Drive Challenge overlap with objectives of the contract, including converting the GEM e2 vehicle to drive-by-wire operation and equipping it with sensors for autonomy. The team elected to use the RTK software to control the vehicle in order to keep the software implementation and development required in the contract aligned with the goals of competition.

2.2. Team Organization

The development of this vehicle required a multidisciplinary engineering team. The project was supported by six undergraduate students in the fall semester and five undergraduate students in the spring semester. These members cumulatively contributed to the design, manufacturing, and implementation of this vehicle and its software.

2.3. Design Process and Design Assumptions

For each innovation and change to the system, a seven-step engineering design process is used in order to develop a new system. Four key factors were prominent in the planning portion of the design and development process: safety, performance, reliability, and modularity. These key factors allow for the designer to consider many important factors that one may not thinking about. Within this design process, key design assumptions are necessary in order to proceed with the process. A number of key assumptions have been made for the vehicle including the senor range, field of view and update rates needed to traverse the competition course at the maximum allowable speed of 5 mph.

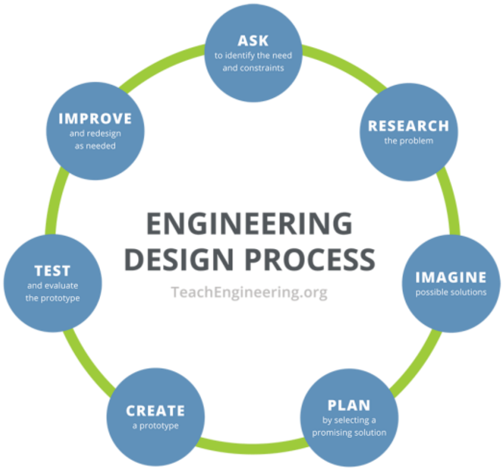

Figure 1: Seven-Step Design Process

3.0 Effective Innovations

3.1 Overview

The innovations and progress made on the vehicle by the team this past year have greatly improved the vehicle safety and performance. Specifically, the adaptations to the vision capabilities and vision processing software as well as modifications increasing the robustness of the E-brake and tuning the controller for smoother power steering.

3.2 Vision Box

In the most recent Self-Drive competition in 2019, A-Rex used a single Gigabyte Brix on-board for all sensor and navigation processing tasks. This system had performed well in testing at the university, but the relatively large, cluttered competition field and heating due to prolonged operation in the summer temperatures caused the CPU to bottleneck. This resulted in long delays in calculating acceptable paths and poor navigation performance. To alleviate this problem, all image processing operations have been moved to a newly added NVidia Jetson AGX Xavier computer. A 5-Port Gigabit Power-Over-Ethernet (POE) from TP-Link was used to receive Ethernet from the Rocket M5 and distribute it to the Jetson and Blackfly Camera, while also providing power to the camera. A small fuse box with two 5A fuses was used for overcurrent protection as well as a 5-screw ground terminal to complete the power system. A USB temperature sensor was included to actively measure internal temperature, and the components were all secured in a UL NEMA 4X Enclosure. A total of 8 NLinko connectors were secured to one panel on the box: 2 Ethernet, 2 Power, 3 USB, and 1 HDMI. The USB and HDMI connectors serve the purpose of allowing the team to plug in peripherals for debugging purposes (mouse, keyboard, storage, & monitor) without needing to remove the box from the vehicle.

3.3 Stop Sign Detection

An innovative stop sign detection system was developed to reliably detect stop signs in a variety of driving settings. The protocol uses a neural network, trained on various images of stop signs in different conditions, to compare the de-noised image. Once the stop sign is found, the system will output a message. The team has implemented the protocol into simulation with the GEM vehicle, and it is able to consistently recognize a stop sign on the course. The protocol was developed with the intent of implementing other traffic signs in the same neural network. An example image of the output of the neural network can be seen in **Error! Reference source not found.**, where the detected stop sign location is highlighted by a green bounding box.

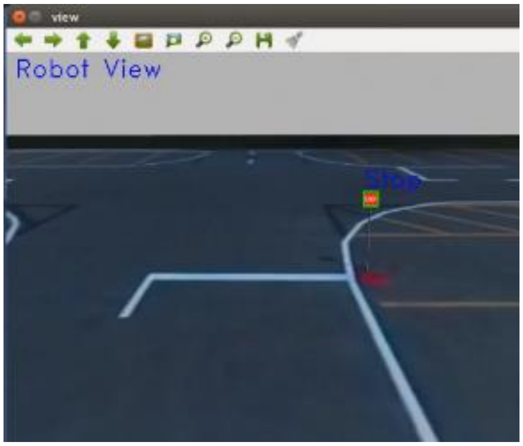

Figure 2: Stop Sign Protocol Simulation

3.4 Lane detection

An edge-detection-based lane finding algorithm was developed in MATLAB. Based on the developed algorithm, pseudocode was developed for a ROS/OpenCV implementation. This method takes in image data from the camera and converts the RGB image to HSV, making it less susceptible to changing lighting conditions when focusing on white and yellow colors. Then, a mask is applied to the image to extract only white and yellow pixels, and a Gaussian Blur is used to reduce noise. Finally, the Canny Edge

Detection algorithm is applied to the now Black and White image using pixel gradients to pull out the edges, and a Hough Space Transform is used to convert those edges into lines in Cartesian coordinates which the vehicle uses as lane boundaries. An example output of this process is shown in Figure 3.

3.5 E-brake

The E-brake implemented in previous years failed to meet competition requirements and therefore was redesigned, the new E-brake system now meetss requirements and is more robust and reliable. Additionally, it minimizes unwanted motion in the system and brings the vehicle from the competition speed of 5 mph to a complete stop within 10 ft when tested under expected competition conditions. The same system, which operates as the parking brake, can hold the vehicle on a paved surface with a 12.5% grade.

Figure 3: Edge Detection-Based Lane Detection Results

3.6 Power Steering motor PID Tuning

The 2020-21 ERAU self-drive team focused significant effort on determining the optimal steering control settings for smooth, precisely controlled steering performance. This was an important improvement, since the previous motor response to Controller Area Network (CAN) steering commands has been erratic for the last 2 years. This was achieved through tuning the GLOBE POW-R STEER® Electric Power Assisted Steering (EPAS) steering motor supplied by Allied motion, over the CAN bus. The tuning now always ensures steady turning for the vehicle during autonomous operation. Table 1 summarizes the method used for PID gain tuning based on the instructions provided by Allied Motion.

Table 1: Power Steering PID Tuning Process

Gain (Tuning Order)	Process	Effect	Starting Value	Approx. Range
P (1)	Increase to desired responsiveness, decrease if wheel motion becomes "Jerky".	Increased responsiveness of steering commands.	10	50-150
I (3)	Increase until control becomes more volatile with new commands.	Stabilize control when giving new command.	0.1	0.1-2.5
D (2)	Start low, increase until unsteady shaking stops improving.	Balance wheel while holding steering angle.	0.01	0.01-0.5
C (4)	Increase until final desired state is achieved.	Lower total error/oscillations.	0.01	0.01-0.05
SL (N/A)	Do not change.	Anything but 0 will be unstable.	0	0

4.0 Description of Software Strategy and Mapping Techniques

4.1. Overview

The team is using the Robotic Operating System (ROS) on a Linux computer to run the Robotic Technology Kernel (RTK) to give the GEM e2 self-driving capabilities. RTK is a software system provided by the United States Army CCDC Ground Vehicle Systems Center (GVSC) and is used by the University for research and development under the restrictions of a Non-Disclosure Agreement. An Mbed microcontroller is used to interface with necessary vehicle systems. Nodes were built in C++ code that pick up encoded serial data from the Mbed and publishes the data as the ROS topics necessary for the RTK nodes to run. A hex file was built to launch some of the sensor drivers and to pass correct parameters to their corresponding launch files. The ROS launch files initialize the nodes and nodelets that interpret sensor data and Mbed commands. These nodes communicate back to the Mbed through serial with the proper driving commands for actuation. For vision and lane detection, a MATLAB script was used to process vision and serialize the data to be published as a ROS topic similar to the LIDAR's point cloud. A remote Windows machine is used to send GPS waypoints and paths through the Warfighter Machine Interface (WMI), which packages the data in a way the RTK understands and can use for navigation. An overview of the RTK modules that are used by the vehicle is shown below, in Figure 4.

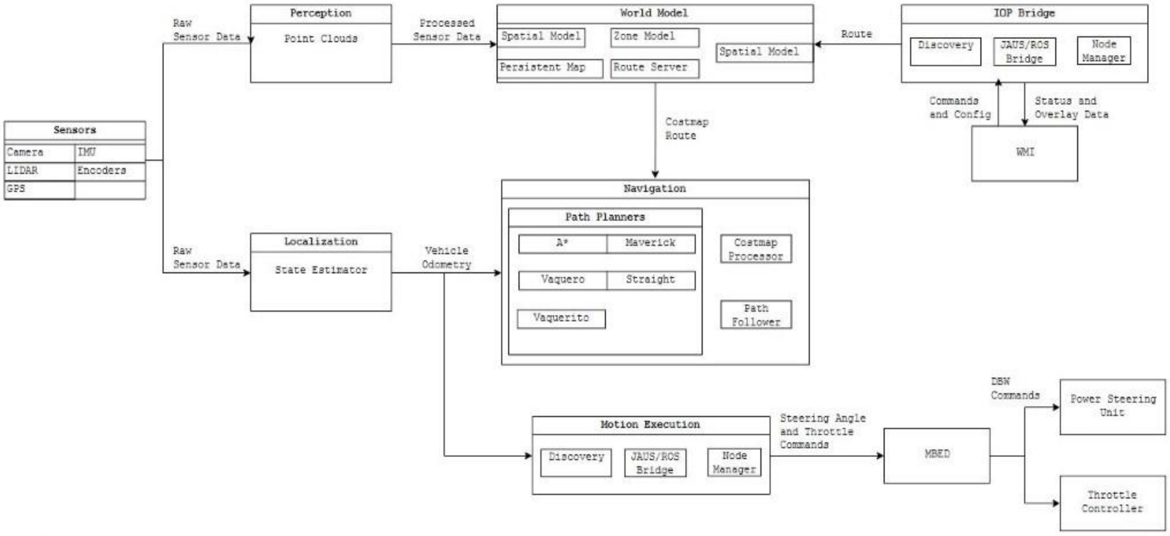

Figure 4: RTK Module Overview

The A-REX team made a ROS package that mimics necessary topics from the RTK motion control system. This package contains the state of the transmission which is used for determining the direction for the near field positioning, the robotic state of the vehicle whether it be stopped, manual, or autonomous, the behavior state which determines which path planner that is going to be used, and transfers and published the encoders. Along with this package, a vision package is publishing positions of the lanes using array markers. These will act as road boundaries or add immediate waypoints that are centered between the lanes. The workflow for this package is depicted in Figure 5.

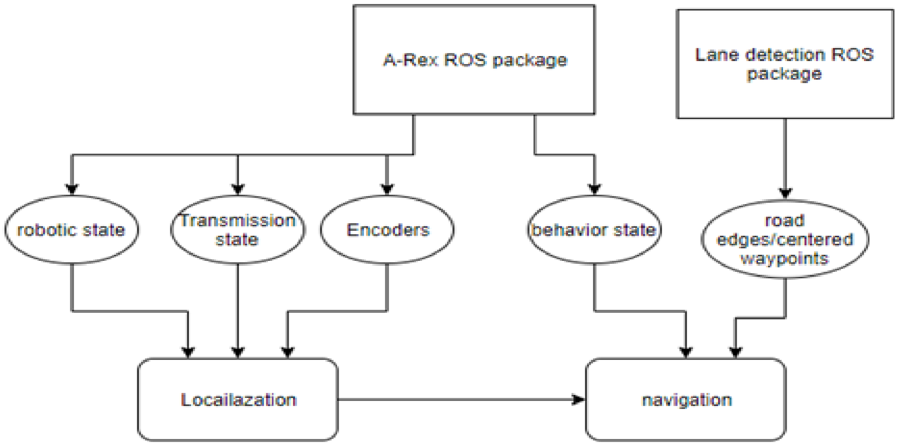

Figure 5: Layout of ROS Packages and Communication

4.2. Obstacle Detection and Avoidance

The vehicle uses a Velodyne VLP-16 LiDAR (Puck) as its main obstacle detecting sensor. When active, this 16 beam LiDAR returns a point-cloud of distances to objects at a rate of 20Hz. The effective field of vision of the LiDAR is 30 degrees vertically (+15 and –15 degrees) and 210 degrees horizontally. To integrate the sensor into the system, the ROS driver for the Puck is used. The device communicates to the main computer via ethernet connection. RTK natively supports this sensor so little modification was needed to be made in order to integrate it into the system.

The Point cloud produced by the Velodyne Puck is fed into the Perception module of RTK where it is processed and fed forward into the World Model module. Once in World Model, cost maps are generated; these cost maps are used to identify areas the vehicle should avoid with higher costs corresponding to elevated avoidance priority. In addition to generating cost maps, the World Model module generates a near-field frame based on sensor data and vehicle starting position, which, along with GPS, is then used to build the global frame or far-field frame. From the world model the cost maps are sent to the Navigation module.

Ultimately, the Navigation module uses one of five available search algorithms (A*, Maverick, Vaquero, Vaquerito, or Straight) to identify a path that avoids any detected obstacles while heading to the current GPS waypoint; the A* Path Planner was selected for use with the vehicle as it achieves what is necessary for the competition course. From navigation corresponding speed and curvatures are outputted to follow a path. These topics are sent to motion execution.

Motion execution takes the speed and curvature output from navigation. This module sends the speed as measured from the encoders through a PID controller, which then outputs a throttle command. The module also sends the curvature though the steering calibration that also includes a PID to the feedback position of the steering wheel, which outputs a steering command. Both the throttle and steering command are sent to the Mbed where these topics are executed. Figure 6 shows a flowchart detailing the entire obstacle-avoidance process.

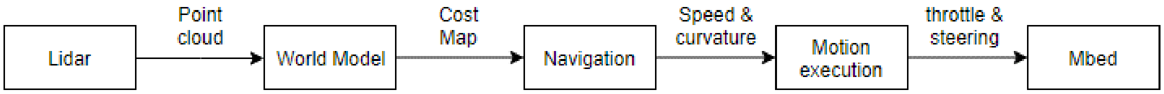

Figure 6: Flow Chart of Obstacle Avoidance Decision-Making

4.3. Software Strategy

The software strategy to integrate the RTK modules given by GVSC into the vehicle's system was to divide and conquer. Independent subgroups worked on getting specific modules operational later integrate everything together. One group worked on the localization sensors (GPS, IMU, Encoders) that we chose and made ROS drivers to publish data in a format that the RTK Summet State Estimator node would be able to use. The second group was working on the vision and perception nodes (Camera, LiDAR). The VLP-16 LiDAR is natively compatible with RTK, so their job was to install the correct driver and change the source code to work with the 16 laser Velodyne rather than the 32 laser one. Vision in RTK is not used for the same purpose that the competition requires (Lane detection, obstacle detection/classification), so the team only focused on getting ROS to publish camera data. These teams were also in charge of making hex and launch files that would ensure the sensors were connected and sending data, as well as creating any other C++ code necessary for pre-processing of sensor data. The was another group working on Mbed code that would be used for the control module and drive-by-wire.

4.4. Path Planning, Goal Selection and Path Generation

Path planning in the RTK depends on the WMI waypoints given and on the RTK's obstacle avoidance, world model, and localization nodes. The waypoints need to be set in the WMI when the vehicle is in autonomous mode and has a "ready" health status (not E-STOP or any other error state). The vehicle uses localization node to transform the coordinates into the near-field and builds and navigates a modified Euclidian path towards the goal. The LiDAR data used by the world model node creates a cost map with obstacles and no-go zones that are used by the built-in A* algorithm in the RTK path-planning node to create the modified Euclidian path as the vehicle moves and the map gets updated. Vision will detect lanes and create intermediate waypoints between for the RTK's path planning and navigation unless no lanes are detected or there is an obstacle in the way. No lanes will depend solely on RTK path planning, and detection of an obstacle during lane following will cause a change of lane or full stop. Figure 7 depicts a flowchart illustrating the logic flow from user input GPS waypoints to output throttle and steering commands.

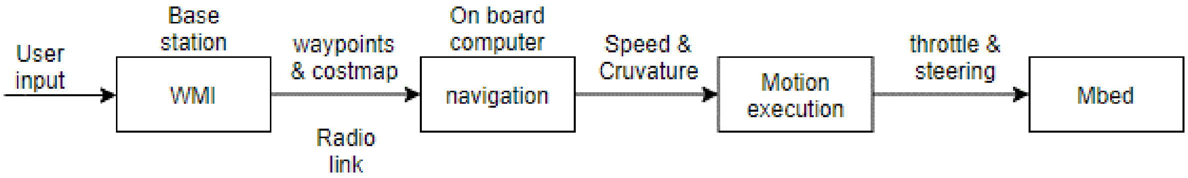

Figure 7: Flow Chart of Path-Planning and Autonomous Decision-Making

4.5. Map Generation

The World Model portion of RTK is responsible for generating maps that are used for vehicle navigation. Specifically, as sensor data is acquired and processed by the Perception module, cost maps are generated that correspond to the environment directly around the vehicle. Furthermore, these cost maps are used to create a vehicle near frame: the near frame is a map that includes obstacles and their distances to the vehicle with respect to the vehicle's origin. Once a near field map has been generated, it is placed into the global frame by attaching GPS data. These maps are then used by the navigation module to generate a path for the vehicle to travel that avoids obstacles and reaches designated waypoints.

5.0 Description of Mechanical Design

5.1. Overview

The mechanical design approach for this project was to limit the modifications from the stock vehicle. This also aligned with our modularity and marketability goal of making the autonomy kit an “add-on” to the stock GEM e2 vehicle. The specifications of the A-REX GEM e2 are shown in Table 2.

Table 2: Polaris GEM e2 Vehicle Specifications

Vehicle Specification Table	
Engine & Drivetrain	
Battery Voltage	48V
Drive	Direct Front Wheel
Motor Size	6.7 HP (5.0 kW)
Motor Type	AC Induction
Top Speed	25 mph
Dimensions	
Cubic Feet of Cab	70 ft³ (2 m³)
Estimated Dry Weight	1,650 lb (748 kg)[1]
Ground Clearance	8 in (20 cm)
GVWR	2,000 lb (907 kg)
Overall Vehicle Size (L x W x H)	103 x 55.5 x 73 in (261.6 x 141 x 185.4 cm)
Payload Capacity	800 lb (363 kg)
Person Capacity	2
Rear Cargo Box Capacity	330 lb (150 kg)
Turning Radius	150 in (381 cm)
Wheelbase	69 in (175.3 cm)
Tires/Wheels	
Front Brakes	Disc
Rear Brakes	Hydraulic Drum
Tires	13 in. street-rated 155/80 R13

---

[1] Note that the maximum weight requirement, as specified in 1.2.2 Design Specifications in the 2019 IGVC Rules, was waived for teams using the GEM e2 vehicle, due to the additional weight of modifications for the Drive-by-Wire conversion and extended battery pack. This was approved by Mr. Gerald Lane of GLS&T via email on 09/18/18 stating that the Gross Vehicle Mass Weight (GVMW) of up to 2000 lbs. was allowable.

Suspension	
Front Suspension	MacPherson Strut - 5.6 in (14 cm)
Rear Suspension	Independent Trailing Arm - 6 in (15 cm)

Two major classes of physical modifications were made to the stock GEM e2 vehicle: the addition of sensors and electronics and conversion of steering, throttle, service brake and emergency (parking) brake to by-wire operation. The emergency stop system can be activated by any of the five emergencies stop buttons on and inside the vehicle or by remote control.

To help select and locate the sensors, a Simulation Cuboid Model was developed in MATLAB to confirm the sensor field of view and to determine the optimal sensors and sensor placement detailed in the "Electronic Suite Description." A top-down view of the modeled environment is shown in Fig. 7. The Simulation Cuboid Model is a MATLAB tool that uses the camera and LiDAR position and technical specifications to obtain an expected range of view. The Simulation Cuboid Model ensured no blind spots with the current sensor placement.

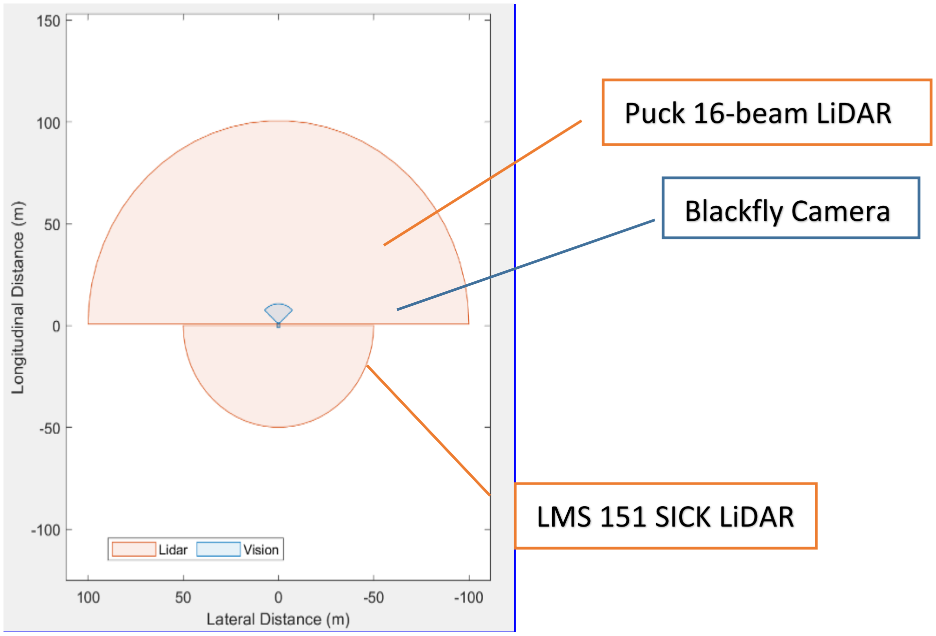

Figure 8: MATLAB Simulation Cuboid Model

An 80-20 T-slot aluminum framing system was added to the top of the vehicle to mount the Hemisphere GPS and two safety lights, as shown in Figure 9. The sensor bar attaches to the preexisting rails of the GEM e2, which allows for a clean finish and limits any physical modifications needed. The sensor bar attachment also includes risers and a track guard. The risers prevent the sensor bar from physically touching the vehicle, eliminating any friction, or chafing leading to potential cosmetic damage. A rubber track was also added to prevent rain and litter collection in the 80-20 rails. Because the sensor bar attaches to the vehicle rails, it can be adjusted forward and aft on the vehicle.

Figure 9: Sensor Bar Mount

5.2. Emergency Braking System

A fail-safe emergency braking system was a modifying the stock parking brake system. With the new design, an electromagnet holds the parking brake in the deactivated position. When the vehicle loses power, or when any of the e-stop buttons are pressed, the magnet is deenergized and parking brake is activated by a spring with stored potential energy. This system is fail-safe in the sense that the brake is applied by default, and it is only disengaged when it is held in place by the magnet.

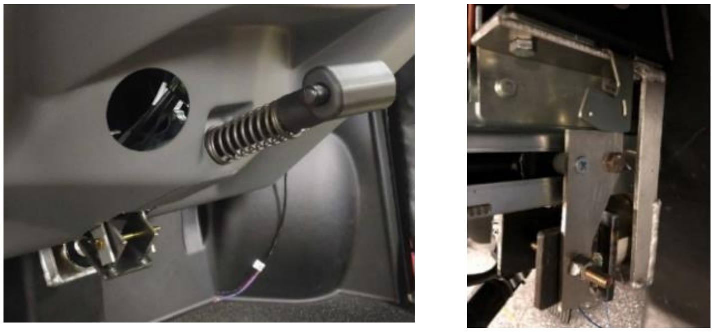

Figure 10: Stock-issued Braking Mechanism

When the vehicle is powered and in use, an active electromagnet is used to keep the parking brake disengaged. Once the vehicle loses power, the electromagnet will also lose power. This will result in a fail-safe braking system. When de-energized, the electromagnet loses contact with a plate connected to the parking brake. The spring will then force the parking brake to engage, thereby causing the car to come to a safe stop.

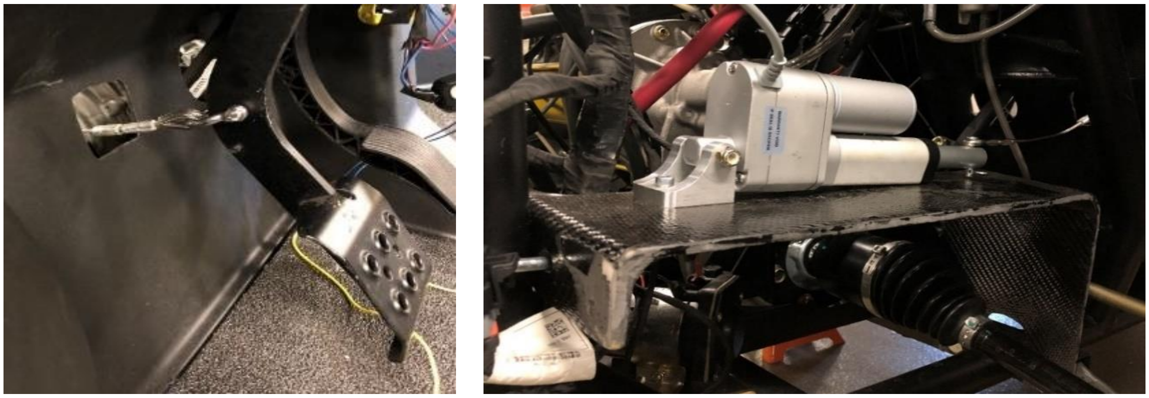

Figure 11: Re-designed Emergency Brake System

There are five E-Stop buttons located on the vehicle. Side safety stops are mounted on the side panels, as shown in **Error! Reference source not found.**. The exterior buttons are mounted on the existing vehicle track system and oriented to the side of the vehicle. This discourages someone from trying to stand directly in front of or behind the vehicle to use the e-stop button.

A fifth e-stop button is located between the driver and the passenger seat of the vehicle so that a passenger could press the button if they felt unsafe. If any of the buttons are pressed, power is cut to the contactor, thereby cutting power to the motor, as well as cutting power to the electromagnet. The RC controller will also have an E-Stop switch that works the same as the buttons on the vehicle.

Figure 12: Emergency Stop Buttons Drive-By-Wire Conversion

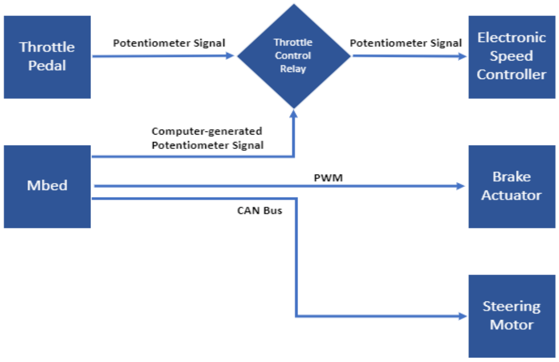

Figure 13: Drive-by-Wire Structural Outline

To convert the vehicle to a drive-by-wire system, steering, throttle, service braking, gear shifting and turn signal functions were first analyzed. Figure 13 details the Drive-by-Wire functions and their relationships.

The Polaris GEM e2 uses two potentiometers to send signals to the motor controller to change the speed of the vehicle. The potentiometer's output voltages from the throttle pedal were first read using an Arduino UNO microcontroller. After the voltage data was collected, a linear function relating the voltage output from the throttle pedal to the vehicle's speed was developed. This relationship was then used to create digital signals that related to a desired speed from an Mbed microcontroller. The digital signals sent from the Mbed are pulse-width-modulated (PWM) signals. The speed controller on the GEM e2 expects an analog signal, therefore a low pass filter was implemented into the circuitry to control the vehicle's speed. The low pass filter consisted of a capacitor and a resistor, in which high frequencies are filtered out by the capacitor only allowing lower frequencies to pass through the circuit. This smoothed out the PWM signal to be read by the speed controller as a continuous analog signal. The low pass filter had a cut off frequency of 10 Hz. To switch from manual and autonomous modes, the throttle pedal and the Mbed were connected to a relay that normally allows the raw throttle pedal signals to pass through.

To limit the speed of the GEM e2 to 5 mph the software will not allow the Mbed to send PWM signals that relate to a speed higher than 5 mph.

The GEM e2 uses an Allied Motion autonomous ready POW-R STEER Actuator. This actuator is controlled using CAN signals. CAN signals are 64-bit messages in which the first part of the message is the message identifier. Identifiers for a command message and a feedback message were found in the manual for the power steering actuator. Once the identifiers were found the next step was to send a message to the power steering actuator to make it actuate. To send a message to the power steering unit the first byte of the CAN message indicated the mode for the power steering actuator as explained in the Manual. Mode 5 was used to allow the user to send a desired wheel position and wheel speed to the power steering actuator. The desired steering wheel position is sent as a 32-bit signed integer that is then split into 4 separate bytes in the CAN message. The same process was done for the steering wheel speed with a 16-bit integer split into 2 bytes. For the feedback, the same process for sending a message was used in reverse where 4 bytes representing the steering wheel position were converted into a 32-bit integer and then converted into position in revolutions. The power steering actuator was limited to 2.5 revolutions to the left or right.

The vehicle indicators and gear shifter operate on similar systems, with 3-state pass through relays governing the mode of each system. For the blinker, the three modes are left blinker, right blinker, or neither. For the gear shifter the modes are forward, neutral, and reverse. Using simple MBED-controlled relays, the vehicle can easily toggle any of these modes, and set the desired driving gear or activate the required turn signal.

5.3. Weatherproofing

A waterproof IP65 (Rated Dust tight and protected against water projected from a nozzle) L-com NEMA (National Electrical Manufacturers Association) enclosure was used to house the computer, many of the power converters and other sensitive components. This enclosure makes the electronic subsystem modular and easily transportable. The electronic connections are routed through a connection panel that is also IP65 rated to allow sensor communication with the internal laptop while allowing for the system to remain waterproof.

6.0 Description of Electronic and Power Design

6.1. Overview

The power system was designed to provide the ability to run all on-board components from the existing vehicle 48-volt power system. The first step in the design process was to create a wiring diagram to develop an understanding of how to place the wiring throughout the vehicle. The wiring diagram was also used to determine the component box requirements based on equipment stored.

Next was to outline the initial component boxes and design a mockup. Once completed and verified the component cases were ordered and received. Then the mockup design was made a reality be modifying the ordered component cases. Wires were then installed in the vehicle and connected to the installed component cases. This marked the completion of the setup of the electrical system.

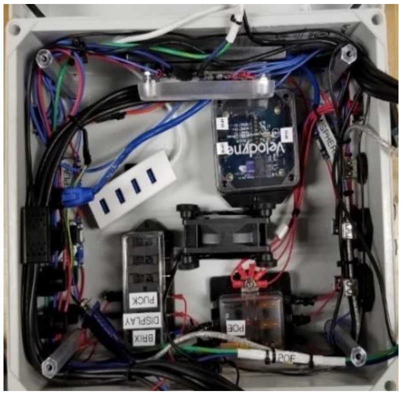

Figure 14: Internal Top View of Main

6.2. Power Distribution System

The battery and charging system remain the same as on the manufacturer-supplied vehicle. The manufacturer distance AGM battery option that doubles base vehicle rage from 30 to 60 miles was selected to allow for longer run times with added equipment. The battery pack for this option includes 8 batteries at 6v each for a 48v system. The power for all added components was converted from 48v to 24v and 19v. The component box system is controlled by a switch in the cabin of the vehicle controlling a relay. Buck Converters were used to convert 48v battery power to these other required voltage levels. This eliminated the need to add a second power source to the vehicle. The power for the drive-by-wire system was kept separate from the power for the sensing system. Designing it this way was important because it allows the sensing system to continue sensing the surroundings even if the power is removed from the drive-by-wire system. Power could be removed either by turning it off or by hitting one of the emergency-stop buttons.

Figure 15: Buck Converters

6.3. Electronics Suite Description

The Gigabyte Brix S computer utilizes an Intel Kabylake i5-7200U processor to process data and perform navigation calculations. The Sparton IMU houses a 3-axis gyroscope, magnetometer, and accelerometer. The Velodyne Puck 16-Beam LiDAR is a multi-plane LiDAR used to detect objects in front of the vehicle. The Intel Real Sense Camera is a depth perception camera, being utilized for sign detection and could be used in the future as a redundant detection device to the Velodyne LiDAR. The SICK LiDAR is a single-plane LiDAR used for detecting objects behind the vehicle while in reverse. The POE Switcher is used for the computer to manage the data coming more easily from the added components. The Rocket M5 Radio is used for communication to and from the base station. The Hemisphere GPS is used in conjunction with the Sparton IMU to provide precise location and heading information.

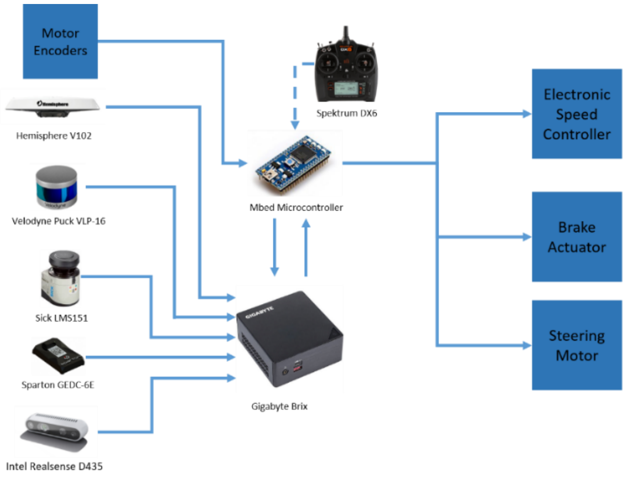

Figure 16: Electronic Suite and Functions

7.0 Description of Failure Modes, Failure Points and Resolutions

Although the vehicle was designed for maximum safety, failures are always possible. To combat this possibility, several different failure modes have been identified and resolutions have been implemented to minimize the negative results of any failure.

7.1. Vehicle Failure Modes and Resolutions

The software is only as reliable as sensor data, so communication with the sensors is a critical function. The RTK Health Monitoring System is constantly monitoring the sensor communications for any errors that may arise from communication protocol issues or disconnections. It does this by monitoring the "heartbeat" frequencies of the sensor signals. If the frequency of the data packets from these sensors or systems changes, the RTK will recognize that an error may be occurring and will enter the appropriate mode to counteract the error. Entering any failure state will result in the emergency stop system be engaged, and an error message will be displayed on the dash-mounted monitor. The failure state is also reported to the base-station computer through the WMI, so remote operators will know that the vehicle has encountered a fatal error.

7.2. Vehicle Failure Points and Resolutions

Mechanically, the vehicle is a stock GEM e2 platform enhanced with actuators, sensors, and computer control. We expect the base mechanical vehicle systems to generally maintain the reliability of the original system. In designing the subsystems added to the base vehicle, the team focused on simplicity and modularity. As a result, components are easily removable and replaceable. An onboard toolbox that contains necessary tools to swap out many components and subsystems is carried on-board the vehicle for quick repairs.

It was important that the driver always has priority in controlling the vehicle. Along with the software controlling the linear actuator to stop the vehicle, the driver can always use the standard brake pedal to bring the vehicle to a stop. If power to the vehicle is lost, the electromagnet is disengaged and the parking brake will be applied, bringing the car to a gradual stop.

7.3. All Failure Prevention Strategy

If failure occurs, emergency precautions have been implemented into the software and hardware of the vehicle. Our vehicle is equipped with five emergency-stop buttons. Four are located on the outer sides of the vehicle and one in the center console inside the vehicle. The outer E-Stop buttons were arranged so that someone on the outside of the vehicle would be able to access it without placing themselves in the path of the vehicle. The inner E-Stop button is easily accessible to both the driver and passenger, located between their seats.

7.4. Testing Vehicle Safety Design Concepts

The vehicle was tested in an outside environment similar to the IGVC competition course. Before the outdoor test could be conducted, each subsystem had to pass a quality assurance check to make sure the systems were still working. Each of the sensors, being reused from last year, were checked in the lab to ensure the cables still had a good connection for both power and data.

Testing started with jacking up the vehicle to test the throttle, steering and breaking control for the drive by wire system. Before outdoor testing occurred, the vehicle was thoroughly checked over to ensure all safety features were working. This included the emergency stop buttons, seat belts were used, helmets were worn securely, and the testing cite was cleared.

The fail-safe emergency brake system implemented in the original GEM conversion was unreliable and would not always allow an operator to set it. Furthermore, when in the brake off mode (set for fail safe actuation) a microswitch that was designed to indicate the brake was off and the vehicle was ready for operation would sometimes not be depressed. This motivated the redesign of the e-brake system discussed previously. After implementing the redesigned parking brake, all wiring related to the emergency braking system was tested. Once all wiring was confirmed to be fully functional all e-stops were tested to verify they would disengage the electromagnet causing the parking brake to engage and cutting power from the vehicle motors. After all stationary testable safety features were checked, the vehicle was driven to an isolated area with similar terrain to that expected at competition were the limits of the system were tested. On marked pavement the vehicle was driven up to competition speed and the e-brake engaged upon crossing predetermined pavement markers. All e-stop buttons were used in the tests and results were consistently within predetermined acceptable margins. The parking brake limits were further tested on measured pavement ramps to determine the maximum slope at which the parking brake could hold the vehicle with two passengers. In a similarly isolated area, after bringing the vehicle to a complete stop on a ramp with measured slope, the parking brake would be engaged followed by slow release of the brake pedal to see if the vehicle could maintain parking on the slope with parking brake alone. The angle tested on was gradually increased until failure and then tested several times on similar slopes to that of initial failure.

8.0 Simulations

A simulation using the ANVEL program was developed. The program implements the parameters of the sensors used by the Self-drive vehicle. These sensors include the Velodyne Puck for the front LiDAR, the Hemisphere V2 for the GPS, the Sparton GEDC-6 for the IMU, the Intel RealSense D435 for the camera, and the SICK LMS151 for the rear LiDAR. The vehicle used in the simulation is the first iteration model of the Polaris GEM e2. The CAD model of the vehicle was created in SolidWorks. It is comprised of the front bumper, the side bumpers, the rear portion, and the middle frame as well as all the sensors that have been mounted to the vehicle. All parts were drafted by the team besides the sensors since the manufacturer's provide STEP files for their products. Although detailed interior information such as seats, battery placement, and engine components were not included in the CAD model, sensor placement, vehicle height and width, and all sizes were replicated very accurately in order to make simulations as realistic as possible. While creating the vehicle for ANVEL, the vehicle definition file was altered to resemble the specifications of the GEM e2, including the electric motor and the sound that the vehicle would produce.

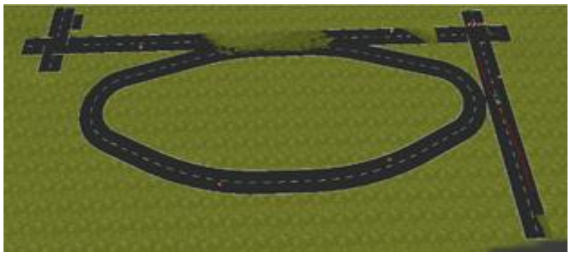

Figure 17: Recreated 2017 IGVC Self-Drive course

Once the sensors in the simulation were coded to generate data , the complete vehicle was set on a reconstructed course, as shown in **Error! Reference source not found.**. The course was designed based on the specifications from the 2017 IGVC Self-Drive Challenge. The resources for this information were a video and pictures by Peter Hanlon of USMA West Point that was posted on Great Lakes Systems and Technology's website. The course was finished by adding signs and obstacles like those seen in the previous competitions. Once this was completed, a mock simulation was run to verify the collection of data by the sensors.

With the ROS plugin for ANVEL 3.0, the ANVEL simulation will be able to run with RTK. This will simulate the vehicle and test RTK modifications, prior to testing the software on the actual vehicle. Doing so will allow the team to detect software issues in a safe environment, limiting the risk to people and equipment. The simulation will also allow virtual testing on the 2017 IGVC Self-Drive Challenge track, which is not available to teams in a physical form.

9.0 Performance Testing to Date

Integration unit tests were performed at each stage of development. The simplest tests involved just the individual sensors. These tests ranged from verifying that a sensor was sending appropriate data, to testing the camera system outdoors. More complex testing was done on the Drive-By-Wire system and on the autonomous systems.

Figure 18: Obstacle Avoidance Testing

The first system that was developed and tested was the Drive-By-Wire system. Each sub-system was verified for safety, beginning with the throttle, followed by steering and braking. Testing the throttle began in remote-control mode while the vehicle was lifted on jack stands. The throttle was tested at various levels to confirm that throttle control was precise and reliable, especially at the low (below 5 mph) speeds required in competition. Reverse and Neutral "gear" changing were also tested initially with the vehicle on jack stands. Initial testing of steering control was also performed using the remote control with the car was lifted. Once the steering system was determined to be reliably controlled, steering and throttle were tested at the same time. This process was repeated for the braking system. Live road tests were conducted afterwards. These tests included driving the vehicle exclusively by RC with the driver prepared to hit the emergency stop and the service brake at any time.

The autonomy tests were by far the most rigorous tests done on this vehicle as the autonomous systems create the greatest safety concerns should any errors occur. All autonomy tests took place in a controlled environment, with only testing personnel in the vicinity. During autonomous testing, two people were always in the vehicle wearing helmets and seatbelts. In addition, the remote control was also with them, and the driver was prepared to hit the service brake to override the autonomous system while the passenger operated the emergency stop system. Autonomy tests were conducted progressively, starting with simple paths with no obstacles and gradually getting more complex. The first tests involved testing only GPS waypoint following. The vehicle was given a single waypoint to head towards, and it was then started in various positions and orientations relative to the waypoint. Once these tests were wholly successful, obstacle avoidance was added. In this case, the vehicle was given a single GPS waypoint to head towards, but this time an orange traffic barrel was placed in its path. Similar to the waypoint following tests, the vehicle was started in various positions and orientations. Other tests were conducted as well, such as guiding the vehicle to navigate between two obstacles on its way to the goal and having to weave between obstacles and hit multiple waypoints along the way. **Error! Reference source not found.** shows an example set up of obstacle avoidance testing, where the vehicle is being encouraged to drive between two orange traffic barrels. The last autonomous system to be tested was the lane-following system. In order to test this, similar tests to the ones described above were conducted. First, the vehicle was given one waypoint, and made to reach it by following a set of lanes. Then, the tests grew more complex, including the addition of obstacles, as well as multiple waypoints to hit along the path. In these tests, the vehicle was also started in various orientations, and the lanes were set up in such a way that there was no straight shot to each waypoint.

10.0 Initial Performance Assessment

At the time of this report, RTK-based GPS waypoint following, and obstacle detection and avoidance are functioning reliably. Additionally, all hardware and communication systems are functional. A vision module has been developed that can perform lane detection, but this capability has not been fully integrated on the vehicle as part of the RTK. A user can insert waypoints using the Warfighter Machine Interface (WMI) provided by TARDC and the vehicle is able to autonomously plan a path and follow it, avoiding any obstacles that it encounters. As discussed in section 9 above, this capability has been successfully tested. Both the main navigation computer and the drive-by-wire microcontroller can communicate with all appropriate sensors, actuators, and onboard electrical components. The mechanical design is complete, and all sensors have been mounted and are functioning as intended.