RoboJackets

Competitive Robotics at Georgia Tech

Georgia Institute of Technology

JAYMI

Date Submitted: 5/15/2017

Noah Daugherty - Project Manager
noahpd@gatech.edu

Contributors:
Noah Daugherty, Kelvin Chong, Yongjae Won, Daniil Budanov, Rouyang Xu, Ryan Waldheim, Fengrui Zuo, Matthew Keezer, Jason Gibson

Faculty Advisor:
Dr. Frank Hammond - fhammond3@gatech.edu

Introduction

RoboJackets

RoboJackets is the competitive robotics organization for students at the Georgia Institute of Technology. Founded in 1999 as a BattleBots team, the organization has grown to include the RoboCup Small Size League, the Intelligent Ground Vehicle Competition, the International Autonomous Robot Racing Challenge, and a large outreach team. Though the organization is chartered under the school of mechanical engineering, our members represent nearly every department on campus, predominantly computer science, mechanical engineering, aerospace engineering, and electrical engineering. Having first competed in IGVC in 2004, the RoboJackets have competed in every IGVC since 2006.

Team Members

Our team is organized into mechanical, software, and electrical subteams. Each subteam contributes designs, assemblies, and code per the requirements of the competition and other subteams. Table 1 shows a listing of our active membership.

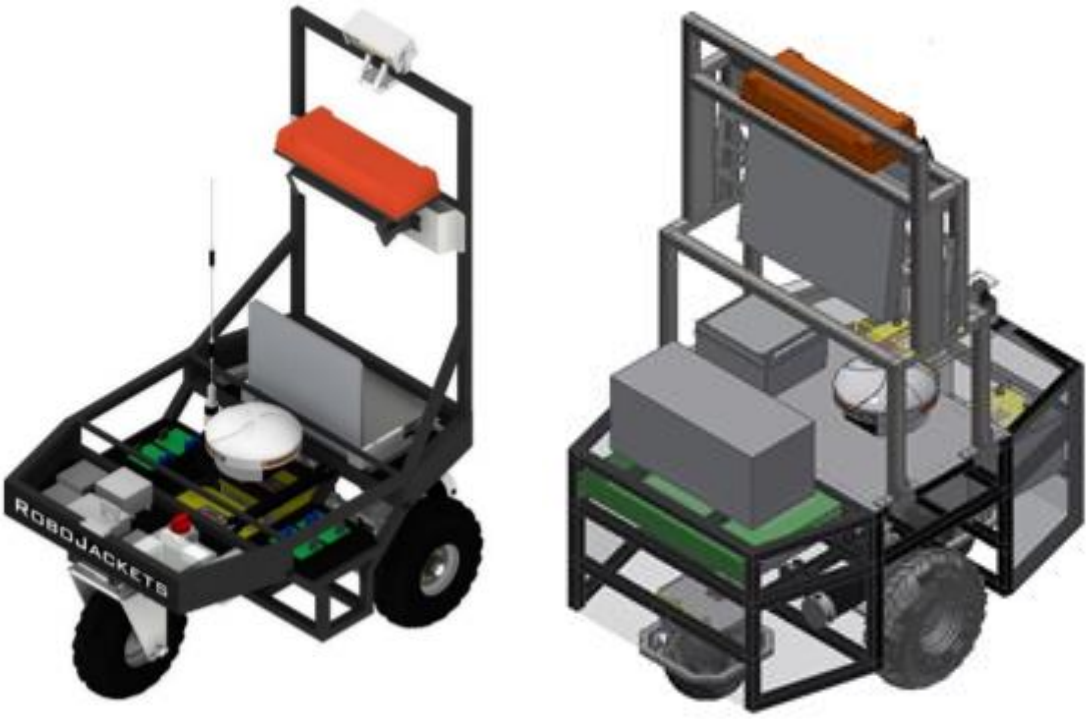
Table 1. RoboJackets 2017 IGVC Membership

Name	Role
Noah Daugherty	Project Manager
Kelvin Chong	Mechanical Lead
Daniil Budanov	Electrical Lead
Matthew Keezer	Software Lead
Ryan Waldheim	Electrical
Ruoyang Xu	Electrical
Fengrui Zuo	Electrical
Choukri Nyon	Electrical
Yongjae Won	Mechanical
Dallas Downing	Mechanical
Jeff McKendree	Mechanical
Tomas Osses	Mechanical
Shane Kearney	Mechanical
Justin Zheng	Software
Jason Gibson	Software
Dominic Pattison	Software
Raymond Ortiz	Software
Jeremy Schonfeld	Software
Kenny Scharm	Software

Mechanical

The design for the 2016-2017 robot focused on creating a whole new machine and chassis that would be able to overcome the issues with the previous machine, Mistii, seen in Figure 1a. Those issues included:

- Imbalance with drive system in the back
- Being caught in potholes due to said system hindering flexibility
- Issues with inconsistent drive performance



Left - Figure 1a. 2015-2016 Robot Mistii
Right – Figure 1b. 2017 Robot Jaymi

Design Premise

The vehicle is designed with an emphasis on keeping loads as close to centered and balanced as possible to avoid those issues with our previous machines. After many discussions and design sketches, it was decided to move in a new direction that RoboJackets IGVC hadn't attempted recently. By placing the drive system in the center of the robot, we can keep the weight distribution from impacting performance as negatively as before.

Chassis

The structure of the robot is primarily composed of 1”x1” square steel tubing and 1”x1” square aluminum tubing. (Figure 2) The lower section of the frame, henceforth referred to as the “main body,” was assembled with the steel through MIG welding, and it was painted over to prevent corrosion. The upper portion, known as the “mast,” was assembled through MIG welding aluminum. This was a key design change from before: in the past, the entire frame was created out of steel. After consulting with the software team, we quickly concluded that a change had to be made because the steel, being ferromagnetic, would cause the GPS to become disoriented.

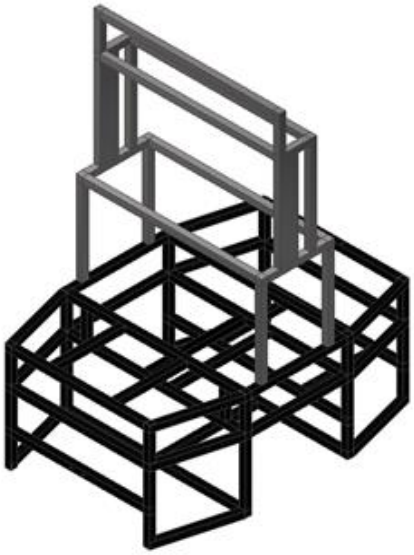
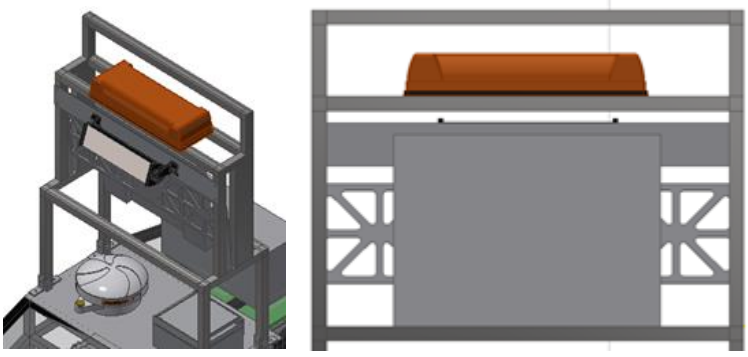


Figure 2. Frame

Because welding aluminum onto steel would be near impossible, the mast is attached onto the body via aluminum brackets instead. The batteries sit on one end while the payload and electronics tray sit on the other, providing a more balanced machine. The drive wheels sit in the center of the body, while our monitor and other components are mounted in the mast. The light and cameras would be on the mast as well, allowing a more center focused approach for our imaging software.



Left – Figure 3a. Mast and Components
Right – Figure 3b. Mast and Monitor

Drive System

This robot's drive system is very different from what has been done in the past. In the center of the machine, directly below the mast, are the 2 drive assemblies.

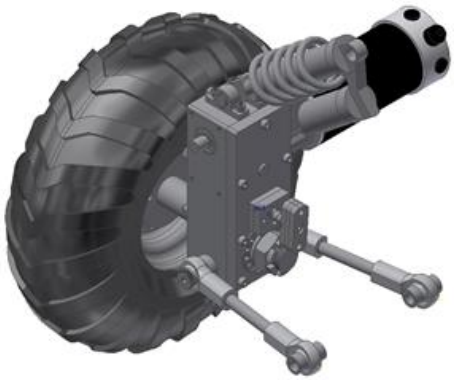


Figure 4. Right Drive System

This is by the far most complicated mechanical system we have designed for our machine. Similar to previous designs, a chain and sprocket system was avoided due to inconsistent performance. This setup has the motor interfaced through a gear assembly that translates the force directly to the wheel. A combination of retaining rings and bearings are used to keep the gears in the proper positions. Keys and keyways were also crafted to allow for our decoupling mechanics, which will be discussed later.

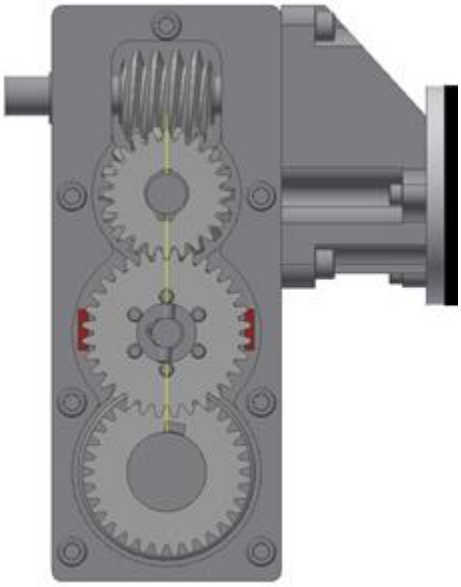


Figure 5. Gearbox and Gear Assembly

The suspension of the machine works through 2 control rods (seen in Figure 4) and a shock mounted directly on top of the gearbox. The control rods were threaded with opposite direction threads on each end, allowing for ease of adjustment.

The wheel is attached onto the axle via connecting plates and keys. Tolerances are tight in order to reduce the amount of wobble that may occur.

Decoupling Mechanism

This year's decoupling mechanism is very different from what was done in the previous assemblies. Instead of having a system on the outside of the wheel, this one was designed with the gears inside the gearbox itself meshing and moving apart.

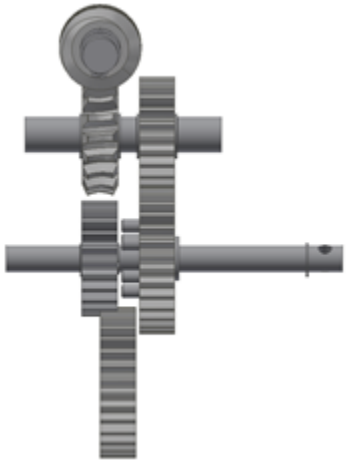


Figure 6a. Gears Decoupled

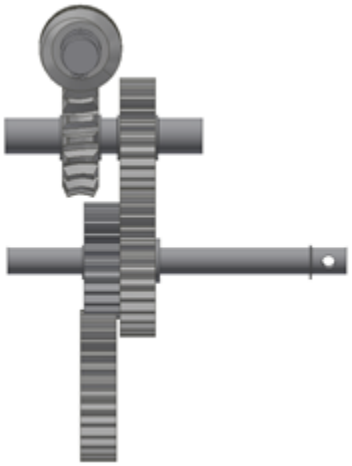


Figure 6b. Gears Coupled

As can be seen in Figures 7a and 7b, the gear in the center has a series of pins on it. These pins mesh directly into a series of holes within the gear in front. The gear with the pins sits on a bearing sleeve, allowing it to move freely on the center axle. The gear with the holes is bound to the axle with retaining rings, so that any movement by the axle will cause it to move as well. Since the motor interfaces and rotates the worm and worm gear, by pushing the bound gear off of the gear with pins, we effectively remove all power to the wheel.

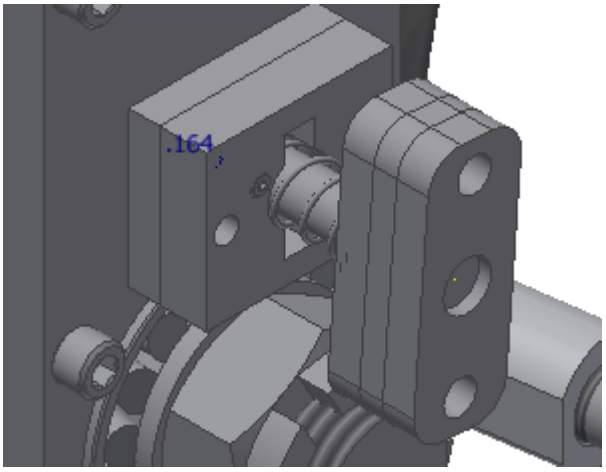


Figure 7. Decoupling Plates

To achieve this kind of behavior, the center axle as noted above has plates attached to the gearbox. A pin sits within the rod, causing it to lock in place depending on how it is twisted. A spring forces the pin to stay within a specific region.

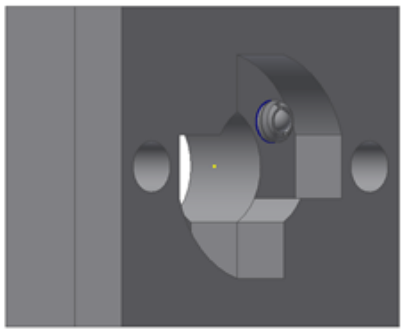


Figure 8. Decoupling Plates, Opposite View

The plates seen attached at the end of the axle in Figure 8 allow for easier turn of the axle. This reduces the danger of reaching under and having to trigger the mechanism manually. Instead, a tool and be used to couple or decouple the machine.

Caster Balls

For this design, a less conventional approach was taken for the design of the forward and backward casters. After many discussions within the sub-team, it was very clear that “caster bias” would be a significant issue. Due to how a pivoting wheel would work, there would be major complications when rotating the machine. To combat this, a caster ball system was developed. These would sit on the front and back ends of the main body.

Figure 9a. Caster Ball Assembly

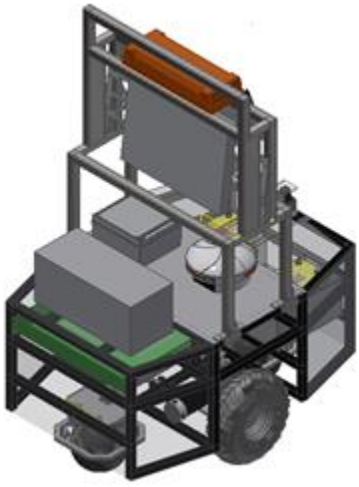


Figure 9b. Overall Machine

A plate sits on top of the ball, and under the plate are ball bearings that allow the ball to spin in the assembly. A ring sits further down, and it also contains a series of ball bearings. A shock is attached to the plate and the frame body. The hinge seen in Figure 9a is mounted on the frame body as well.

There may be issues with the ball coming loose due to high elevated terrain. However, after a few tests, the weight of the robot’s overall body was able to prevent this from happening. Further issues have been mitigated by placing magnets under the plate.

Ease of Maintenance and Safety

While creating a robot that can traverse rough terrain and follow the rules of the competition is important, keeping the machine easy to operate and maintain are also crucial for success. The following is a list of improvements made to facilitate maintenance and safety:

- A slide-able electronics tray was implemented to allow for a simpler method of access
- Folding panels in the cover for easy access to batteries
- Monitor mount onto the mast, reducing the need for software team to bend over to troubleshoot and software issues
- Decoupling mechanism with tool, eliminating need to reach under robot (preventing potential injuries)

Electrical Design

Overview

A major focus of the electrical subteam this year has involved offloading power components to a single, more compact electrical tray and more efficiently organizing the electrical construction of the robot. This allows the power system to remain distant from the sensitive robot sensors. A standardized coloring scheme was created, where every power cable's voltage is assigned a corresponding color. Lastly, a Fritzing diagram (Figure 10) of the robot was implemented, which provides a clear and easily editable illustration of the entire electrical system.

Power Distribution System

Power is supplied by two Optima D35 12V automotive lead-acid batteries connected in series to provide 24 VDC to the system. Battery power is connected to a disconnect, which is then routed to the 24V rail on the Electronics Tray. A cable detaches from the connection node on the rail to connect the raw battery voltage through the normally-open e-stop solenoid to the Open Source Motor Controller (OSMC).

A 24-12V PYLE DC-DC converter attaches to the 24V rail and supplies power to devices connected to the 12V rail, including a light, the NUC computers, the LiDAR and the GPS. A separate set of XL4005 Buck DC-DC converters provides power to the onboard monitor. A LM7805 5V voltage regulator powers the 5V rail.

Ultimately, all return connections are run through a shunt before being connected to the negative battery terminal. Measuring the voltage drop permits calculation of the instantaneous current through the system, which in turn allows the onboard battery monitor to estimate the amount of charge remaining on the battery.

Every device is connected through an appropriately rated fuse to its respective voltage rail. Rails, voltage regulators, and motors/light controllers are all mounted on an electronics tray capable of sliding in and out of the body of the robot, whereas sensors and actuators are powered through extended wires, all run through wire guards attached to both the tray and robot frames.

Four wire guards allow us to group wires carrying like currents together, thus limiting interference across wires.

A color standard is developed for all cables providing power to the robot. This has given us a much clearer, and therefore more easily repairable, electrical system.

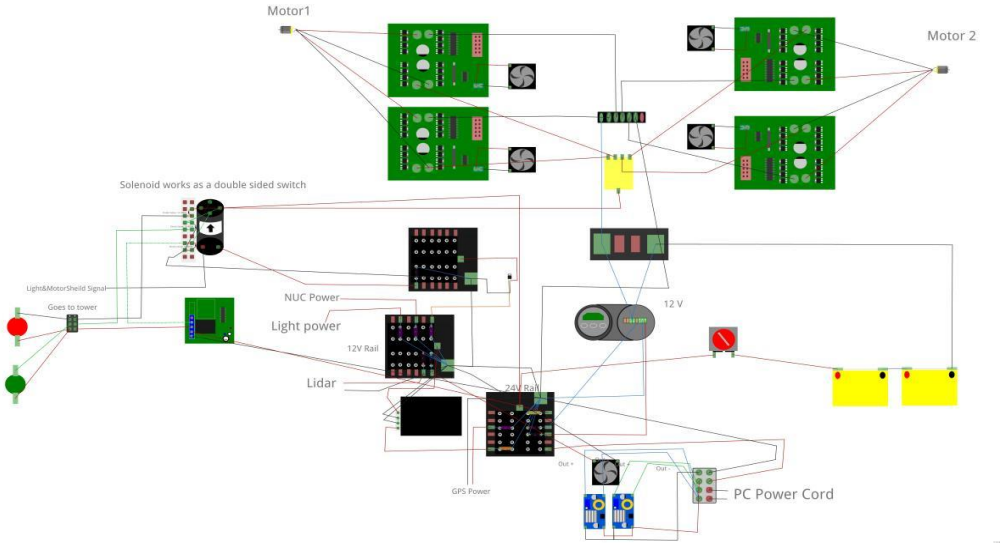


Figure 10. A *Fritzing* diagram of the electrical system.

Electronics Suite Description

Light Detection and Ranging (LiDAR)

A LiDAR is a laser scanning device that identifies objects in a certain area and helps the robot to map the course. The LiDAR on Jaymi is the TiM551-2050001 device by SICK AG. It can scan a 270-degree, 0.05 to 10 m circular sector area, identify objects within and return the distance to the objects. The data acquired from our LiDAR is then sent to the onboard NUCs for global mapping and path planning.

Global Positioning System (GPS)

An Outback A321 Dual Frequency GPS unit is placed in the center-left of the robot. It takes in both GPS and GLONASS signals and utilizes RTK (Real Time Kinematics) to enhance the performance of the positioning system. The GPS data is directed into the NUCs through a USB-to-Serial converter to help set waypoints for path planning.

Inertial Measurement Unit (IMU)

An IMU measures the robot's specific force and angular velocity. Jaymi is equipped with a 9DoF Razor IMU by Sparkfun. There is an ITG-3200 (MEMS triple-axis gyroscope), a ADXL345 (triple-axis accelerometer) and a HMC5883L (triple-axis magnetometer) integrated in this model and thereby can give 9 degrees of inertial measurement. We use the data from the IMU to assist in processing data for global mapping and to find the robot's pose.

Camera

A Logitech HD Webcam C920 is mounted on the top of Jaymi. It is capable of 1080p video recording. Jaymi uses the data input from the camera for line and obstacle detection.

Computers

In order to process sensor data, two Intel NUC5i3RYH computers are mounted on the robot. Each unit has an Intel i3-5010U processor, and USB 3.0 ports for high-speed data transfer. One NUC performs vision processing, and the other runs our path planning and sends commands to the motor controllers.

Monitor

Jaymi is equipped with an Acer FT200HQL 19.5" monitor and it interfaces with robot through HDMI. The monitor's primary function is to help the software team work with NUC units when they are on the robot and cannot be accessed from external devices.

Open Source Motor Controller (OSMC)

The OSMCs are high-power H-bridge circuits controlling permanent magnet DC motors. The OSMCs, instead of heavy heat sinks used by many other types of motor controllers, use cooling fan to remove heat. Four OSMCs total are used and a pair of OSMCs jointly controls one motor on each side of the robot.

Motor Shield

A custom shield is mounted onto an Arduino Uno board. This shield receives inputs from the rotary encoders, and outputs a 12VDC signal to the OSMCs. The Arduino counts the encoder interrupts and outputs PWM, direction, and select signals, which are level shifted from 5 to 12 VDC.

Light Shield

A custom shield is used to control the onboard lights. The light shield Arduino monitors the e-stop solenoid, and when the motor system is engaged toggles the onboard safety light. This shield also provides underglow functionality to the robot.

Rotary Encoders

The robot uses US Digital EM1 optical quadrature encoders to track its wheel revolutions. This information is fed to the PID loop running on the motor shield Arduino.

Safety Devices

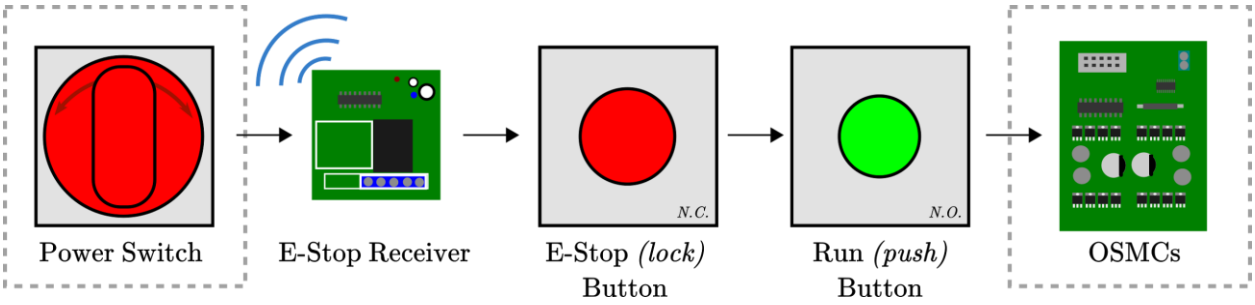


Figure 11. The e-stop mechanism.

The robot is designed to contain stages that both assure that motor motion does not occur when undesired and halt the motor system should the robot run out of control, as shown in **Figure 11**. Activating the Power Switch connects the batteries and the rest of the electronics. Initially, for the current to flow to the motors, the e-stop receiver must be wirelessly toggled and the e-stop lock button must be left in the un-pressed position. The receiver allows the robot to be halted remotely, and the button acts as a safety mechanism: should some error cause the robot to move backwards as a team member is writing code behind the robot, a press to this button will cut off all current to the motor system. Pressing the run button allows current to begin flowing through the e-stop solenoid mechanism, thus permitting the OSMCs and motors to be powered.

Software

At the beginning of Fall 2016, the software subteam created a list of improvements to work on over the next year. After ranking these by priority, it was determined that the most important issue was the lack of both real world and simulated testing scenarios. Due to various problems, no robot was available to test software methods in real world scenarios for a significant portion of the year. Thus, it was decided a simulated testing environment was needed.

After transferring the codebase from Qt to ROS (Robot Operating System) a year prior, there were many tools from the ROS toolkit to use such as ROS's Gazebo. Gazebo uses the ROS communication framework and allowed the code to be linked with this environment in order to accurately simulate the robot's sensors performance in a variety of course scenarios. In order to prepare for this implementation, the beginning of fall semester was spent updating to the newest ROS framework, Kinetic, and the newest Ubuntu distro, Xenial 16.04. The rest of the work was split into four groups: Gazebo, Path Planning, Vision Detection, and Mapping.

Gazebo

A large amount of effort this year was put into working on creating a useful simulated environment in Gazebo. Since Gazebo is highly integrated with the ROS communication framework and the codebase also uses ROS's latest communication framework, integrating our code with Gazebo was faster than expected. Thereby, the main effort was not spent on modifying our code but on creating a 3d world with sensors that perform as close to the real-world sensors' performance as possible.

The first task was to import a simplified version of the mechanical team's CAD model of the new robot. A URDF file was then created that adds metadata to this mesh. This metadata is important since it tells Gazebo how these meshes should interact in the simulation. The main aspects of this include collisions and torque such as how the wheels move in the simulation.

Another important aspect of the URDF is that it creates the virtual positions of the sensors such as the cameras in the Gazebo environment. Since the camera positions will be slightly off in practice, we have included a simulated error in Gazebo. This allowed for a more accurate simulation to see how our methods perform in real world scenarios.

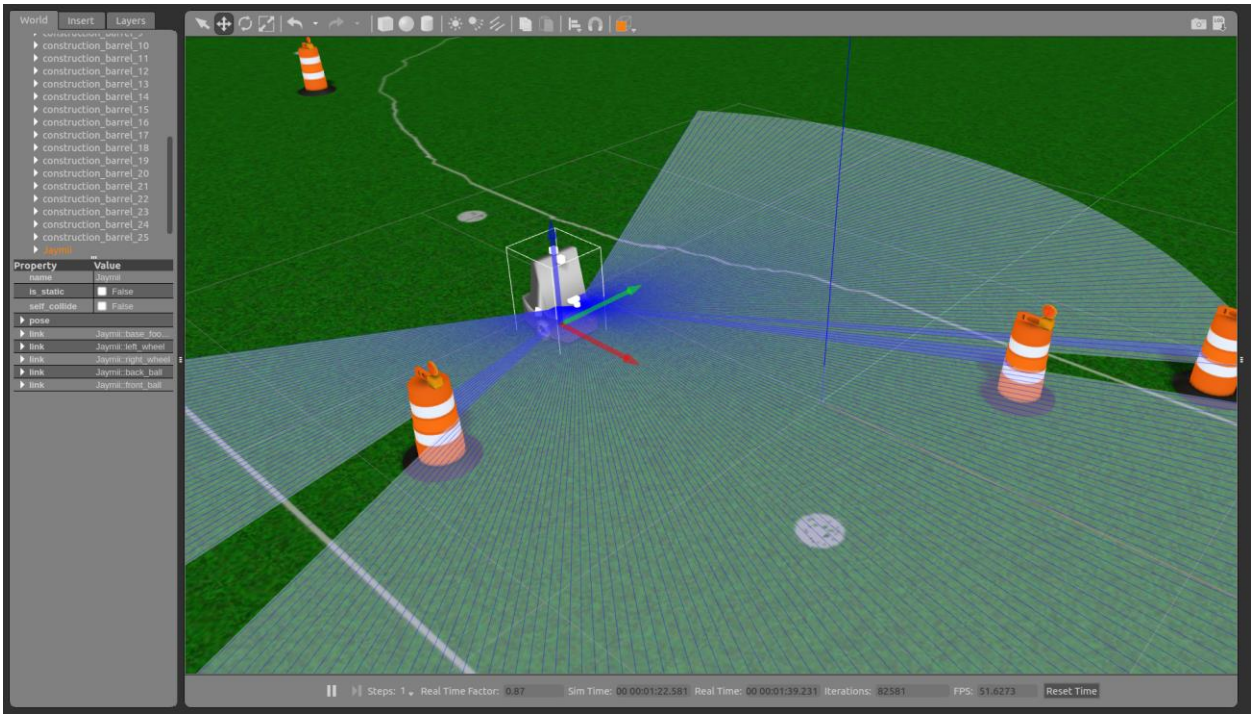


Figure 12. Gazebo Simulation Environment

Path Planning

The pathing for the robot uses a kinematic model of a differential drive robot to generate nodes for a graph search dynamically. A* is used to search through all valid nodes and to determine what nodes to expand and thereby generate their children. The validity of a node is checked by seeing if, along the path from the previous node to that node, an obstacle is within a certain distance. There is both a hard and soft threshold for the robot in pathing. Any node created

within a predefined distance of an obstacle is not considered or expanded by the graph search. Once a path is generated the relative costs are calculated using the soft threshold to weight the different paths and reward the paths that have the greatest minimum distance between their nodes and an obstacle.

Vision Detection

Our vision detection methods are similar to last year's vision detection techniques. However, similar methods in both line and pothole detection modules have been merged into a shared module. We also fixed an issue in how both modules were transforming their projections matrices and it has been improved by standardizing how each module calculates their transform using the transform tree. This also helps consolidate the data retrieved from the three cameras and LiDAR.

Line detection follows the algorithm from last year by using a canny edge detector and redrawing the lines with probabilistic Hough lines. In addition to modifying the values that determine the thresholding in these algorithms, these values have been moved to separate config files that can be changed dynamically.

Pothole detection uses a similar method by thresholding the image and using Houghcircles to find circles in the image. The size, shape, and location of these circles are then checked and only the circles which are the most probable of being a pothole are retained.

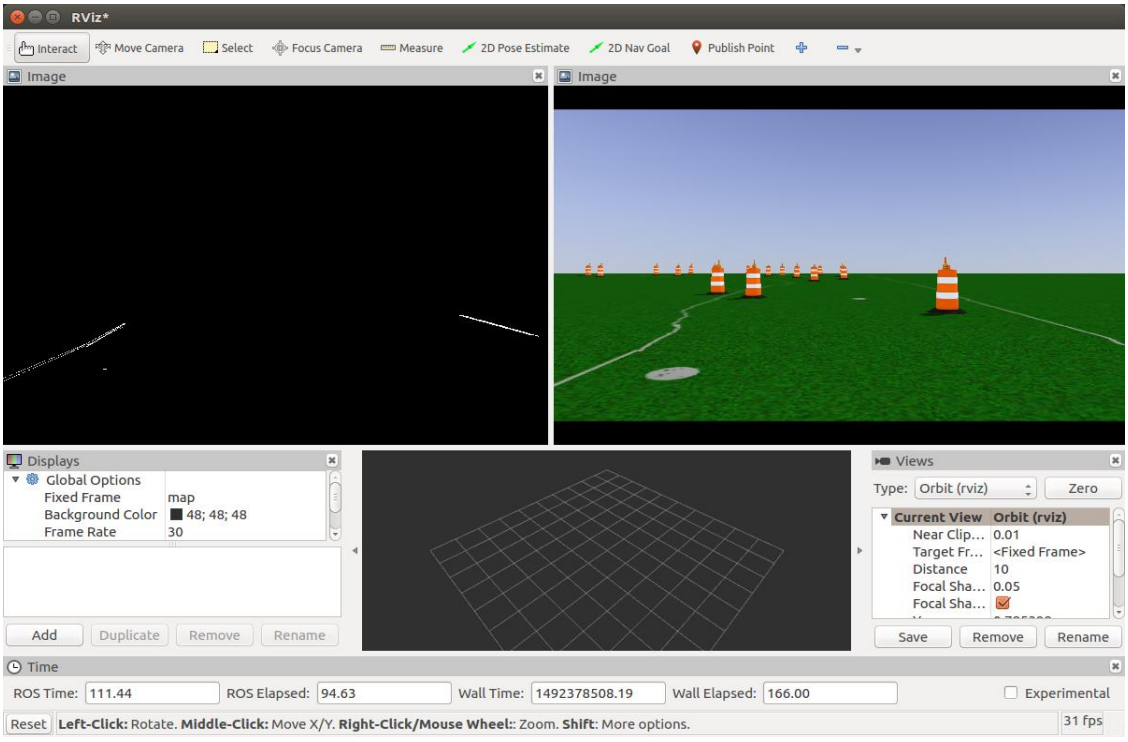


Figure 13. Line and Object Detection

Mapping

Another important change this year was the addition of global mapping and a more efficient way of merging the point clouds together. Before this year, detected obstacles were mapped into a point cloud in the local reference frame of the robot. Since this is local to the robot, it must be redrawn on every update to find a new path. This is sub-optimal and has decreased processing efficiency.

The solution is to use a global reference frame where obstacles can be drawn and assumed to stay in a fixed location if the position error is reliable. While this has improved the performance, and allowed the creation of a 3d map that remembers the history of the obstacle course, it comes with the drawback of causing any small error in the positioning to propagate dramatically.

This has been negated, however, by having newer points match up to the older points using a radius search on an octree representation of these points and shifting the position frame by the average of these offsets using ICP.

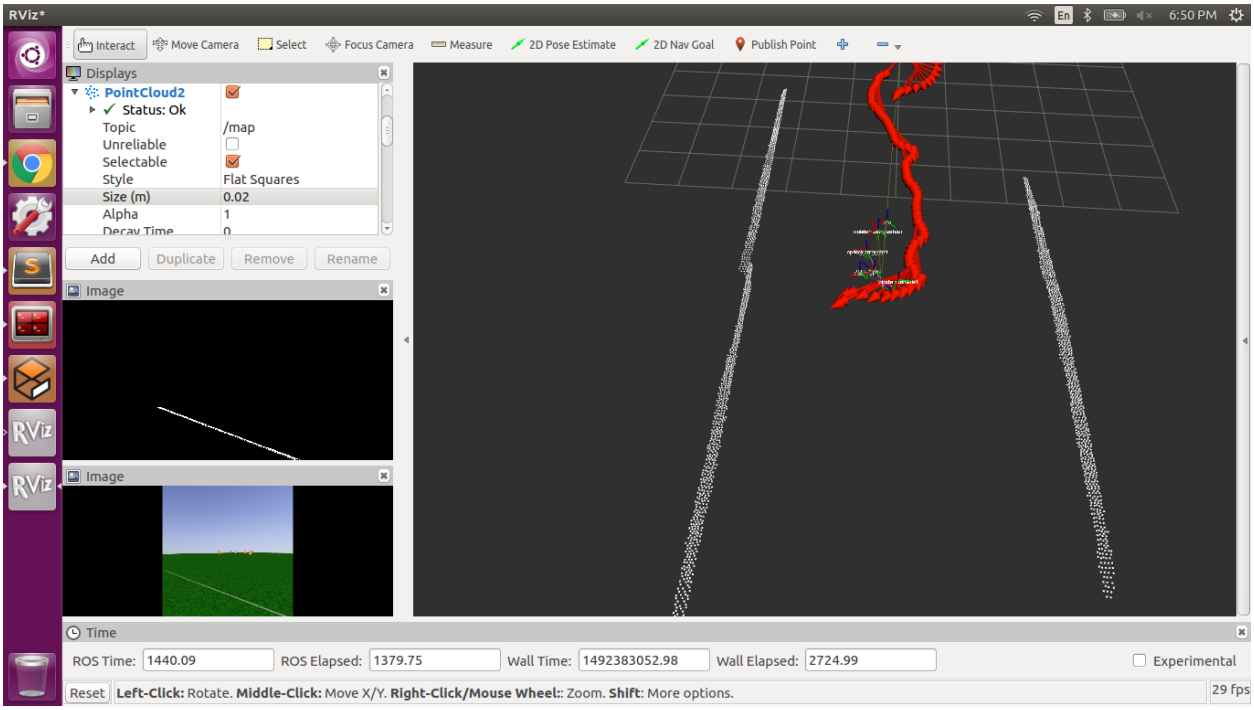


Figure 14. Mapping

References

RoboJackets. "RoboJackets/igvc-software." GitHub. RoboJackets, 30 Apr. 2017. Web. 15 May 2017.

Bill of Materials

Item	Value ($)	Cost to Team ($)	2017 Cost ($)
Steel Tubing	220	220	220
Aluminum Tubing	100	100	100
2x Dampers	1190	1190	0
16x Clevis Rode Ends	200	200	0
2x Gearbox	600	600	600
2x Caster Ball	70	70	70
Misc Mechanical	2500	2500	2150
SickTiM 551	1920	1920	0
Outback A321 GPS	7000	7000	0
2x Optical Encoder	170	170	0
2x DC Motor	900	900	0
4x Motor Controller	820	820	0
2x Battery	460	460	0
2x Intel NUC	730	730	0
DC Touchscreen Monitor	190	190	0
19V Power Converter	50	50	0
12V Power Converter	15	15	0
Razor IMU	75	75	75
2x Arduino Uno	60	60	30
Lioitech C920 Camera	70	70	70
2x Logitech C270 Camera	80	80	80
Headlight	250	250	0
Safety Beacon	350	350	0
E-Stop	80	80	0
Wireless E-Stop	50	50	0
Misc Electrical	350	350	60
Total	**18500**	**18500**	**3455**

Georgia Tech | The George W. Woodruff School of Mechanical Engineering

Dear IGVC team Organizing Committee:

The RoboJackets IGVC team has made significant efforts to rectify the performances issues of the previous autonomous ground vehicle design - Mistii. For the new autonomous ground vehicle, Jaymi, the IGVC team (1) **designed a new driving mechanism** to better balance the robot's mass and prevent instability, (2) **streamlined the electrical assembly** to improve space efficiency, reduce EMI with onboard sensors, and reduce debug and repair complexity, and (3) **implemented Gazebo tools in ROS** for simulating path planning.

I certify that the work done by the team in this extracurricular project is significantly greater than that required for a senior design course at the Woodruff School of Mechanical Engineering.

Sincerely,

Frank L. Hammond III, Ph.D.
Assistant Professor of ME and BME
Georgia Institute of Technology
George W. Woodruff School of Mechanical Engineering
Wallace H. Coulter Department of Biomedical Engineering