Embry-Riddle Aeronautical University

A-REX:
Autonomous Robotic Engineered Experience

Team Members:

Zachary Bryant (Team Captain), bryantz1@my.erau.edu

Date Submitted: 05/15/2019

Zachary Bryant	*bryantz1@my.erau.edu*	Nick Cambria	*cambrian@my.erau.edu*
Justin Bungard	*bungardj@my.erau.edu*	Otto Legon	*legono@my.erau.edu*
Matthew Bolotte	*bolottem@my.erau.edu*	Kody Miller	*millek36@my.erau.edu*

Faculty Advisors:

Patrick N. Currier, currierp@erau.edu Charles F. Reinholtz, reinholc@erau.edu Eric Coyle, coylee1@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. The team modified the vehicle to provide drive-by-wire control of steering, brake and throttle, and added remote control (RC) and autonomous capabilities. The converted vehicle is capable of lane following, obstacle avoidance and waypoint navigation under autonomous control. The sensor suite used for perception in autonomous operating includes a Sparton GED-9 AHRS (Attitude and Heading Reference System), front-facing Velodyne Puck 16-Beam LiDAR, Intel Real Sense 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. The 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

Embry-Riddle Aeronautical University was one of six teams sponsored to preform 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 overlapped with objectives of the contract, including converting the GEM e2 to drive-by-wire operation and equipping it with sensors for autonomy. Although it is not required in the Self-Drive Competition, the team elected to use the RTK software to control the vehicle. This decision kept 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 seventeen undergraduate students in the fall semester and sixteen undergraduate students in the spring semester. These members cumulatively contributed nearly 4800 labor hours into the design, manufacturing, and implementation of this vehicle and its software. Figure 1 shows the breakdown of responsibilities and primary contributions of each sub-team. Sub-teams overlapped on some work statements and efforts. Figure 1 illustrates the organizational structure of the team. Note that the software development branch included responsibility for decision-making logic used on the vehicle.

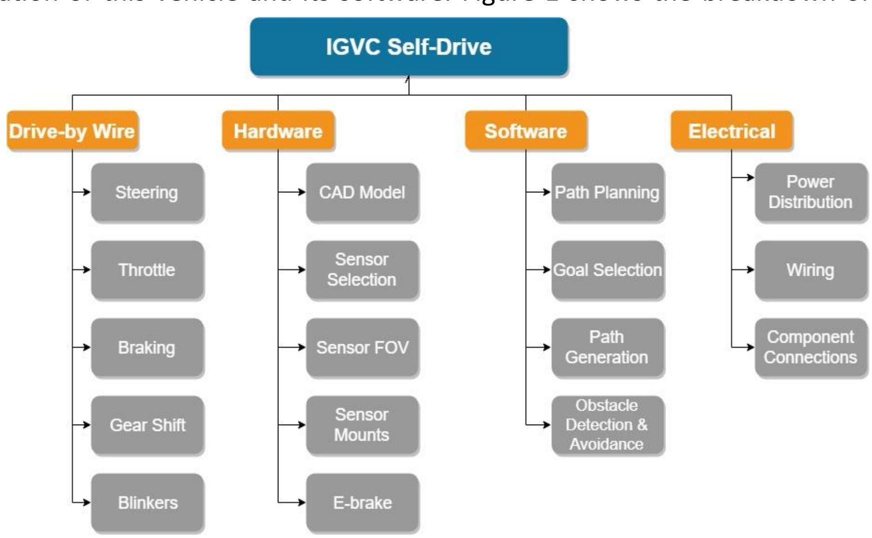

Figure 1: Work Organizational Structure of the IGVC Self-Drive Team

2.3. Design Assumptions and Design Process

The vehicle system was developed using the seven-step design process shown in Figure 2. The problem statement originated from the 2019 IGVC rules. The customer needs were identified from the competition rules, University, and GVSC needs and constraints. The Idea Generation, Solution Exploration, and Selection process occurred in the fall of 2018 during the initial planning and proposal phase of the project. During this phase of the project, design assumptions were made regarding the team's ability to adapt the RTK to the A-REX vehicle and its sensor suite. We also made assumptions about the senor range, field of view and update rates needed to traverse the competition course at the maximum allowable speed of 5 mph. Modeling and testing were performed iteratively throughout the entire process.

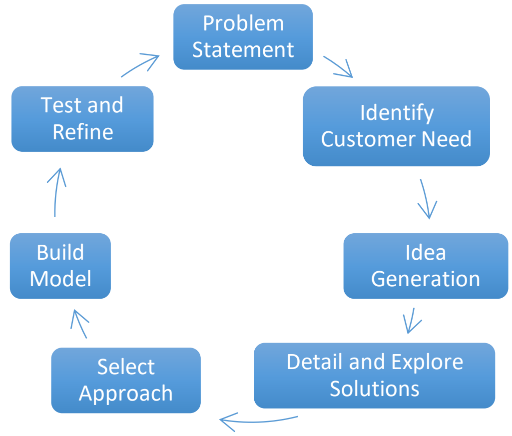

Figure 2: Engineering Design Process

Four key factors were prominent in the design and development process: safety, performance, reliability, and modularity. Throughout the design and development process, the team also considered the ethical and social implications of this technology and how it may affect individuals and industries outside of the competition.

3.0 Effective Innovations

3.1. Rocket M5 Communication System

The computer that was selected for use with the vehicle was a small-form-factor desktop computer, the Gigabyte Brix S. In normal operation as a desktop computer, it would require a mouse, keyboard and monitor to be placed in the vehicle. To avoid this cumbersome set up, the vehicle was augmented with an Ubiquiti Network Rocket M5 system, shown in Figure 3. This system allows for software editing from a ground station with limited latency in the connection, thereby eliminating the need to bring the peripherals into the vehicle to work.

Figure 3: Rocket M5

The Rocket system is comprised of two transceiver antennas that communicate via a 5GHz radio signal. One antenna is mounted on the back-left of the vehicle, appended to the existing track system, and interfaces with the onboard computer via ethernet. The other antenna connects to the external base station computer via ethernet. When the base station computer and vehicle computer are connected to their respective antennas, a private network is established between them, which allows the two devices to communicate wirelessly. TeamViewer, a remote desktop software, is then used to grant the external computer access to control and monitor the main onboard computer without the need to be physically inside the vehicle. While this function will be disabled during competition runs to comply with the rules, it enhances both safety and productivity during testing and development.

3.2. Sensor Initialization and Detection Module

Running setup code for individual sensors wastes time and, if done incorrectly, may degrade performance. To avoid these problems, an innovative software module was developed that condensed the sensor setup and verification into a single executable. This executable can be launched with a single command line. It ensures that all necessary localization sensors are connected and that their data is being published in ROS. It also takes care of launching the core nodes of the localization RTK module with the correct parameters. This is a useful innovation used during all testing as a check to verify that all sensors are plugged in and reporting data.

3.3. Sunlight-Readable Monitor

A common Human-Machine Interface (HMI) problem at the IGVC is the inability of operators and programmers to see information on standard LCD screens in bright sunlight. The A-REX team solved this problem by integrating a Highbright Sunlight Readable marine LCD touchscreen monitor with anti-reflective enhancements. This allows for status checks and quick software fixes in the vehicle outside during testing. Unlike other monitors, it is not affected by glare and it is interactive.

Figure 4: Sunlight Readable Monitor

3.4. Vehicle Service Toolkit

Another common problem in the IGVC is field repair. Teams typically have a large tool set at their disposal in their home laboratory, but they often do not transport this full set to competition. Team A-REX has endeavored to use a limited set of fastener sizes to minimize the number of tools required for maintenance and repair. For example, most on the hex head fasteners on A-REX require a 1/2 inch, 7/16 inch or 15 mm wrench. This allows us to carry a service toolkit, the contents of which are shown in Figure 5, on board the vehicle. The hardware on the vehicle has also been designed to be easily serviceable and replaceable.

Figure 5: On-board Maintenance Took Kit

3.5. LED Safety Panels

As an additional safety feature, two small and three large LED panels will be located around the Polaris GEM e2 top exterior. The three larger panels will be mounted on the back and side panels of the vehicle. The two smaller panels will be mounted on the front of the vehicle. Phrases, including "AUTO," "STOP," and "MANUAL," will be displayed on the LED panels. Although the current safety lights satisfy the competition safety requirements, the LED panels further enhance safety and provide useful information to judges and spectators. An image of the LED panels, which are still in development, is provided in Figure 6.

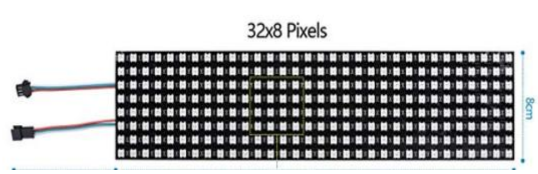

Figure 6: LED Safety Light and Information Panel

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). It 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 7.

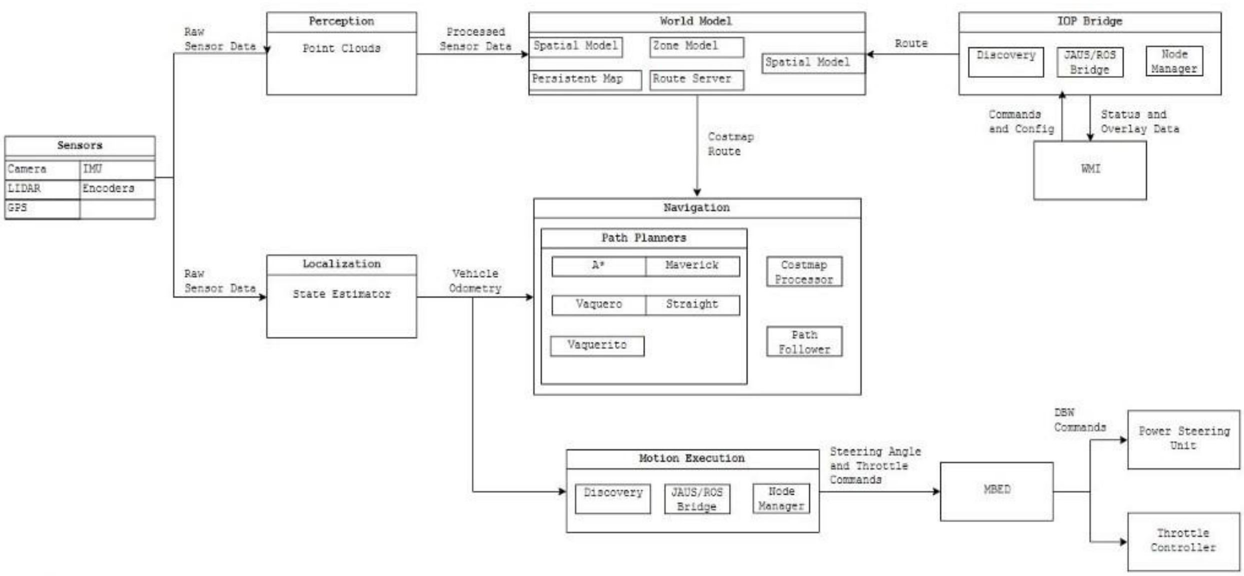

Figure 7: RTK Module Overview

The A-REX team made a ROS package to mimic 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 either 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

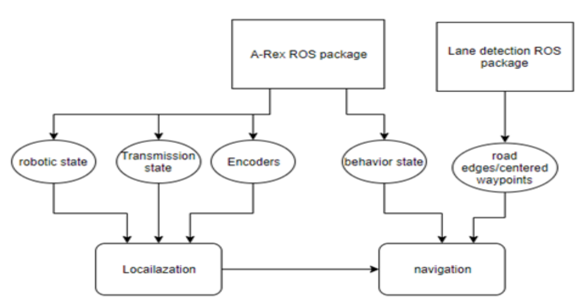

Figure 8: Layout of ROS Packages and Communication

this package, a vision package is being used to publish array markers of the positions of the lanes. These will act as road boundaries, or add immediate waypoints that are centered between the lanes. The way this information is being transferred is shown in Figure 8.

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. Furthermore, the device communicates to the main computer via an ethernet connection. RTK natively supports this sensor so there was little modification that needed to be done 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 9 shows a flowchart detailing the entire obstacle-avoidance process.

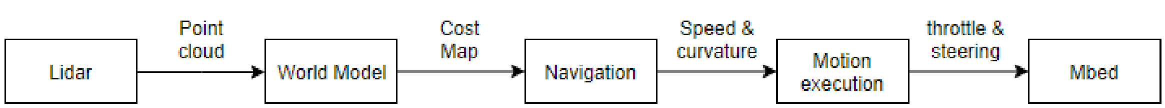

Figure 9: Flow Chart of Obstacle Avoidance Decision-Making.

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 LiDARs 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 actually 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 10 depicts a flowchart illustrating the logic flow from user input GPS waypoints to output throttle and steering commands.

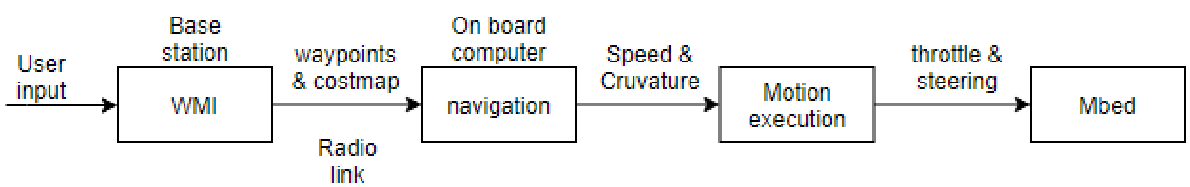

Figure 10: Flow Chart of Path-Planning and Autonomous Decision-Making

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 1.

Vehicle Specification Table	
Engine & Drivetrain	
Battery Voltage	48V
Drive	Direct Front Wheel
Motor Size	6.7 HP (5.0 kW·h)
Motor Type	AC Induction
Top Speed	25 mph
Dimensions	
Cubic Feet of Cab	70 ft^3 (2 m^3)
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
Suspension	
Front Suspension	MacPherson Strut - 5.6 in (14 cm)
Rear Suspension	Independent Trailing Arm - 6 in (15 cm)

Table 1: Polaris GEM e2 Vehicle Specifications

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 or by remote control.

[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.

To help select and locate the sensors, a Simulation Cuboid Model, shown in Figure 11, 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." The Simulation Cuboid Model is a MATLAB tool, which 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.

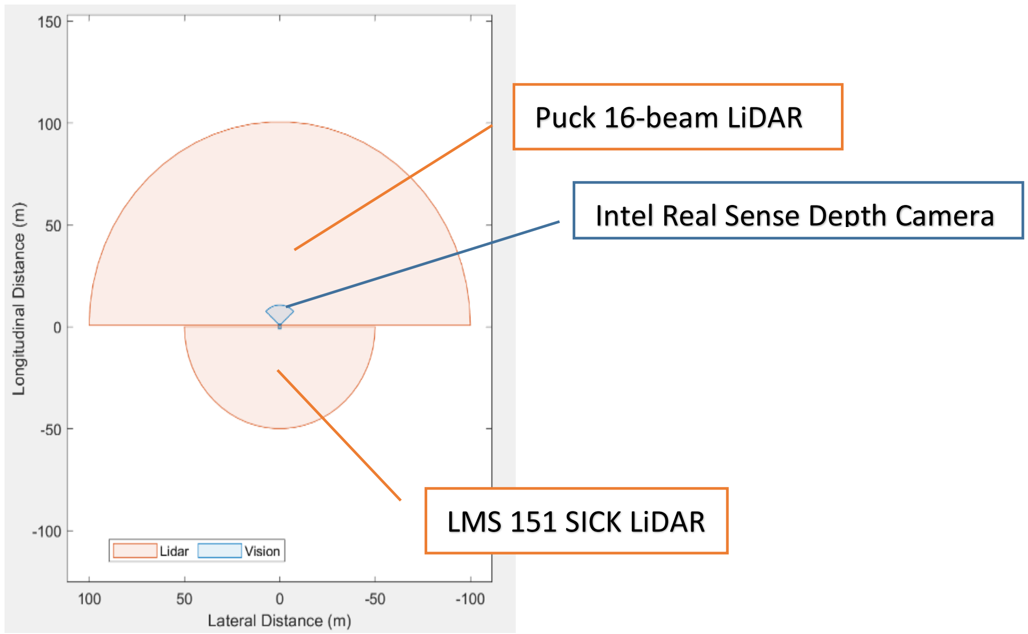

Figure 11: MATLAB Simulation Cuboid Model

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 12. 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 12: Sensor Bar Mount

5.2. Emergency Braking System

The emergency braking system was a modification of the stock-issued parking brake. When the emergency brake is initiated, the rear wheels brakes are applied, which brings the vehicle to a safe stop.

To engage the previous emergency brake, the passenger would pull the handle. With the new design, when the vehicle loses power, the parking brake will be activated by a spring with stored potential energy.

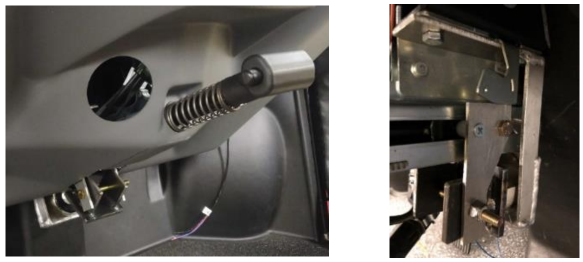

Figure 13: Stock-issued Braking Mechanism

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.

The electrically actuated service brake was implemented by cutting a hole in the firewall so a cable could be attached to the brake pedal and then to a linear actuator, as shown in Figure 14. The linear actuator can apply a force to pull the brake pedal back, resulting in the vehicle either slowing down or coming to a complete stop. Only 25 pounds of force is needed to fully press the brake pedal down, and the linear actuator is rated at 112 pounds of force. The braking system was designed so that if a passenger felt unsafe, the brake pedal could still be used as if the vehicle were in a manual driving state and would override the software. If power is cut to the vehicle, then the parking brake will be activated.

Figure 14: 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 Figure 15. The exterior buttons are mounted on the existing vehicle track system (Figure 15, right) 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 either the one 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.

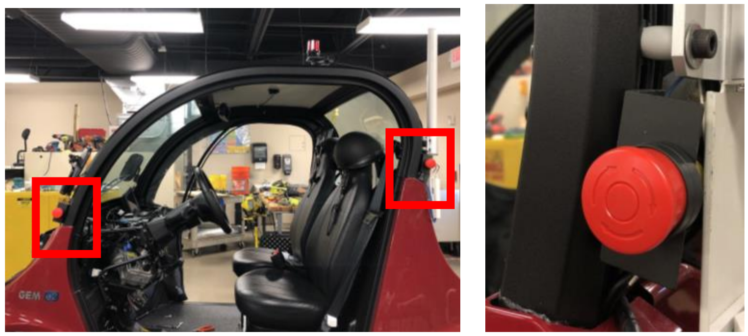

Figure 15: Emergency Stop Buttons

5.3. Drive-By-Wire Conversion

To convert the vehicle to a drive-by-wire system, steering, throttle, service braking, gear shifting and turn signal functions were first analyzed. Figure 16 details the Drive-by-Wire functions and their relationships.

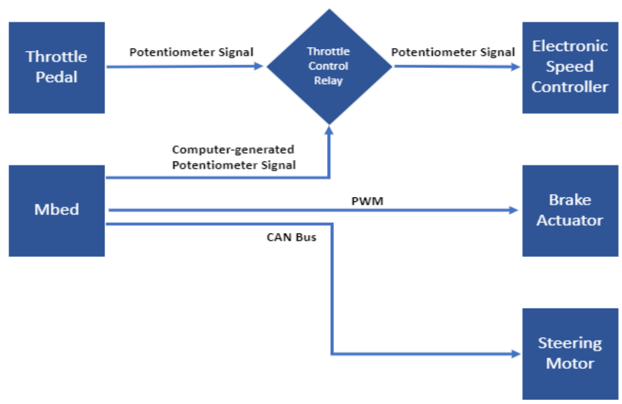

Figure 16: Drive-by-Wire Structural Outline

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 Controller Area Network (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, 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. After successfully sending messages to the power steering actuator a relationship between the vehicle's radius of curvature and the steering wheel position was developed for further use in autonomy. Once the relationship was developed it was then implemented into the CAN messaging code on Mbed. The main computer on board the GEM e2 sends a desired turn radius to the Mbed; the Mbed then sends the CAN signal associated with that turn radius.

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.

5.4. 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 17: Internal Top View of Main Computer Case

After setup, testing was conducted to ensure all electrical and thermal calculations were valid. It also allowed for verification that the all the wiring installed worked as designed. After reviewing with the team, the decision was made to add separate switches to control various sections of the electrical system. One switch for the sensing system component case power, another for the drive by wire system component case power, and a reset switch allowing for the vehicle to be reset without shutting it off.

6.2. Power Distribution System

Figure 18: Buck Converters

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.

6.3. Electronics Suite Description

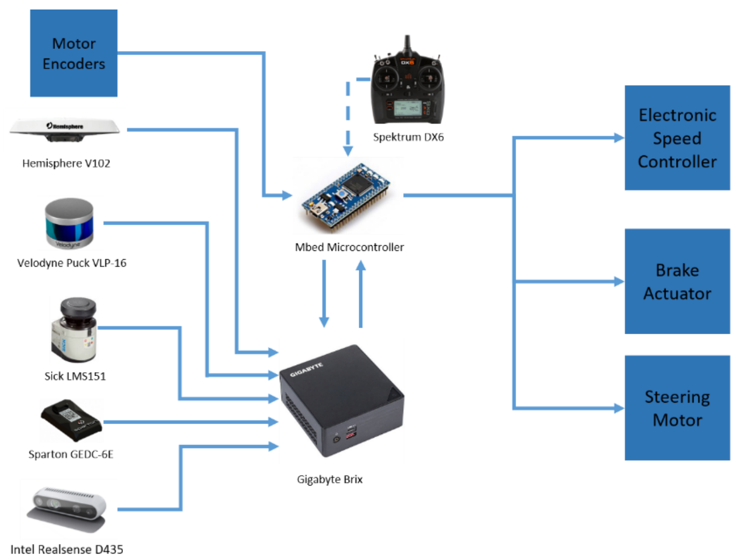

Figure 19: Electronic Suite and Functions

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 more easily manage the data coming 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.

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 that was similar to the IGVC's. 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, doors were closed, helmets were worn securely, and the testing cite was cleared.

Before physically making the sensor mounts, it was first created as a SolidWorks model and the stresses and strains on the mounts could be measured and adjusted using ANSYS before making the first prototypes. This ensured that the mounts could withstand the forces and vibrations when the vehicle is moving.

The subsystem tests were performed before each time the vehicle was tested outside to ensure no problems had arisen between tests. This allowed for problem-free tests of the software. Each software algorithm was tested, including waypoint following, straight path planner, obstacle avoidance, and autonomy. During testing there was always a team on standby incase anything went wrong with the planned test.

8.0 Simulations

Figure 20: Front LiDAR, Camera, and GPS Views from ANVEL, using

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.

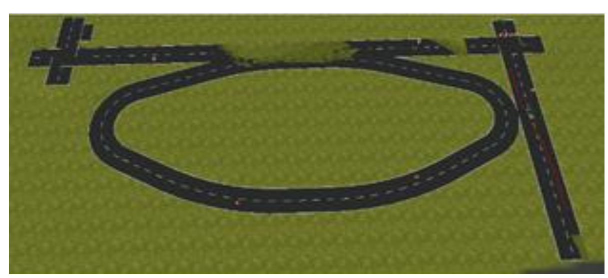
Figure 21: Recreated 2017 IGVC Self-Drive course

Once the sensors in the simulation were coded to generate data (see Figure 20), the complete vehicle was set on a reconstructed course, as shown in Figure 21. 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.

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. Figure 22 shows an example set up of obstacle avoidance testing, where the vehicle is being encouraged to drive between two orange traffic barrels. The last

Figure 22: Obstacle Avoidance Testing

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 the 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.