Boise State University - Design Report

15 May, 2019

Team contact information:

Parker Parrish	parkerparrish@u.boisestate.edu
Doug Williams	dougwilliams@u.boisestate.edu
Nathan Regner	nathanregner@u.boisestate.edu
Griff Allen	griffallen@u.boisestate.edu
Chris Ruby	chrisruby@u.boisestate.edu
Ian Varie	ianvarie@u.boisestate.edu
Jason Kuwada	jasonkuwada@u.boisestate.edu

I hereby certify that the design and development of the vehicle Bender, 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. Steve Swanson
College of Innovation and Design
Email : steveswanson@boisestate.edu

1 Introduction

The Boise State University IGVC Team will be competing for the second year in a row with an improved vehicle: Bender. Design improvements for the 2019 competition began in the 2018 fall semester, with the aim to improve navigation speeds and hardware reliability. New cabling, weatherproof enclosures, and custom circuit board designs contributed to a more robust electronics sub-system. To improve navigation reliability and speed, a simultaneous localization and mapping (SLAM) solution was employed along with a better quality inertial navigation system (INS). Bender still makes use of an all-wheel steering solution, with an improved control law that allows for independent translation and rotation.

1.1 Team Organization

This year's team consisted of seven members. The team leader was responsible for managing the teams budget in tandem with the faculty mentor and ensuring that design goals were completed on time. The mechanical team was responsible for wheel assembly, chassis design, and the housings for the electronics systems. The electrical team designed and produced the printed circuit boards and wire harness assemblies. The software team managed the codebase for the vehicles navigation and mapping systems.

Team Member	Major	ME	EE	CS	Hrs
Parker Parrish (lead)	Computer Science (BS)	X	X	X	180
Doug Williams	Applied Science (BS)	X			100
Nathan Regner	Computer Science (BS)			X	140
Griff Allen	Mechanical Engineering (MS)	X			35
Chris Ruby	Electrical Engineering (BS)		X		30
Ian Varie	Electrical Engineering (BS)			X	20
Jason Kuwada	Mechanical Engineering (MS)	X			30

1.2 Design Process

Our design process consisted of a five step process:

1. Analyze problem: Define the problem that needs to be solved
2. Prioritize: Determine which problems are most critical and which problems must be done first
3. Research solutions: Research different approaches, generate ideas, and select the most promising
4. Implement solution: Implement the solution in code
5. Test solution: Test the solution, first in simulations and then in the real world. Collect data and repeat

2 Effective Innovations

2.1 Electrical Innovations

- Bender’s modular power system allows for hot-swapping batteries that need to be charged. This extends the systems effective run-time indefinitely provided there are extra packs that can be charged. Additionally, the motor groups do not need to be realigned as long as there is power to the microcontroller. The ability to hot-swap power makes usage for extended periods of time simple to manage.
- A custom designed PCB is used to control the motor groups, safety light, and wireless estop. The benefits of a custom PCB, as opposed to a prototype board or breadboards, include reliable electrical connections, less occupied space, and inherently better documentation.

2.2 Software Innovations

- Separate translation/rotation logic: By using camera data to control rotation, it was only necessary to map lidar data for translation and obstacle avoidance.
- Modularized code: Code was written in small, modular ROS nodes. This decoupling enabled logging/replay of sensor data, allowing for simulated tests to be run.

2.3 Mechanical Innovations

- The frame and wheel assemblies are custom welded and machined out of aluminum 6061 sheeting and extrusion.
- The modular design of the wheel assemblies, frame, and electronics housings ensure that upgrades for future competitions can build on progress made in this seasons design.

3 Mechanical Design

3.1 Overview

The decisions made by the mechanical design team centered around meeting three main goals. First, the wheel assemblies needed to achieve point rotation for each wheel as opposed to sweeping turns in order to simplify the steering algorithms. Second, the comprehensive design was built with modular repair and upgrades in mind. Finally, the modular design had to allow for readily accessible electronics throughout the stages of prototyping.

3.2 Frame design

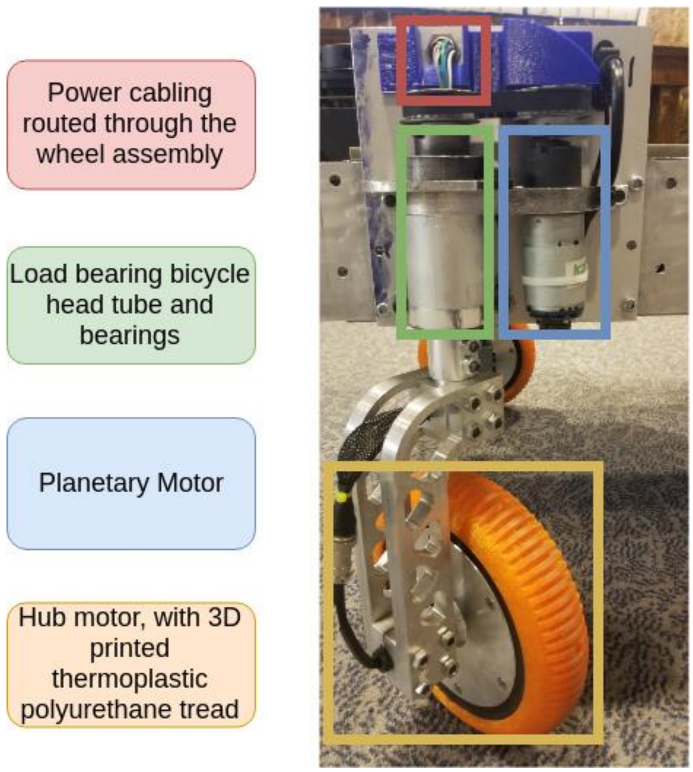
Bender's mechanical components are broken down into two basic sections: frame and wheel assemblies. The frame was created to be lightweight while still providing essential structural integrity, while the wheel assemblies are durable and modular. The frame is constructed from 6061 alloy, two inch by four inch extruded aluminum rectangular tube section with a 0.125 inch thick side wall. A .25 inch aluminum sheet is bolted to the bottom of the frame, and a removable polycarbonate sheet can be slid in place on the top of the frame. Weather stripping is outlined around the top edge of the square aluminum frame underneath the polycarbonate so that moisture does not leak inside.

Tungsten Inert Gas (TIG) welding holds the base frame components together. Custom brackets and braces, also TIG welded, hold the vertical frame portions of the frame to the base, while drilled and tapped holes were used to attach the base panel. The vertical structure was designed with 80/20 aluminum extrusion to maximize potential mounting space for the sensory systems. Sliders attached to the vertical assembly can be adjusted to allow camera angles to be optimized while leaving the emergency shutoff button and signal light on a separate level.

3.3 Wheel Assembly

The drive system is composed of planetary motors geared for steering rotation and internal hub motors for linear forward motion. The entire assembly was modularly designed to aid repair and redesign when needed. The comprehensive single assembly is mounted to a 0.25 inch thick plate which, in turn, is bolted to the side of the frame. A 3D printed wire cowl houses the interface of weatherproofed aviation junctions and a USB port per wheel assembly. These plates are positioned such that the tire contact points on the ground form a square rather than a rectangle in order to simplify the overall modeling algorithm. The rotational axis of each steering arm is designed to be concentric with the center of the tire contact point, which gives a single point of rotation for each steering assembly. This is to avoid sweeping rotations in navigation. Positioning the steering motors parallel and lower than the top surface of each respective steering arm allowed the entire system to have more visual clearance. This is so the top of the steering arm assembly does not interfere with the camera's field of view.

Each vertical steering post was designed to be a tube rather than a solid rod, allowing power and feedback cabling to pass through the center of the steering rotation axis. This, in turn, allows the wheels to rotate a full 360+ degrees. The vertical tubes are mounted in custom bearing blocks with bicycle head tube bearings taking the load of the robot. These bearings were chosen specifically because they are lightweight with a diameter large enough to provide clearance for the vertical tube while being strong enough to easily hold the overall weight of the robot plus cargo. The top of the steering tube and planetary motor are connected by Neoprene and fiberglass. Tension in the belt is achieved through the use of an eccentric friction mounted cam that wraps the planetary gearbox on the drive motor. By rotating the entire steering motor and gearbox assembly, the center-to-center distance of the pulleys can be increased or decreased. This allows for easy removal or application of belts and greatly simplified tensioning procedures. The wheel assemblies are encased in a weatherproofed aluminum box. The box has an access lid for repair and adjustment of the wheel assembly components.



3.4 Wheel Alignment

For the purpose of physical feedback and initializing the drive system on Bender, visual inspection was chosen prior to each power-on event. During the design process, it was proposed to use optical sensors to eliminate any issues of repeatability regarding the visual inspection process. The proposed sensors included line break or reflective sensors. Optic sensors, like the line break and reflective sensors, are prone to interference and would offer a challenge in mounting. There was also significant testing involving microswitches, as they were the only feasible option. However, they were more prone to inaccuracy and had a significant lack of repeatability in comparison to a visual inspection.

Steering is achieved by a planetary DC motor, which in turn rotates the drive assembly via belts. This process can be seen on each of the four drive assemblies. Planetary motors are equipped with a rotary encoder, allowing Bender to track the movement of each of the wheels.

4 Electrical and Power Design

4.1 Overview

The heart of Bender's electronics suite is a custom designed logic board which accommodates the safety light circuit, the microcontroller driving the motors, and the microcontroller driving the wireless e-stop. Generic motor controllers for both the brushed and brushless motor controllers occupy a separate electronics housing. Popular USB and DB15 connectors interface from the board to the motor controllers, while weatherproof aviation connectors interface from the motor controllers to the motors. 20V DeWalt batteries are employed as a convenient and reliable power source.

4.2 Power Distribution

Bender is powered by up to three 20V DeWalt power tool batteries connected in parallel. A minimum of two is required to provide the power necessary to run the system. When running under normal operating conditions with two batteries, Bender can operate for 45 minutes continuously. Power from these batteries is delivered solely to the motor controllers. To conserve space, an aluminum mounting plate was chosen to aid in heat dissipation for the motor controllers; each is synced with thermal tape and secured in place by mounting screws. From the motor controllers power is delivered to the wheel Since the wheel assemblies can rotate 360 degrees and the wires transporting power to the planetary and hub motors pass through the assembly, a stranded high flex silicon insulation cable is used. Initial testing and continued use of the system indicates that this solution can tolerate five full revolutions before damage to the wire is possible. To manage this risk, the microcontroller driving the wheels monitors for wheel wraparound, and will not allow the total passes to exceed this limit.

4.3 Electronics Suite

Lidar

The RPLiDAR A1 laser range finder scans for obstacles in a 270-degree sweep ignoring chassis of the robot from its field of view. It has an update rate of 5Hz at sixty readings per sweep, and a max range of 15m. The lidar is powered via USB from the laptop.

Digital Camera

The Logitech c930e is a standard consumer grade web camera with a 70-degree field of view. This model was chosen for its low image distortion.

Pixhawk 4

The pixhawk4 and associated GPS serve as Benders compass for waypoint navigation. It is equipped with a triple redundancy IMU and is also powered by the laptop

Computer

The central processing hub for the electrical system is a laptop computer equipped with a Core i7 2.50Ghz processor and 64 GB of DDR4 RAM. This provides power for all the sensory systems and runs a Linux to host the ROS meta-operating system that the navigation software relies on.

4.4 Safety Devices

Bender is equipped with multiple safety devices that ensure both the safety of the operators and the system. A safety light, driven by the Teensy, will blink to indicate that a speed is being written to the hub motors, and remain on for as long as the system has power. In case the system malfunctions two safety stops were incorporated into the design, one wireless and one built into the systems power circuit. The wireless estop has a separate dedicated circuit and microcontroller built into the primary PCB. This circuit drives a transceiver module that will open a power relay when a kill signal is received. The second circuit is in a hand-held 3D printed case with an identical transceiver and microcontroller. This circuit will be held by an operator at all times ready to wirelessly shut off power via a red pushbutton mounted to the shell of the case. The other e-stop is mounted to Bender's chassis and when pushed will physically disconnect the power supply. To protect from short circuits in the power supply circuit, a primary 100A fuse was installed. The individual motors are also protected against current overdraw malfunctions with 15A fuses installed on each one.

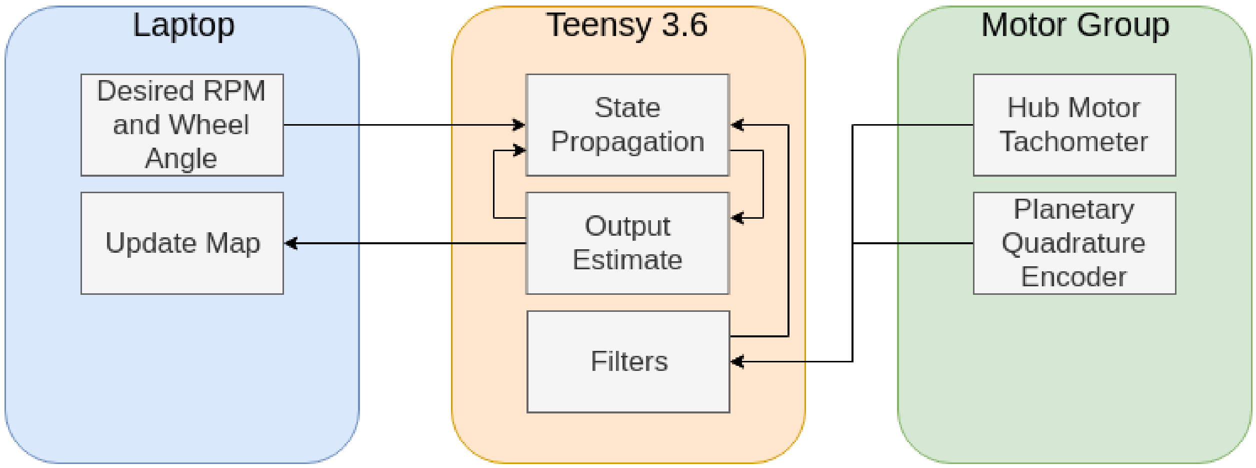
4.5 Drive Control

Drive on Bender is managed by four brushless hub motors, while steering is controlled by four brushed planetary motors which are all driven by a Teensy 3.6 microcontroller. This particular microcontroller was chosen because of its convenient size and a high number of interrupt capable pins. Each motor group requires two interrupt pins to keep track of the motor position. The hub motors provide odometry feedback via hall effect sensors which are converted into tachometer data by the BLDC's motor controllers. The planetary motors also provide a high-resolution quadrature encoder feedback which is processed by the Teensy directly. The desired speed for each wheel and desired angle position are sent to the Teensy from the laptop over serial USB TTL connection. Each motor group has two PID control algorithms running on the Teensy; one for the hub motor to control speed and one for the planetary motor to control angle position.

Integral to the navigation computer's ability to estimate position and state in the world is a reliable set of odometry data feedback from the Teensy. Inherent in an all-wheel-drive system on uneven terrain is the possibility for slippage and zero contact from the wheel surface to the ground. This presents a challenge when trying to preserve the integrity of odometry and combined these events compound the error in reported odometry feedback compared with the robots actual position and orientation. Additionally, the time taken for the planetary motors to reach desired angle positions will not be consistent due to the changes in weight distribution when the terrain is uneven.

To overcome inaccuracies present in raw odometry readings are a set of filters and state propagation algorithms. The filters used include a maximum and minimum delta filter and a sliding average filter. These filters account for the chassis's overall desired translation and rate of rotation about the center and will change the thresholds depending on what maneuver the robot is making. For example, the filters for each wheel if the robot is translating without rotation will all be identical, whereas if the translation is accompanied by rotation, only two motor groups will be performing the same actions so the sliding average will change its scope to focus on the motor groups with identical behavior.

The filtering improves accuracy in reported odometry at the cost of severely delaying the time when reliable odometry readings are available. The sliding average filter requires up to half a second of data to provide the most reliable readings, which is an infeasible delay. This lag is combated by a state propagation estimator that predicts what the wheel speed and angle is based on previous odometry readings and the desired state being written. To avoid accruing error, predictions are constantly being informed based on the actual data reported from filtered odometry.



5 Software Strategy and Design

5.1 Overview

To manage the navigation software we are using the Robot Operating System (ROS). Code is divided into distinct "nodes" that communicate with each other using the ROS publish-subscribe graph model. Nodes are typically assigned a single responsibility, such as communicating with a sensor or aggregating data from other nodes.

5.2 Lane Detection

A single Logitech c930e webcam is used for lane and pothole detection. This particular model was chosen for its relatively wide 70-degree field of view, negligible radial and tangential distortion, and optionality to adjust image parameters like saturation, exposure, and brightness. The camera is mounted approximately 1.2 meters off the ground at a 30-degree downward angle. OpenCV is the primary tool used to detect lanes, but before any image processing takes place the camera's output is filtered for lower brightness, higher saturation, and fixed exposure. These settings were chosen to preserve color integrity across varying illumination conditions, allowing lane detection to work in both sunny and overcast conditions. After the initial processing is performed by the camera a series of OpenCV filters are used to discern noise such as glare, yellow, and laid down grass from the actual lanes. These steps are as follows:

1. Resize
    a. To reduce processing time, the top half of the image is cropped. At the mounted angle, the camera can see approximately ten feet ahead
2. Perspective Transform
    a. A perspective transform to a top-down-view normalizes the camera data to the lidar and odometry reference frames
3. Median Blur
    a. First, a median blur is applied to ensure that perceived breaks in lines due to inconsistent coloring will not result in a break in the detected lane.
4. HSV Threshold
    a. A Hue Saturation Value threshold converts the RGB input image into a binary image
5. Gaussian Blur
    a. While contour filtering can remove noise the gaussian blur is a faster way to achieve this if there happen to be many small patches of perceived lines
6. Contour Filter
    a. OpenCV contours are arrays of points that represent the outer boundary of a binary image. Once this boundary is known additional filters, such as area, moment (roundness), and solidity can be applied. This is how potholes are distinguished from lines.

Once the contours of the white lines have been detected, a first-order equation is solved to represent the approximate slope of the collection of points in the contour. This ensures that even if part of the line passes out of the camera's field of view, it can still be avoided.

Effectively obfuscating the white lines on the construction barrels is the final crucial step in taken in detecting lanes. Since these lines are elevated, the actual position of the obstacle is distorted resulting in bad data. Since the lidar can already detect the construction barrels that the white lines are painted on, there is no utility in trying to preserve the false data interpreted by the camera. In order to remove the bad data from the image, the data is compared with readings from the lidar, and points that overlap in the camera's image are removed.

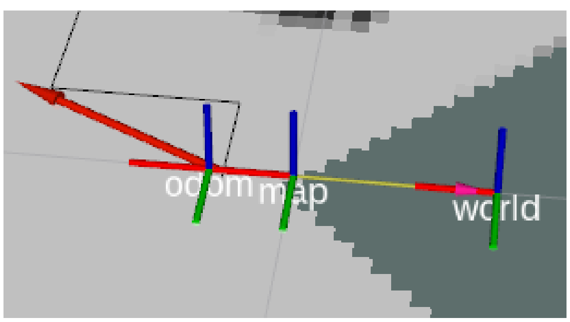
5.3 Waypoint Navigation

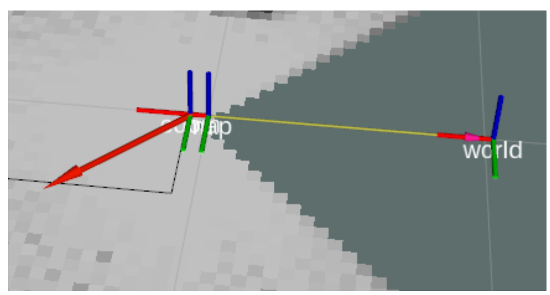
At a high level, navigation is performed using a simple deterministic finite automata with two types of states: line following and waypoint navigation. In the line following state, rotation is corrected to match the angle of the nearest line. In waypoint navigation, rotation is controlled by continuously computing the initial bearing required to reach the waypoint. Translation is directed by the lidar, regardless of the navigation state.

5.4 Map Building/Path generation/Goal Selection

Raw lidar data is used to iteratively build and update a local map of obstacles within a 2.5 meter radius. Mapping is implemented with a ROS package named BreezySLAM. This package handles position prediction, gap filling, and obstacle certainty. Mapping logic is encapsulated in a single ROS node, which outputs a 100x100 grayscale image.

Due to the performance implications of attempting to generate and publish a global map in real-time, it became necessary to use a small map and reset it when nearing the edge. This is accomplished by merging lidar and odometry data together when updating the map. When the map is updated, a check is done to see if Bender has traveled more than 0.5 meters since the last reset. If this is the case, the map image is shifted and rotated such that Bender is at the center of the map, with a heading of zero degrees. This new image is used as the is stored along with the current position for future updates.

<table>
<tr><th>Before</th><th>After</th></tr>
<tr><td>


</td><td>


</td></tr>
</table>

Before path planning is performed, a costmap is generated by applying OpenCV’s dilation operation to the map. This ensures any valid path will have a buffer distance of about 0.5m to the nearest obstacle.

Path generation is accomplished using a variation of Dijkstra’s algorithm. The algorithms starts at the Bender’s current x,y position in the map and iteratively expands outwards. If a pixel is filled, it is flagged as being untraversable and skipped. Otherwise, all unvisited neighbors of the current pixel are added to a list of positions to explore. Along the way, if an edge point is encountered, it is compared with the current best known path. If the length of the new path is less than the old path, it is recorded. The algorithm continues until it has exhausted all possibilities.

In order to break ties, the following weighting scheme is applied to edge points:

*W(Point) = Manhattan-Distance(Point, Ideal) * C*
where Ideal is the directly in front of Bender and C is a constant > 1

This ensures that a path straight forward is always preferred over one that travels sideways.

6 Cost Estimate

Item	Unit Cost ($)	Qty	Team Cost ($)
Laptop	$2000	1	$0
Motor Controllers	$13	8	$182
Teensy 3.6	$40	1	$40
Logitech Cameras	$90	1	$180
DeWalt Batteries	$80	4	$320
Aluminum Frame	$600	1	$250
Assorted Hardware/Electronics	$500	1	$500
Pixhawk4 + GPS	$300	1	$300
Hub Motors	$35	5	$175
Planetary Motors	$80	5	$400
LiDAR	$100	1	$100
Arduino	$15	2	$30
PCB + components	$110	2	$220
Total			$2997

7 Failure Points and Resolutions

7.1 Latency

One of the most significant failure points of the navigation algorithm is high latency. On average, it takes at least 0.4 seconds to propagate state from the sensors to the wheels.

Translation Latency: ~0.4s	**Rotation Latency:** ~0.35s
● 0.10s average lidar latency @ 5hz ● 0.15s map updates ● 0.15s costmap and path generation	● 0.10s+ webcam latency ● 0.25s seconds OpenCV processing

The solution to this problem is traveling at or near the minimum required speed of 1mph. This allows for plenty of time to react to new information.

7.2 Battery

Another point of failure is the laptop battery. At max charge, it is only able to last about 40 minutes. Aggressive CPU throttling also kicks in when unplugged, which creates cascading lag throughout the ROS network. The solution to this problem is to tweak available power settings for maximum performance and avoid major system updates that may break or alter these settings.

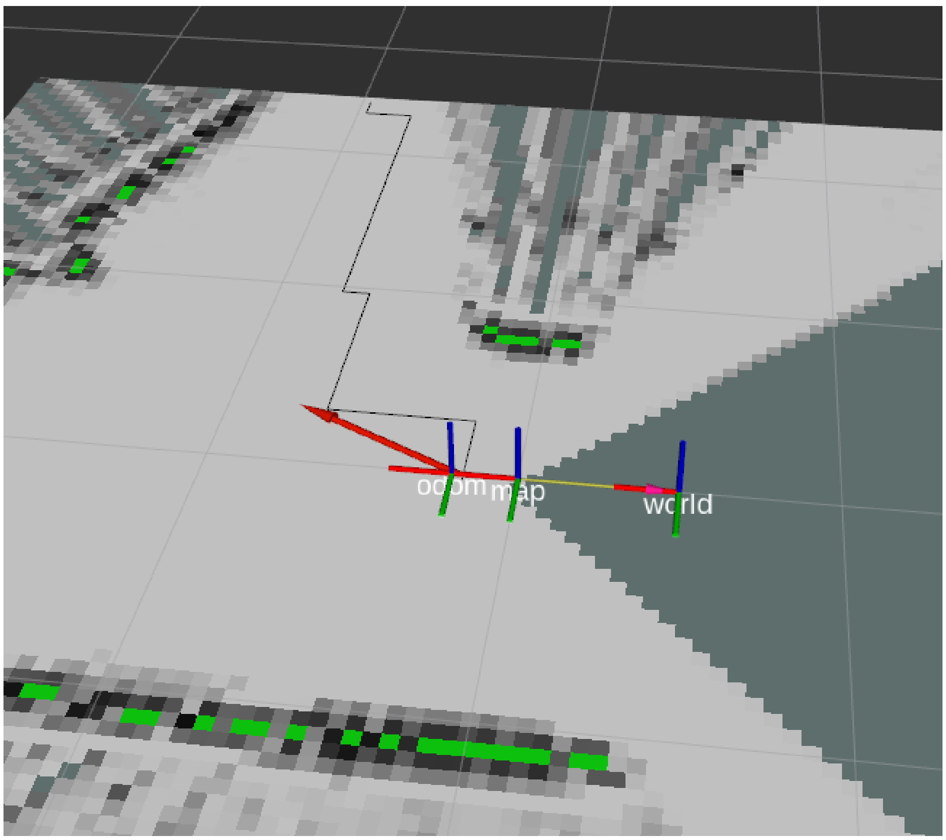
7.3 Wheel Control

While the Teensy 3.6 microcontroller can prevent the wheel subassemblies from damaging the wiring by twisting the cabling excessively if power from the laptop is lost and the wheels cannot reset this can result in accrued revolutions that the system is unaware of. Too many revolutions will damage the wiring, to prevent this failure from happening, visual inspection is necessary prior to each power-on event.

8 Simulations

Simulations are performed with pre-recorded sensor data from the lidar, camera, and odometry. This data is recorded and replayed using the rosbag package. This allows for fast iterative development of mapping, lane prediction, and path planning in a controlled environment.

Below is a screenshot of the simulation in RViz, a 3D visualization tool for ROS. The background image is a visualization of the map built from the lidar scans. Each of the red, green, and blue axes represents a translation and rotation relative to the starting position (world). Finally, the black line shows the output of the path planning node. This path is used to compute the translation vector, represented by the large red arrow.



9 Performance Testing and Assessment

9.1 Initial Performance Assessments

Bender was tested on an outdoor course that mimics the IGVC competition layout. Operational performances under these test conditions are listed below:

- Translational velocity of 1.5mph during runtime.
- Obstacle avoidance and line following work well with the exception of “U” or “V” shaped obstacle formations. Situations such as these are not solvable by the current path planning algorithm.
- Bender is capable of climbing ramps of a 15 degree incline. However, this skews the camera’s line detection algorithm and throws off lane prediction. To solve this problem before competition, we will integrate with IMU data from the Pixhawk to ensure the vehicle follows the upward incline of the ramp, regardless of any invalid camera readings.