Bob Jones University Robotics Team
Bruin 3

Date Submitted: May 18, 2021

Team Captain:
Erick Ross | eross156@students.bju.edu

2020 Team Members:

Michael Crum | mcrum425@students.bju.edu
Jonathan DeGirolano | jdegi461@students.bju.edu
Zach Kilpatrick | zkilp142@students.bju.edu
Jonathan Layton | jlayt118@students.bju.edu
Charleton Musselman | amuss590@students.bju.edu
Matthew Palermo | mpale711@students.bju.edu
Joshua Barr | jbarr997@students.bju.edu
James Emery | jemer391@students.bju.edu
Douglas Flynn | dflyn187@students.bju.edu
Daniel Zhuang | dzhua133@students.bju.edu
Joshua Heinrich | jhein429@students.bju.edu
Erick Ross | eross156@students.bju.edu

Faculty Advisors:
Bill Lovegrove | blovegro@bju.edu

Statement of Integrity:

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.

Faculty Advisor: Bill Lovegrove, PhD
Professor
Department of Engineering
Bob Jones University

Faculty Advisor Signature: Bill Lovegrove Date: 5/14/2021

INTRODUCTION

Students from Bob Jones University have worked to implement and integrate several subsystems to transform a Polaris GEM e2 vehicle into an autonomous vehicle named Bruin 3. This has required work from many different fields of engineering to design the different parts of the vehicle. Students from BJU have been working for several years to transform this vehicle to have fully autonomous capabilities. In 2018 and 2019, students redesigned most of the hardware on Bruin 3. In 2020 and 2021, students added several new components to the vehicle, then focused on getting all the software working. This report focuses on the designs on Bruin3 over 2020 and 2021.

ORGANIZATION

Table 1 lists members of the project for 2020-2021 and the number of hours contributed to the project for a total of 600+ hours.

Table 1. Team Members and hours worked.

Name	Hours
Michael Crum	28+
Jonathan DeGirolano	28+
Zach Kilpatrick	28+
Jonathan Layton	28+
Charleton Musselman	28+
Matthew Palermo	28+
Joshua Barr	28+
James Emery	28+
Douglas Flynn	28+
Daniel Zhuang	28+
Erick Ross	320

DESIGN ASSUMPTIONS AND DESIGN PROCESSES

In 2018 and 2019, students designed most of the hardware and mechanical components of the vehicle. Thus, much of the basic mechanical design for Bruin3 was already in place from the progress made in past years. The design process for these components is covered in detail in the 2019 IGVC design report, and many of the mechanical details are parallel in this report. This old design had a few unresolved problems. So, the 2020 team designed Bruin 3 in order to address these issues. They worked to design a dual braking system, an autonomous transmission control, and a CAN bus speed sensor. They also designed corresponding ROS (Robot Operating System) nodes to control those components. In addition, the 2020 team worked to redesign all of the software on the vehicle, and the main design of the software is almost entirely new. The team attempted to completely transition over to the Robotics Technology Kernel software. This design process met with several challenges. RTK is very poorly documented, and the team did not have access to the source code. As a result, much of the design and protocols for RTK had to be reverse engineered.

INNOVATIONS

Dual Braking System

This vehicle implements a dual braking system in order to reduce stress on the hydraulic actuator. Initially, the vehicle had a brake actuated by a hydraulic pump. But in some cases, the vehicle is stopped for a longer period of time, and there is no need to continually apply the hydraulic brake. So, we designed an additional parking brake. When the vehicle detects a long stop, it switches from the hydraulic brake to the parking brake.

Speed Sensing (Optical Flow Odometer and CAN Speedometer)

Typically, we would use rotary wheel encoders to determine the speed of the vehicle. But Bruin 3 does not have any wheel encoders, so we had to come up with some replacement sensors to determine speed. First, we used a PX4 Flow Optical Odometer to provide feedback on the speed of the vehicle. Mounted between the two rear wheels, the odometer determines the speed of the vehicle by capturing images of the ground, rather than with a rotary encoder on the wheels as is traditional. We also were able to design a CAN bus to obtain speed measurements. This CAN bus connected to a speedometer, and we were able to publish that speed information on a ROS topic.

RTK

The main innovation on Bruin3 is the use of RTK (Robot Technology Kernel) software. This software was provided by United States Army CCDC Ground Vehicle Systems Center (formerly TARDEC) and integrated into our vehicle, as part of a grant to help develop RTK. The 2019 vehicle bypassed large portions of RTK and used many custom nodes and hacks. The current vehicle fully uses RTK with a minimum of custom nodes. RTK works alongside the software designed by the team members, but the difference between them is that RTK is the brain that tells the vehicle what to do. According to the information or input that comes in, RTK decides what the next action should be and then the system designed by the team sends signals to the various parts that control the vehicle. This software is not well documented, and it is innovative to even get the software integrated into a vehicle. Several components of RTK were designed for specific sensors, which were not available on Bruin 3. So we had to redesign much of the low-level topics coming into RTK, and simulate some others that were not being produced. In addition, the entire Drive-By-Wire system within RTK had to be redesigned in order to work with our vehicle and its actuators.

MECHANICAL DESIGN

Overview

The rear of the vehicle is divided into two sections, the bottom rear and the top rear. The bottom rear of the vehicle contains the Hydrastar, the PX4-Flow optical odometer, and the back-up battery. A metal plate sits on top of the back-chassis frame and supports the three pieces of equipment. The top rear section contains the two CPUs (Fanny and Freddy), a 16-port ethernet box, two power converters, the IMU, and the Lidar ethernet box.

The following sensors are located on the top of the vehicle: LIDAR, Mako camera, GPS unit, and a light beacon. These are attached on top of a metal plate that is secured with brackets via the T-slot feature on the sides of the car.

The front of the vehicle contains the RADAR and the stereo camera. The RADAR is mounted to the front, diagonal beams of the frame. The stereo camera is attached to the vehicle right below the windshield in the charging port area.

Drive-by-wire kit

The team designed and installed their own drive-by wire-kit allowing the computer to control the vehicle's steering, brakes, and acceleration pedal.

The system consists of three actuators. The brake actuator is a pump called the Hydrastar. This pump is in charge of pushing hydraulic fluid into the brakes. An electric stepper motor turns the steering wheel. Finally, an electronic accelerator pedal interface was created by the team that produces the same signals as the accelerator pedal from the factory car.

Suspension

The 2018 GEM e2's front suspension is a MacPherson strut and the rear suspension is an independent trailing arm. No changes were made to the vehicle's stock suspension.

Weather proofing

Bruin 3 must be protected from the weather. Concerning the sensors, the three cameras, LIDAR, and Radar, GPS unit, and IMU are unprotected. The enclosed sensors are the DAC, PX4Flow odometer, and the steering equipment. In the back, a waterproof box protects the vehicle's batteries and various electronics. Soft doors and a back window were purchased to protect the rest of the vehicle.

Parking Brake

For the design of Bruin 3, we wanted to be able to control the parking brake through a ROS node. We did this using a RoboteQ actuator. This design for the parking brake required a connection between the actuator and the parking brake. The final actuator housing design is a vertical square aluminum and carbon fiber shell that slides over a protrusion in the floor of Bruin 3 as shown in Figure 1.

Figure 1: The Parking Brake Actuator with its Housing

The base of this housing has two horizontal flat pieces extending out from the two vertical walls to create broad feet on which the entire weight of the housing and motor sit. On the interior of these vertical walls there are 4 separate pieces which locate the actuator in relationship to the outside walls both horizontally and vertically.

DESCRIPTION OF ELECTRONIC AND POWER DESIGN

Overview

The vehicle uses a 48V Battery Pack as power source and various converters that power the computers and sensors. Additionally, a 12V battery pack serves as a backup battery for the HydraStar braking system.

The vehicle consists of three main computers and sensors. The sensors include cameras for obstacle detection and localization sensors.

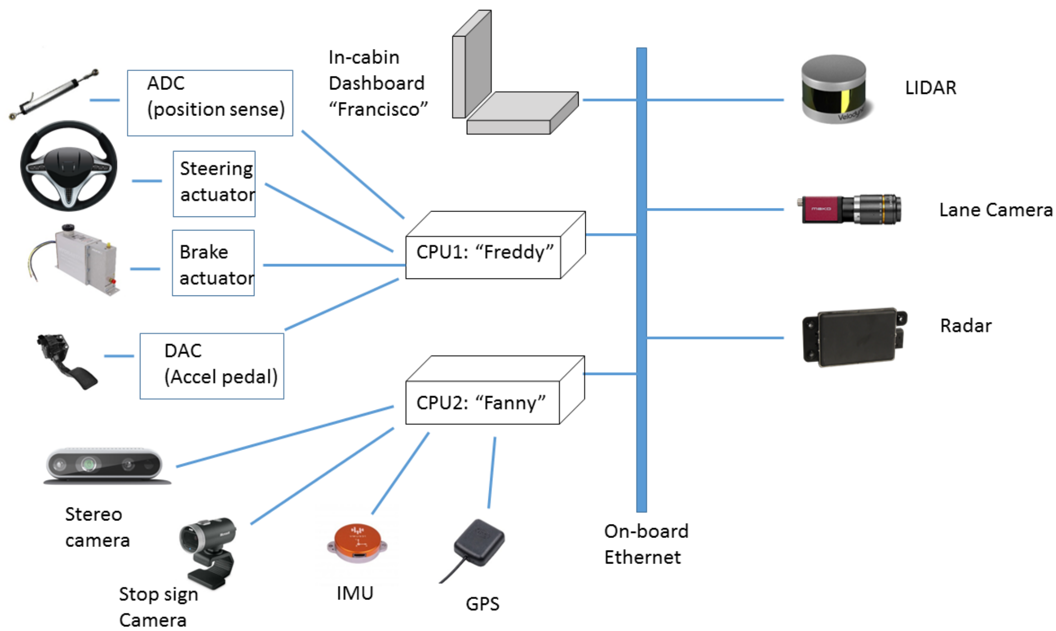

Figure 2. Block Diagram of the Control System.

Power Distribution System

There are two 48-to-12V DC-to-DC converters from which the computers are powered. The computer that processes images, "Fanny," requires a 12V power supply and draws 10A. All the cameras are connected to this computer. The sensors require 12V power supply and draw 3A. The other computer, "Freddy," receives feedback from the other sensors and sends signals to the motor controller and actuators which requires a 12V power supply and draws 5A as seen in Figure 3.

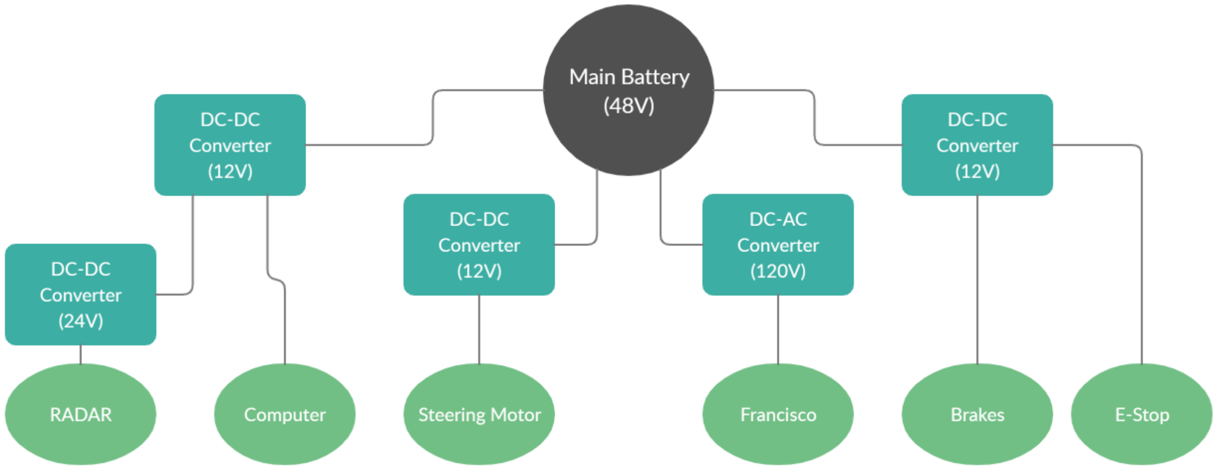

Figure. 3 Power Distribution Diagram.

Electronics Suite Description

Computer Hardware

1. *Freddy* is a LINUX PC that runs the actuation nodes.
2. *Francisco* is a LINUX laptop that runs the high-level RTK nodes and provides a software dashboard in the cab of the vehicle.
3. *Fanny* is a LINUX PC that runs sensors and localization nodes.
4. *Whyme* is a Windows laptop that runs the WMI (Warfighter Machine Interface) and can be used by the vehicle's occupants or a remote operator.

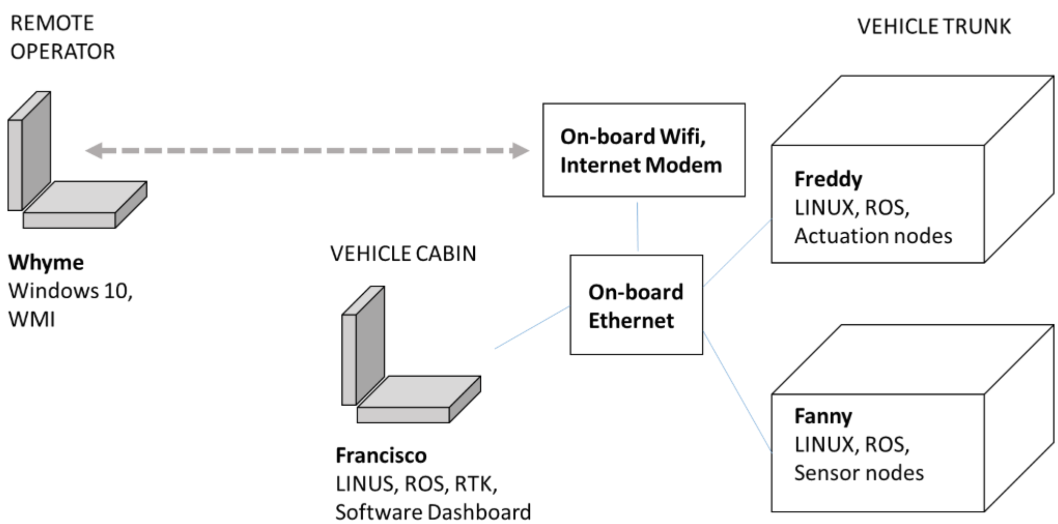

Figure 4. Computer Hardware Connections.

Sensors

1. Cameras
    a. Lane detection camera
    b. Stereo camera
    c. Road sign detection camera
2. LIDAR
3. RADAR
4. Localization sensors
    a. GPS sensor
    b. Odometry sensor
    c. IMU (Inertial Measurement Unit)

Figure 5. Sensor locations in the vehicle.

Safety Devices

Fuses are inserted between the DC-to-DC converters and computers to prevent short circuits. An E-Stop system is also implemented in the case that the vehicle must be shut down immediately. A backup battery is installed in the HydraStar's braking system in the case of failure.

CAN-bus Speed Sensor

For Bruin 3's speedometer, we used a Kvaser Leaf v2 to collect and process Bruin 3's CAN bus speedometer data. We implemented Kvaser's Linux drivers to receive the Leaf's CAN data stream. The Kvaser then reads and isolates Bruin3's speed from the stream and sends it to a publishing ROS through a Linux FIFO pipe. Once the publishing ROS node receives Bruin3's speed data from the pipe, it translates the scalar speed values to an odometry message and publishes it to the RTK topic /localization/speed.

Autonomous Transmission Control

Forward-Neutral-Reverse mode was previously controlled through a mechanical switch. Each mode was activated by connecting different ports to create a short circuit to ground. Each of the voltage probes represents a wire that tells the motor to be in reverse mode, neutral mode, or forward mode. Utilizing this information, our initial design consisted of taking 8 switches and using them to control which ports were connected to induce the same logic automatically instead of manually. This idea was inspired by the fact the internals of switch was not fully understood. However, after finding a schematic of what was going on inside the switch and the car, we were able to reverse engineer it, as shown in figure 1. The two switches on the left indicate the Forward-Neutral-Reverse (FNR) switch in the car. The 3 voltage probes on the right side indicate the 3 sensor wires to indicate which gear we are in. When the sensor wire is at 0 volts, then we are in that gear. From top to bottom the sensor wires indicate Reverse, Neutral, and Forward gears. Using this information, we were able to control the transmission autonomously through a relay board. So we designed a ROS node to change the transmission, based on what RTK requested.

SOFTWARE STRATEGY AND MAPPING TECHNIQUES

Overview

Our software is based on RTK (Robotics Technology Kernel) which is built on top of the ROS (Robot Operating System) framework which is an open-source software package maintained by the Open Source Robotics Foundation (OSRF). In ROS each major function is managed by a separate smaller program called a node. The nodes communicate between each other and perform the various tasks needed for our vehicle to function. RTK is a compilation of ROS nodes managed by GVSC, the Army CCDC Ground Vehicle Systems Center. Our team wrote several custom ROS nodes to implement different features.

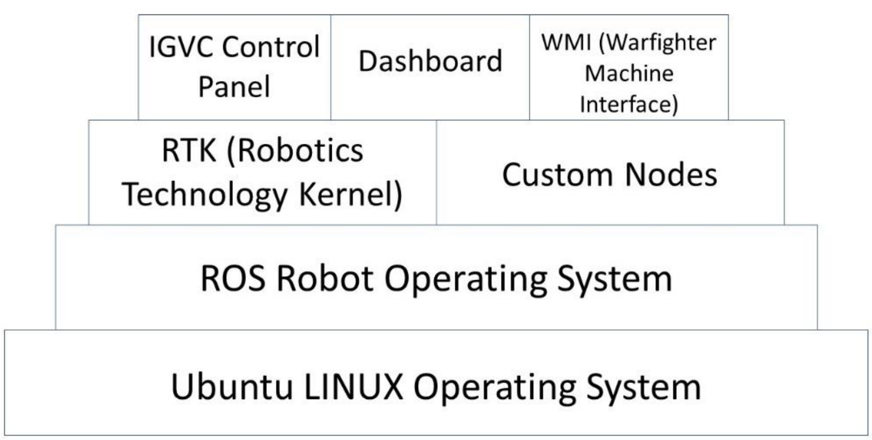

Figure 6. Software Architecture.

Much of the standard RTK system had to be modified in order to get it working on Bruin 3.

First of all, the localization subsystem of RTK receives sensor inputs from the Sensor subsystem. RTK expects a standard localization sensor suite, including a specific Novatel GPS, Microstrain IMU, RTD rear wheel encoders, and KVH single axis gyroscope. We did not have all of these hardware components available, so the sensor part of RTK had to be entirely rewritten. We had a Garmin GPS, and used the GPS driver gps_umd from SWRI Robotics to generate sensor messages. In addition, we used the imu VMU931 from Variense. Variense provides driver code that we used to generate sensor messages from the imu. We also used this driver to get gyroscope data. Bruin3 does not have wheel encoders, so we used a px4flow camera to get speed measurements. This camera scans the ground and detects how fast the ground is moving relative to the camera and uses that data to calculate speed. We used the px-ros-pkg/drivers/px4flow ROS node to obtain speed measurements. Most of these sensor messages were not formatted correctly, so we wrote an additional ROS node to transform each of these topics into the correct format.

Second, RTK has an IOP Bridge subsystem connecting to the OCU (Operator Control Unit). We used the standard OCU for RTK: the WMI software (Warfighter Machine Interface). The IOP Bridge and WMI use JAUS (Joint Architecture for Unmanned Systems) messages to communicate information back and forth. We ran into some networking issues to connect RTK up to WMI, but we were able to configure the network in such a way to allow communication.

Third, the motion execution and low-level CAN subsystems had to be rewritten. These are the DBW (Drive By Wire) system that RTK uses to control the low-level actuators on the vehicle. In standard RTK, the motion execution system sends DBW commands to the CAN Bridge to control the vehicle. The actuators on Bruin 3 are very different than the ones that RTK expects, so we had to completely redesign the DBW system. We took the four commands from the Motion Execution System: brake, throttle, steering, and transmission inputs, and wrote our own ROS nodes to implement those commands with the actuators on Bruin3. RTK is also designed to expect several messages from these low-level components indicating that everything is working fine. We had to fake many of these messages with a stub in order to make the RTK system happy.

Finally, there were several other details in the main part of RTK that we had to change in order to get everything working properly. One of these was the VMS (Vehicle Management System). It was expecting a lot of messages from different parts of the RTK system that weren't being published. So we had to change some of those things and fake different topics in order to get the VMS to do its job.

Obstacle Detection and Avoidance

We used three major components for obstacle detection: a stereo camera, LIDAR, and RADAR.

1. LIDAR is used to create a 3D map of the area around the vehicle as seen in Figure 8.
2. RADAR senses certain obstacles in front of the vehicle, for example, pedestrians and other vehicles.
3. The stereo camera detects obstacles that are in front of the vehicle as seen in Figure 9.

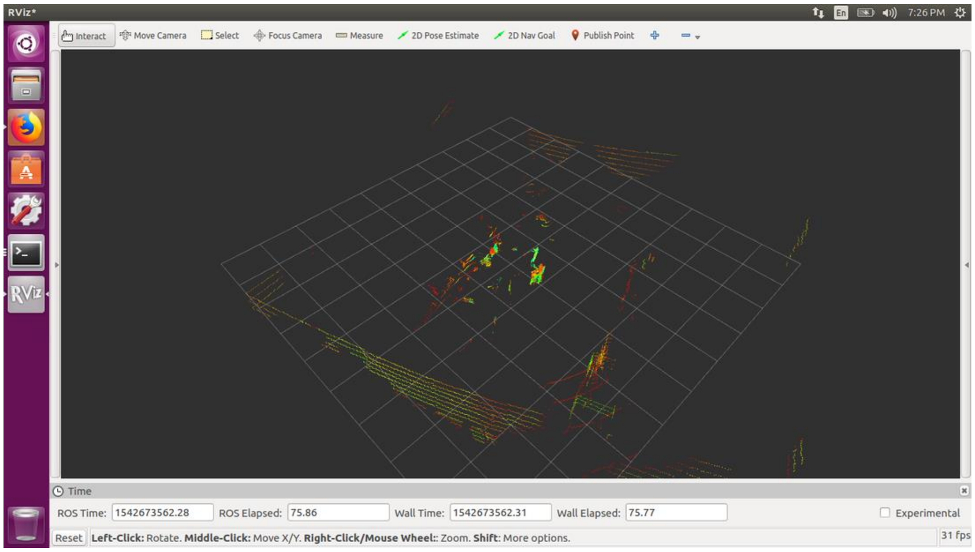

Figure 7. LIDAR sample.

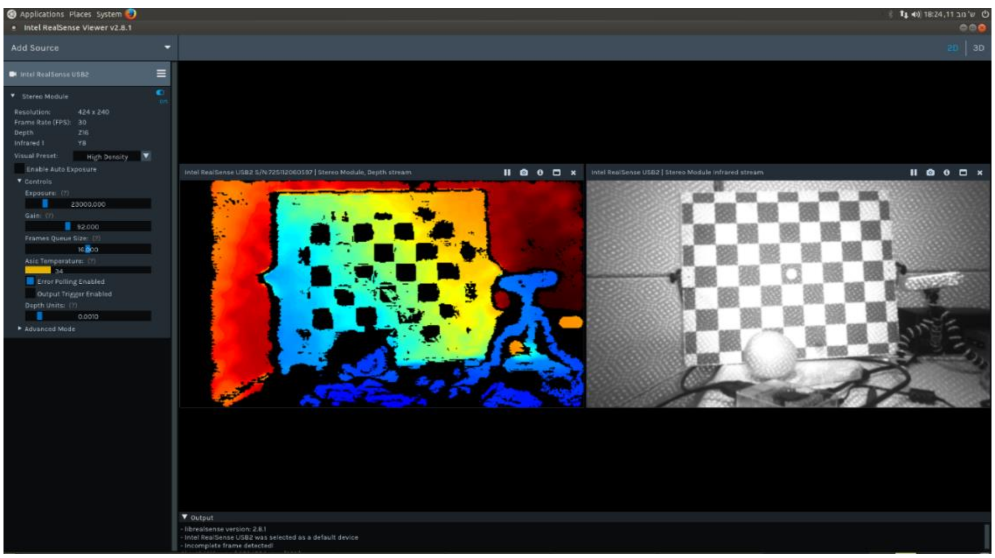

Figure 8. Sample depth image from the RealSense stereo camera.

Lane Following

A camera on the top of the vehicle scans for road marking lines. These marking lines are turned into 3D obstacles and added to the cost map so that the path planning will stay between them.

Map Generation

The world model module of RTK combines data from the LIDAR, stereo camera, and other sensors to generate a map of the world around the vehicle with the obstacles and other parts of the course. This world model then generates the costmap for the entire situation. The generated costmap indicates the riskiness of different paths the vehicle can take. See example cost map in Figure 10.

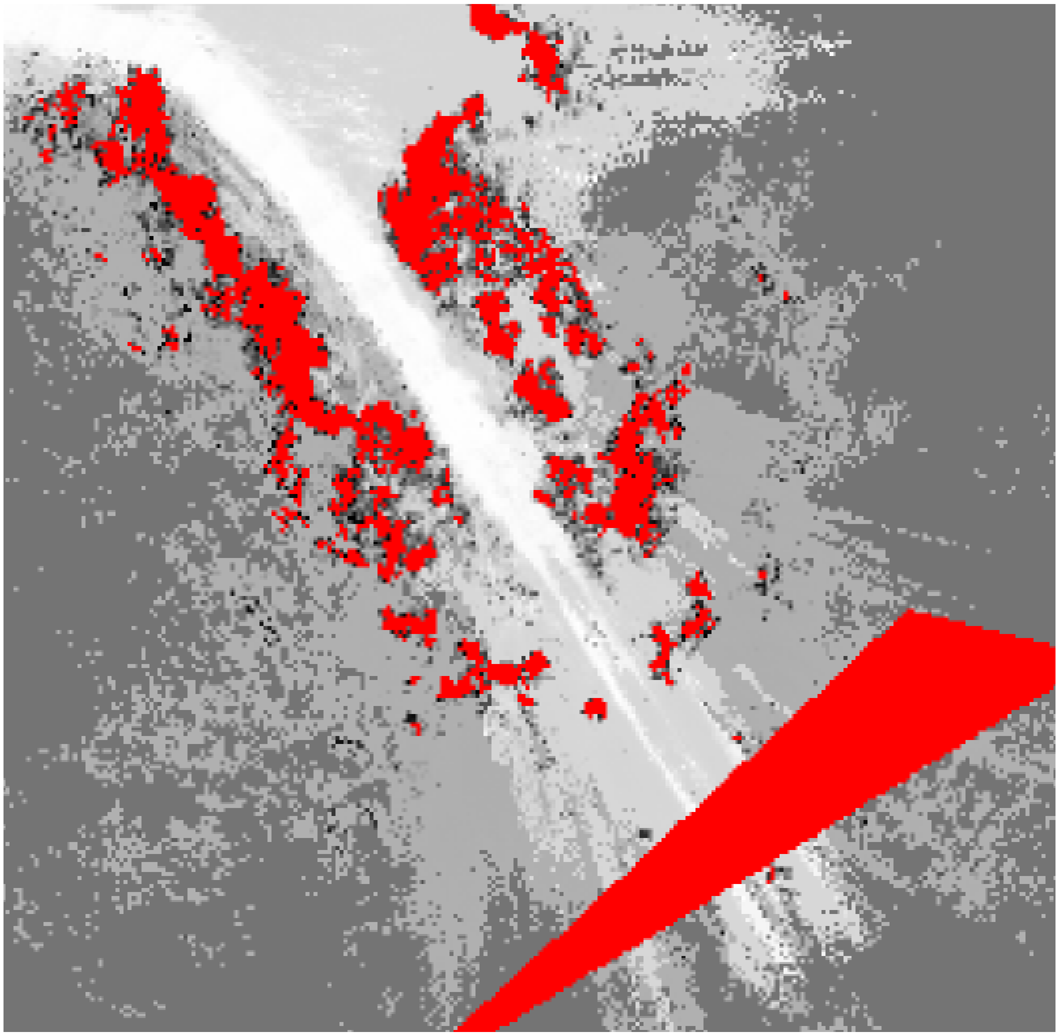

Figure 9. Example of a costmap generated by RTK.

Path Generation

The path planning module of RTK uses the costmap and the A* algorithm to find the path of least total cost. The RTK module uses the sensor information to determine where the vehicle cannot go, like going off the road or crashing, and associates that with a very high cost. The clear road ahead of the vehicle will be assigned a very low cost and that is the path that the vehicle will take. The vehicle will then use this path that RTK plans to drive the motor and steering of Bruin 3. There will be a user interface where the user can provide their desired destination. The vehicle requires GPS to perform waypoint navigation. When the next point is entered or identified, the system then uses the sensors to ensure all obstacles are avoided. The vehicle stays in the lane while the GPS indicates the location to direct the system to head in the right direction.

DESCRIPTION OF FAILURE MODES, FAILURE POINTS AND RESOLUTIONS

Vehicle failure modes and resolutions

If the steering actuator subsystem fails, the vehicle may attempt to drive into obstacles. The obstacle sensors should detect the obstacles. It will attempt to steer around the obstacles, without success, and then stop the vehicle when it becomes clear that a viable path is no longer available. The human safety driver is also responsible to observe the operation of the vehicle and intervene if the path is toward an obstacle.

If the brake actuator subsystem fails, we are dependent on the human safety driver to stop the vehicle using the independent front wheel braking system. In the unlikely event of failure of both brake systems, the parking brake may be used by the human safety driver to stop the vehicle.

If the accelerator pedal actuator fails, the vehicle may accelerate out of control. The obstacle detection systems should intervene and attempt to brake the vehicle. If full acceleration and braking are both actuated at the same time, the brakes will be able to stop the vehicle but at a reduced rate. The human safety driver may need to intervene in this situation as well.

Vehicle failure points and resolutions

If the battery fails, we are dependent on the back-up battery. This battery will continue powering the vehicle estop which will be triggered by loss of power.

If the actuators fail, we are dependent on the e-stop to stop the vehicle, in order to avoid further complications.

If the e-stop communication fails, this will cause an e-stop.

If the communication between the computers fails, the vehicle will stop driving.

All failure prevention strategy

The vehicle health system can detect multiple failure points across the vehicle, and it will stop driving.

The vehicle operation currently requires a human safety driver in the vehicle at all times. The human operator can engage the estop at any time. The operator can also brake or steer the vehicle manually.

Testing

The estop system was tested and the stopping distance was measured at 11 feet at 5 miles per hour. It also engaged when a wire was disconnected from the e-stop system.

The wireless estop system has a specified range of 600 feet, well beyond the IGVC requirement of 100 feet. This range has yet to be tested but will be tested before IGVC.

The torque required to override the steering actuator was measured to be 5 lbs. at 5.5 inches or 2.3 foot-pounds (3.1 Nm) of torque, which is easily achievable by the human safety driver.

The manual brakes were tested and are fully functional in autonomous mode.

The vehicle and our on-campus test track are modeled in the Gazebo simulation environment. The vehicle can be driven in the simulated environment using the same software as the real vehicle. The resulting vehicle trajectories can be compared. See section 8 for simulation details.

The actual vehicle was tested on a grassy field on the BJU campus. On March 12 we successfully demonstrated driving to a waypoint as seen in Figure 11. See section 9 for testing details.

Vehicle safety design concepts

The Polaris E2 vehicle meets all the safety standards for a low speed electric vehicle (LSEV) including headlamps, tail lamps, stop lamps, reflectors, mirrors, a parking brake, a windshield and seat belts. We have not modified any of the safety features except the brakes as described below.

The speed and path curvature are limited by the software to stay within the limits of the vehicle.

The vehicle includes a fire extinguisher as required by the IGVC rules.

Four on-board e-stop buttons and a wireless e-stop provide hardware shutdown of all of the actuators. The estop buttons activate normally-closed switches, so any hardware fault in the system that results in an open circuit causes an estop.

The steering wheel and brake pedal are fully functional in autonomous mode, giving a safety driver capability to control the vehicle manually at all times. With the current vehicle we intend to operate the vehicle only with a human safety driver in the driver's seat.

The HydraStar braking system uses a backup battery to provide positive braking in the event of an e-stop; the vehicle does not coast after e-stop and will stop even in the case of a total loss of primary system power.

The front (manual) and rear (estop) brakes have separate hydraulic systems, so that if either system fails the vehicle can be stopped with the other.

SIMULATIONS EMPLOYED

Simulations in virtual environment

Within RTK's Operator Control Unit, WMI, RTK provides a sample simulation to test out the software and learn how WMI should work. We used simulation to test out driving the vehicle and testing how the RTK modes should work.

The vehicle is modeled in Gazebo for virtual simulation.

PERFORMANCE TESTING

Component testing, system and subsystem testing, etc.

The vehicle has a specified battery range of 20 to 30 miles. In a worst-case battery life test (hilly terrain, high speed stop-and-go driving) the vehicle reached a "low battery" level after 12 miles. This lowers the range between 8 to 18 miles but is more than adequate for IGVC.

The vehicle is able to climb a 6-degree (11%) slope easily.

All sensors have been successfully tested in ROS.

INITIAL PERFORMANCE ASSESMENTS

At the time of completion of this report, the RTK is not fully implemented in the vehicle, and the vehicle is not yet capable of performing the full set of IGVC tasks. All of the subsystems are installed and tested individually.