Georgia Institute of Technology
RoboJackets

# Jessi

**Project Manager**
Dallas Downing | dallasd@gatech.edu
**Software Subteam Lead**
Jason Gibson | jgibson37@gatech.edu
**Mechanical Subteam Lead**
Tomas Osses | tomas_osses@gatech.edu
**Electrical Subteam Lead**
Ryan Waldheim | raw@gatech.edu

## Members

**Mechanical**

| | |
|---|---|
| Yongjae Won | ywon30@gatech.edu |
| Daniel Kilgore | dkilgore8@gatech.edu |
| Shane Kearney | skearney7@gatech.edu |

**Electrical**

| | |
|---|---|
| Alex Xu | rxu74@gatech.edu |
| Baran Usluel | baran@gatech.edu |
| Noah Daugherty | noahpd@gatech.edu |
| Soohyun Kim | sokim@gatech.edu |
| Mark Faingold | faingold@gatech.edu |

**Software**

| | |
|---|---|
| Justin Zheng | jzheng84@gatech.edu |
| Raymond Ortiz | rortiz9@gatech.edu |
| Dominic Pattison | dpattison3@gatech.edu |
| Matthew Keezer | rkeezer3@gatech.edu |
| Andrew Tuttle | atuttle7@gatech.edu |
| Evan Strat | estrat@gatech.edu |
| Aaron McDaniel | amcdaniel39@gatech.edu |
| Billy Vaughn | wvaughn9@gatech.edu |
| Caitlin Choate | cchoate6@gatech.edu |

**Faculty Advisor**
Frank L. Hammond III | frank.hammond@me.gatech.edu

May 15th, 2017

# Contents

# 1 Who We Are

## 1.1 Introduction

RoboJackets is the student organization for competitive robotics at the Georgia Institute of Technology. We are the Intelligent Ground Vehicle Competition team, one of the five teams in RoboJackets, mostly consisting of undergraduate students from many disciplines. This year, we designed and built a completely new robot, Jessi, for the 26th IGVC. After failing to produce a robot fit for competition last year, we wanted to approach the problem with a completely different mindset. Jessi's mechanical design features more ready-made parts for ease of manufacturing and focuses on usability for our software members. Jessi has an overhauled electrical system with new and custom components to replace old hardware. The software features a new mapping, path planning, and path following strategy, built off of last years improvements, as well as a completely new vision system. Jessi is a 250 lb. robot, consisting of a drive train, electrical compartment, and a sensor array. She drives with two electric wheelchair motors (Section 3.2.2) and navigates with a camera, LiDAR, GPS, and IMU (Section 4.3). She has two boat batteries for power (Section 4.2) and uses an Intel NUC and NVIDIA Jetson TX2 (Section 5.1). Overall, Jessi is a complete robot, ready to take on every aspect of this year's competition.

## 1.2 Design Process

Our design process started by analyzing the requirements for the competition and the areas of our previous designs that needed more attention. We decided on four main areas to improve for our mechanical design: manufacturing time, ergonomics, modularity, and waterproofing. Based on these priorities, we created an initial design in CAD. The design went through two iterations of peer review by other RoboJackets mechanical members and revision. During this process, some components, such as the base, underwent more focused review. We finalized the design and later handled any issues unforeseen in these two design iterations. Throughout the year, we updated the CAD with small design improvements to address unforeseen issues. By keeping a consistent and updated CAD model, we were able to avoid remaking parts as design changes were needed, an issue in previous years. The software design process was centered around improving the individual subsystems within the overall architecture rather than a complete redesign. This year we put a strong focus in upgrading our localization, mapping, pathing, and vision systems. Each system was assigned to a group of software members to complete for this year. Beyond these overarching projects there was significant time placed into new member training on the test platform at the start of the year. The electrical team focused their design on a more robust system through modular substitutions of key components while maintaining the general overall design.

## 1.3 Organization

The RoboJackets IGVC team consists of three subteams, lead by Project Managers Dallas Downing and Justin Zheng. The subteams focus on the three major disciplines of IGVC: mechanical, electrical, and software. There is a subteam lead for each subteam, responsible for coordinating and planning the efforts of the members. Over the course of Fall 2017 and Spring 2018 semesters, the team put in over 2700 hours working on Jessi. The chart below shows all our members, their area of work for the team, their class standing, and their major.

| Position | Name | Standing | Major |
|---|---|---|---|
| **Spring Project Manager** | Dallas Downing | Sophomore | Computer Science |
| **Fall Project Manager** | Justin Zheng | Junior | Computer Science |
| **Mechanical Lead** | Tomas Osses | Sophomore | Mechanical Engineering |
| **Mechanical** | Yongjae Won | Junior | Mechanical Engineering |
| | Daniel Kilgore | Sophomore | Aerospace Engineering |
| | Shane Kearney | Senior | Mechanical Engineering |
| **Electrical Lead** | Ryan Waldheim | Junior | Chemical Engineering |
| **Electrical** | Alex Xu | Sophomore | Computer Engineering |
| | Baran Usluel | Freshman | Electrical Engineering |
| | Noah Daugherty | Junior | Electrical Engineering |
| | Mark Faingold | Freshman | Computer Engineering |
| | Soohyun Kim | Freshman | Electrical Engineering |
| **Software Lead** | Jason Gibson | Senior | Computer Science |
| **Software** | Raymond Ortiz | Senior | Computer Science |
| | Dominic Pattison | Senior | Computer Science |
| | Matthew Keezer | Graduate | Computer Science |
| | Andrew Tuttle | Sophomore | Computer Science |
| | Evan Strat | Freshman | Computer Science |
| | Aaron McDaniel | Sophomore | Computer Science |
| | Billy Vaughn | Sophomore | Computer Science |
| | Caitlin Choate | Sophomore | Computer Science |

# 2 Innovations

## 2.1 Vision Processing

Jessi uses a convolutional neural network to detect lines and potholes. The network is robust, works under various lighting conditions, and touts an almost nonexistent false positive rate. Previous iterations often confused the white strips of construction barrels as the white of the course lines. The architecture and implementation details are explained section 5.4.

## 2.2 Localization

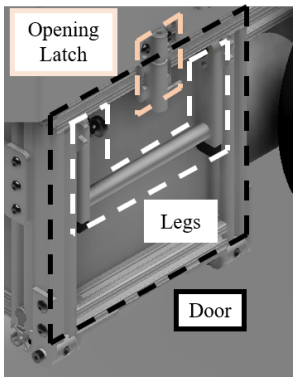
The localization of Jessi was completely revamped by utilizing the GTSAM (Georgia Tech Smoothing and Mapping) library, backed by factor graphs, to perform a sliding window trajectory optimization. This is a research level approach and is currently being studied and developed by multiple university research teams as a novel approach to localization. This approach is further explained in section 5.2.

## 2.3 Ergonomics

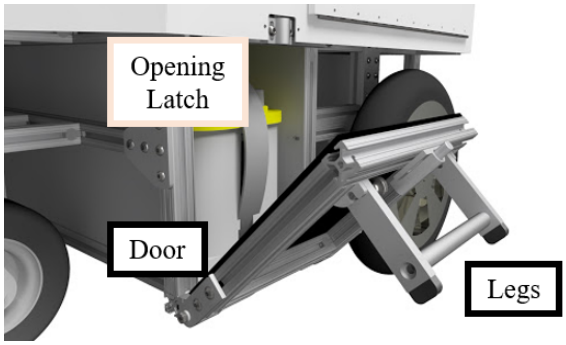
Jessi's mechanical design includes ergonomic innovations to make the electronics easy to operate, parts of the base easier to move, and the monitor easily viewable. The cover contains five separate access panels on the top, sides, and back of Jessi, such that the entire electronics tray can be accessed just by opening the panels. Typically, adding holes to a cover would increase the chance of water seeping through;

however, the design includes a latch that avoids this trade-off by applying pressure to a seal around each opening. A similar system can be found in commercial grade automobiles for waterproofing trunks.

We designed the battery doors ergonomically. We chose to line the doors with high molecular weight polyethylene, reducing the friction. This makes it easy to slide the bulky 35 lb batteries into the compartment. Additionally, the battery door includes two sets of spring loaded latches. One latch allows the battery door to fold out while the other allows a set of support legs to come out perpendicular to the door. This system allows the batteries to be placed on top of the door parallel to the ground and then slid into the compartment with ease. Figure 1 shows the battery door in both the closed and open positions, with labels for the opening latch, legs, and door.




(a)




(b)

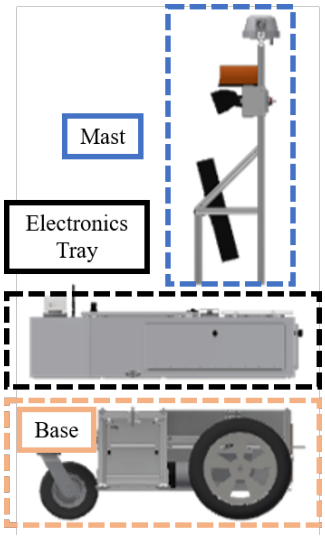
Figure 1: Battery door folded (a) and extended (b)

The monitor contains the final ergonomic innovation. We chose to place the top of the screen at the average eye level for software team members when they are sitting down. Visibility also improved by surrounding the sides of the monitor with black polycarbonate sheets to reduce glare from external sources of light. Additionally, we mounted the monitor to the mast using a hinged VESA mount, yielding two rotational degrees of freedom. This makes the monitor easy to adjust during operation as well as easy to view from the front of the robot for demonstration purposes.

# 3 Mechanical

## 3.1 Overview

Our main design objective for Jessi is manufacturability. Jessis aluminum 6105 T5 frame contains three modules: the base, the electronics tray, and the mast. For a faster manufacture time, we used 80/20 Inc.'s T-slotted framing as the cross section. The base supports the weight of the robot and houses the payload, batteries, and drivetrain. The base contains these heavy objects to lower the center of mass and improve stability. The electronics tray houses the majority of the electrical hardware safely under a water-resistant cover, which includes access panels for easy access. The mast refers to the top portion of the aluminum frame that supports the monitor, GPS antenna, Inertial Measurement Unit (IMU), camera, safety light, flood light, and button box. Figure 2 shows the three modules from a side view.



## 3.2 Structure Design

### 3.2.1 Frame

6105 T5 aluminum was used as the frame material for its relatively low cost and for its high strength to weight ratio. The 80/20 Inc.'s cross section decreases manufacture time and allows failed parts to be replaced easily. However, the ease of removal comes with the cost of fasteners loosening over time. Medium strength heat release Loctite Threadlocker 243 fixes this issue by increasing the friction between the male and female threads. This reduces vibration-induced thread lossening and allows a member to be removed just with a torch and an allen key. The main fasteners used are the anchor fastener, T-nuts, and end fasteners. The fasteners met the specifications of the appropriate loading conditions at each connection. We chose anchor fasteners where the end fasteners would not suffice and chose T-nuts in conjunction with plates for higher bending moment loads. For example, a long 12 T-nut plate and 5 T-nut plate fastened the members above the motors and behind the casters, respectively, to avoid bending-induced failure. Figure 3 shows the two aforementioned T-nut and plate assemblies. We chose end fasteners where the load allowed because the end fasteners are very convenient for removing individual elements. End fasteners also helped in members that experienced large torsion.


(a) (b)

Figure 3: T-nut and plate assemblies behind caster (a) and above motor (b)

### 3.2.2 Base

The base endures the greatest load experienced by the robot, so its structural integrity was prioritized. Aside from fastener specifications, we designed the base to have a safety factor of at least 2 under static conditions while at the same time resisting fatigue failure. We performed a finite element analysis (FEA) on the base to verify this safety factor (see section 7.2). The bases design accommodates the drivetrain, batteries, and payload. The position of these items places the center of mass low to the ground and slightly in front of the wheels. The two separately powered gearmotors and caster wheel make up the differential drivetrain. Two pre-built ElectroCraft MP-36 MobilePower Geared Motors power the drive wheels. They have a max torque of 108.5 Nm - far greater than the 61.2 Nm needed on each drive wheel to scale 15° inclines - and a max output rotational velocity of around 115 rpm. This coupled with the 14 inch wheels lets Jessi reach a maximum of 4.79 miles per hour - almost the maximum allowed speed for the competition. This drive system with the single caster wheel allows a fairly mobile robot with the capability of rotating 360° in place. The wheels are two feet apart from each other and three feet away from the caster wheel in order to be as small as possible without going below the minimum size constraints for the competition. This small size improves Jessis overall maneuverability.

### 3.2.3 Electronics Tray and Mast

We mounted the rest of the sensors on the mast and on top of the electronics tray. The camera, GPS antenna, IMU, and safety lights are all at the top of the mast - around 57 inches from the ground - slightly below the maximum height constraint for the robot. These sensors need to be as high as possible because the camera needs as big a field of view as possible and the GPS antenna and IMU need to be isolated from interference put out by the electrical system. We mounted the LiDAR to the very front of the robot on top of the cover to maximize its field of view at a height that would not hit the reflective lines on the barrels.

## 3.3 Suspension

We did not develop a suspension system for Jessi. It would have been beneficial for line detection, as it would stabilize the image; however, testing on a prototype platform without suspension proved that this was not a major issue. As such, we opted for no suspension to minimize cost and simplify manufacturing.

## 3.4 Weather Proofing

The electronics tray and mast focused on protecting the electrical systems as well as the sensors. We welded, body filled, and painted the cover such that no water would get in to the body of the robot. We also added epoxy to the inner edges of the cover to further prevent any seepage. The cover had two paints - white and clear coats - that prevented glare from external sources of light. The white paint also serves to reduce albedo and keep the inside cool. We lined the previously mentioned access panels with rubber seals in conjunction with the tight latching mechanism to prevent water from entering the holes machined in the cover.

The camera, IMU, LiDAR, encoders, E-Stop button box, GPS, and monitor all come with their own waterproofing mechanisms. The encoders go inside the end cap of the motors. The E-Stop and safety light switches go inside a water-resistant box. The encoder housing on the motor is not IP rated; however, the holes for screw access are protected with Buna-N O-rings. We added a gasket between the cap and the motor to further prevent water seepage. The button box is NEMA 4X rated. To further weatherproof the button box, we mounted the box inside a rectangular tube covered on both sides with bent aluminum sheets. The GPS antenna is rated IP69K by the manufacturer. Clear polycarbonate encases the LiDAR and camera to prevent water from getting directly into the sensors. Similarly, an opaque polycarbonate sheet with epoxy sealed seams covers the monitor. Finally, a 3D printed casing encloses the IMU mounted on top of standoffs to prevent water seepage.

Wires that connect the sensors to the electronics tray came with their own waterproofing issues.We mounted an aluminum lip on top of the wire access holes to prevent accumulated water from going inside. A bent sheet of aluminum covers the lips with some clearance and contains holes on the sides for the wires to stop water ingress. Figure 4 shows the wire access hole assembly with the sheet, lip, and access hole.

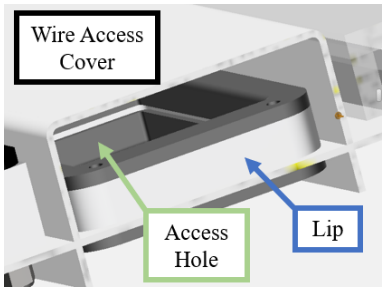



Figure 4: Section view of the wire access hole lip and enclosure

# 4 Electrical

## 4.1 Overview

Jessi incorporates a high-powered rail system that distributes power between two boat batteries put together in series to generate a main 24V rail. The batteries power the two motors, a variety of sensors, and all custom electronics on board. All motor control is handled by a custom printed circuit board which is powered by an mbed LPC1768 microcontroller. Emergency stop mechanisms are also handled by a custom printed circuit board, which is in series with the on-robot button.

## 4.2 Power Distribution

The main power of the robot is supplied via two 12V lead-acid batteries in series, generating a main 24V system. The battery power is switched via a large disconnect inside the robot which is rated to handle 180A (more than twice than any max reading observed on the robot). The motors, flood light, and E-Stop circuitry all feed off of this main rail. Two voltage converters then step down the voltage to both 12V and 19V. The main computers of the system, the Intel NUC and Nvidia Jetson TX2 (Section 5.1), and the monitor are both powered from 19V. The 12V powers the rest of the sensors, except for those powered off of other modules, including the IMU, the LiDAR, and the encoders.

## 4.3 Electronics Suite

### 4.3.1 Motor Board

The printed circuit board located near the center of the electronics tray was designed and assembled entirely by our team. We used Autodesk Eagle to design the board and then had it printed by Seeed Studio. All components were ordered through Digikey and assembled on site. The central compute module of the board is an mbed LPC1768 with most of the other components handling lower-level logic and voltage regulation. The board was custom designed to handle control of the motors, safety light patterns, and some future-proofing features.

The motor board primarily relays commands from the on board computer to the motor controller. Communication to both the NUC and motor controller are handled via a serial interface, with the NUC utilizing the mbed's USBSerial library. Data is fed from the motors to the motor board via encoders, and the board calculates the current speed at which the motors are spinning using the mbed's built-in interrupt functions. Software motor control is handled by a PID controller running on the mbed. The PID

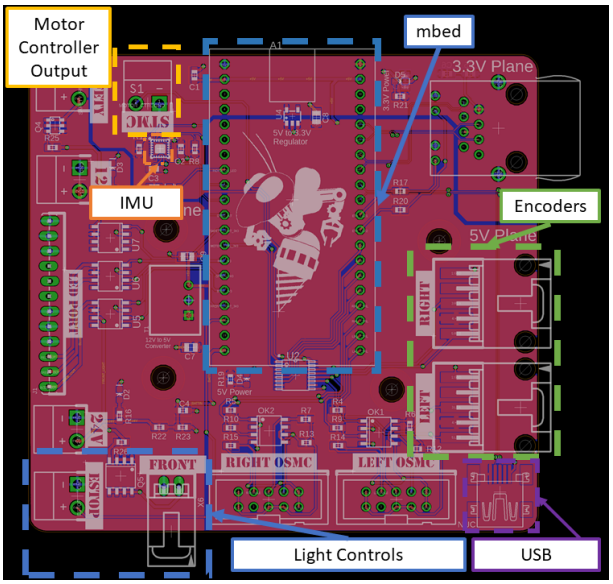



Figure 5: The Motor Board

controller compares the values of the actual speed with the desired speed sent from the NUC and adjusts the ramp-up and oscillation rate to allow the smoothest transition. The calculated speeds are then sent to the motor controller.

In addition to the motors, the motor board controls all lights on the robot. In compliance with the rules of IGVC, the motor board triggers the safety light to flashing when the robot is live and steady when stopped. There is also the added capability for cosmetic lights for use in the future.

In preparation for the future, the motor board contains two features that allow it to evolve with the robot: the addition of an Ethernet port and an on-board IMU. The Ethernet port takes advantage of the mbed's built in Ethernet capabilities, which will eventually allow for much faster communication with the NUC. The on-board IMU is meant to serve as a comparison against the standalone IMU. In the past, the IMU has proven to be somewhat unreliable, so this second IMU would serve as a point of comparison to improve accuracy.

### 4.3.2 Light Detection and Ranging (LiDAR)

Located in the front of the robot, the Scanse Sweep Scanner scans the surroundings located in front of the robot via fast pulses of light on a singular plane. It has a sampling rate of 1000 samples/sec with full 360° field of view. It is powered directly from USB using a USB-to-Serial adapter board as an intermediate. The LiDAR is used to detect physical obstacles such as barrels. We have chosen to use a LiDAR for easy determination of the location of obstacles compared to vision, which requires significant computation to derive location data.

### 4.3.3 Camera

The camera is located at the top of the robot above the lights on the mast. We use the Logitech c920 1080p. As with the LiDAR, it is also powered directly off of USB from the NUC. The camera enables us to capture obstacles that the LiDAR cannot see, most notably the white lines that form a perimeter on the outside of the track. It also allows for software to use color in its path planning and obstacle detection in addition to the points in a point cloud provided by LiDAR.

### 4.3.4 Encoders

Jessi contains two AMT103-V optical encoders, one located on each motor. The optical disks within them each contain 48 ticks per revolution. They communicate directly with the mbed via the motor board and provide the current speed.

## 4.4 Safety Devices

### 4.4.1 Wireless E-Stop

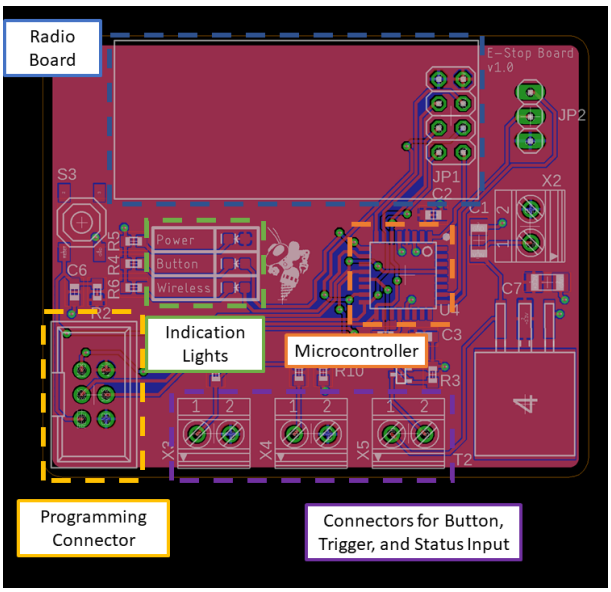



Figure 6: The E-Stop Board

The second of our custom designed PCBs is a wireless E-Stop module. The system is composed of two sister boards that are identical in design but vary in their programming and component placement. By not adding certain connectors, one board is able to trigger a 24V signal used in the E-Stop trigger and the other is used as a button remote. One is located on the robot on the ceiling of the cover with its antenna position located just outside in the front of the robot, while the other is a handheld meant to be used by an operator. All programming is handled by an ATmega328p which is responsible for generating and receiving messages. Two NRF24L01 radios handle the wireless component. They interface with the ATmega328p via a SPI interface. We have tested this system and found that it works reliably up to 500 feet away through walls.

# 5 Software

## 5.1 Overview

Jessi uses a ROS-based (Robot Operating System) system centered around modularity in code, as is common with ROS systems. The software runs on two computers, an Intel NUC and a NVIDIA Jetson TX2, that are connected over Ethernet and communicate using ROS. Functionality is broken into independent processes called nodes. ROS allows for communication between nodes using TCP (Transmission Control Protocol) for asynchronous message passing. The Jetson includes an on-board GPU resource in order to support our new convolutional neural network for line and pothole detection. Our path planning and mapping strategies were entirely overhauled to address significant issues with any localization and detection noise persisting in the global map indefinitely.

## 5.2 State Estimation

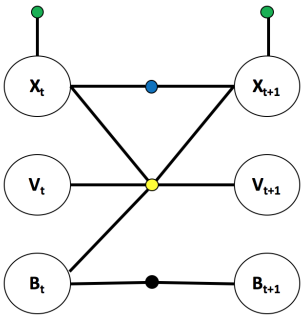


Figure 7: Factor Graph

State estimation in robotics refers to estimating the position, velocity, and orientation of the robot in some globally defined reference frame. This is done by collecting information from sensors on the robot and performing some optimization to find what state is the most likely. Our approach to this problem was to integrate IMU, GPS, and wheel odometry measurements into a factor graph representation and perform trajectory smoothing where the optimization is done using the open-source factor graph optimization library GTSAM.

A factor graph is a bipartite graph whose vertices are divided into factors and variables and whose edges join factors and variables [1]. In factor graphs, each factor represents a factor in the factorization of the joint probability density function where the neighborhood of the factor node is the set of variables involved in that particular factor of the function. Factor graphs are a natural representation of the SLAM or robot localization problem because the observations are treated as givens and the graph only encodes variables that we are interested in [1]. There has been significant work done on the implementation and theory of incremental inference on factor graphs which enables the real-time operation of systems [2, 3, 4].

IMU integration uses a state-of-the-art implementation from GTSAM which is detailed in [5] and [6]. The integrated IMU measurements create a factor in the graph that constrains the previous position, velocity, and bias variables along with the next position and velocity variables. These bias variables allow for online IMU bias estimation. GPS measurements are incorporated in a very simple manner where the measurement is transformed into a local coordinate frame using an initial measurement. This then creates a unary constraint on a position variable. The wheel odometry model is a simple differential drive kinematic model which takes readings from the wheel encoders and computes an estimated velocity. These velocities are then integrated to create a positional constraint between two consecutive position variables.

From the previous state, IMU measurements are integrated to get an initial estimate of the next state which is used as a starting point for the optimization to find the next state. In this way, the estimate can be propagated recursively with this incremental inference.

## 5.3 Map Generation

Last year we made the switch to retaining a global map rather than using only the obstacles detected in the last sensor frame for planning. With a global map any obstacles seen previously remain in the global map of the world. No longer will obstacles detected in the previous sensor frame that are not detected in the current frame be ignored. Unfortunately, the localization and sensor errors compound over time to generate a map that is not traversable by our robot. To compensate for localization and sensor noise, we propose a discretized mapping strategy. The map is made up of a matrix where each grid cell is a small square of the world that is to be treated equally. The values stored in the matrix represent the probability that the grid cell is occupied. Two points that fall into the same square affect the probability of the square equally. All sensor data is transformed from the local reference frame of the sensor into the global map using the global reference frame published by the localization node. This strategy aids in preventing or at least decreasing the rampant smearing that occurs in continuous mapping, which was a constant point of failure in the previous mapper. Furthermore, the new map representation allows for path planning through a grid rather than in continuous space which significantly increases the speed of path planning.

## 5.4 Obstacle Detection

Our obstacle detection is split between vision-based and LiDAR-based perception. LiDAR-based perception covers barrels and other physical objects. The LiDAR returns its reading as a set of points which is projected onto the ground plane and published to other modules as a point cloud. The range of the current LiDAR is 30 meters; however, it only has useful accuracy and density at less than 15 meters. LiDAR works well for the physical objects but clearly cannot work for visual obstacles such as lines and potholes. For these obstacles, the vision-based perception system is used.

The vision system was completely revamped this year in favor of moving to an entirely learning based approach. The first step to any learning based approach is to collect and label a dataset. Images were extracted from recordings of various test runs in environments similar to that of competition. These images were then uploaded to LabelMe where 500 of these images were annotated with polygonal bounding boxes of all objects in the image. The network used was FCN8 [7] which uses the first layers of VGG 16 [8], pre-trained on ImageNet classification [9], which feeds into several layers of transposed convolution, along with a couple skip connections, to create a predicted segmentation. Given the heavy use of transfer learning, the network was able to converge very quickly and robustly, even with few training images. We used an open-source implementation of this algorithm that was developed in [10]. Only two classes were predicted, white paint and background. This proved to be sufficient as potholes and lines share enough similarities. Figure 8 shows an example of the input and output of an image used for testing. The image segmentation from the CNN is then projected onto an assumed flat ground using basic camera geometry. These points are then passed to other modules as PCL point clouds. The range of the detection algorithm depends entirely on the orientation of the camera but in the optimal configuration is close to 10 meters.

Figure 8: (Left) Original input image, (center) heat map of probability of a line from the CNN, and (right) the thresholded output of the CNN. This result is very promising given the lack of false negatives, particularly on the white stripes of the barrel (a previous weak point).

## 5.5 Path Planning

Previously, we planned in kinematic space directly and then followed the path that was generated blindly. This approach is no longer feasible given our occupancy grid-based mapping. Furthermore, planning was a significant hindrance due to its slow nature. If a path was not able to be generated, the robot would continue to follow the previous path to its conclusion, often resulting in collisions with the obstacles that were preventing a new path from being generated. Planning in discrete space is faster than kinematic space but requires a robust path following strategy to smooth out the inherently blocky paths.

The path planner uses A* with distance to reach the current location as the cost and euclidean distance to the goal as the heuristic. Locations in the map are considered obstacles when their probability crosses a configurable threshold. The search algorithm disqualifies these locations as traversable using the

C-Space of the robot. The C-Space of the robot is defined by the shape represented by the grid cells the robot would occupy with an additional buffer to account for error anywhere in the mapping pipeline. The size of the robot and the resolution of the map are independent of each other, therefore if the C-Space would be partially in a square, it is counted as part of the C-Space. If any point within the C-Space of a given cell is not traversable, then that square is labeled likewise. For example looking at Figure 9, if the C-Space of the robot is 2 grid cells and the cell 2X2 is being evaluated (shown in yellow), then all of the red squares will be treated as impassible. The path planned in this environment would be the green squares with the start being (0,0) and the goal being (5,5). The goal location is defined as the grid cell where the waypoint would be located after it is projected into the global reference frame defined in the localization node.

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

Figure 9: Map with path

Our path following strategy is loosely based off of pure pursuit [11]. Our algorithm takes in the current path, the current location from the localization node, and a set offset in distance. A target location is calculated by moving the offset distance along the path. The path follower then computes the wheel speeds that will move the robot to the target location using the differential drive kinematic model. The path follower runs every time a state estimate is published, close to 100 times a second. The real advantage of this implementation is the decoupling of path following from path planning. Corrective action will now take place between paths being generated and is only limited by the publish rate of the localization node.

# 6 Failure Points

## 6.1 Vibrations

Vibration-induced failure will occur over the lifetime of the robot. While the robot runs, any slight perturbation from the rough terrain will induce vibrations that wear out the fasteners. This occurs quicker with 80/20 compatible fasteners than conventional fastening methods; however, we properly torqued the fasteners and then added high-strength Loctite to alleviate these issues. Replacement fasteners and Loctite will be brought to competition to use in case the threads loosen.

## 6.2 Wear Damage

Wear damage on the encoder mounting assembly will also occur while the robot runs after long periods of use. The encoder mount involves connecting a shaft rotating at 3,700 rpm to a ball bearing underneath the encoder. This shaft hangs out the side of the motor, so no load is applied by the robot on the bearing or encoder. The wear damage comes from any perturbations introducing internal loads to either the bearing or encoder as well as general contact between the races of the bearings. A transition fit with the help of retaining compound keeps the ball bearing in the mounting plate without the bearing experiencing any radial load from an interference fit or axial loads from the shaft. We chose a bearing well above these specifications to avoid wear damage. The pre-lubricated and double shielded bearings

can withstand up to 120 lbs of radial load andm 48,000 rpm. We purchased replacement bearings, extra retaining compound, and a spare encoder in case of wear-induced failure.

### 6.3 State Estimator Converging To Local Minima

Since the state estimator is optimization based it can get stuck in a local minimum and then throw off subsequent iterations due to its iterative nature. This results in a suboptimal state estimate that is locally but not globally consistent. In this situation, the robot can still navigate but with significantly reduced map accuracy

### 6.4 Failure To Handle Loop Closure

Currently our code base does not handle loop closure, the process of realigning the map after seeing the same obstacles again. This is often not a large problem since the IGVC course is not designed with loop closure in mind but can occur in No-Mans Land. The failure would be a smeared map where points from seeing the exact same obstacles could appear in different locations due to accumulation of error without the correction loop closure would provide.

## 7 Simulations

### 7.1 Software

Our team is a proponent of Gazebo, a simulation software that allows for comprehensive software testing and prototyping within a ROS. Gazebo requires a 3D rendering of the robot along with annotated joints to allow for acuation. This information is detailed on a joint by joint basis in a URDF file along with additional gazebo world configurations. Every sensor used on the physical platform can be simulated and publish its data in the same format as the physical platform.

### 7.2 Finite Element Analysis

In order to ensure Jessis structural integrity, we constructed a Finite Element Analysis for the base under the robots static conditions. The loads were the weight of the robot from above the base, the weight of the batteries, the weight of the payload, and gravity. The simulation assumed the surfaces in contact with the motors and caster wheel to be fixed. The mesh size were kept as the default from Autodesk Inventors simulation software. The max stress from the analysis was 13.31 kilopounds per square inch (ksi). The yield stress for aluminum 6061-T6 is at least 40 ksi, resulting in a safety factor of 3.01. The analysis also showed that fatigue failure was highly unlikely. Even if the frame were to experience the maximum stress uniformly throughout as the amplitude of the repeated loading, the average fatigue failure would occur on the order of $10^7$ cycles of loading. In the case that the analysis is incorrect and the frame fatigues, the ability to remove individual members easily allows failed parts to be replaced and reinforced.

# 8 References

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," 2015.

[2] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," CoRR, vol. abs/1409.1556, 2014.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009, pp. 248-255: IEEE.

[4] M. Teichmann, M. Weber, M. Zollner, R. Cipolla, and R. Urtasun, "MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving," 2016.

[5] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation, in Robotics: Science and Systems 2015, 2015.

[6] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, On-manifold preintegration for real-time visualinertial odometry, IEEE Transactions on Robotics, vol. 33, no. 1, pp. 1 21, 2017.

[7] F. Dellaert, Factor graphs and gtsam: A hands-on introduction, 2017.

[8] M. Kaess, A. Ranganathan, and F. Dellaert, Isam: Incremental smoothing and mapping, IEEE Transactions on Robotics, vol. 24, no. 6, pp. 13651378, 2008.

[9] M. Kaess, V. Ila, R. Roberts, and F. Dellaert, The bayes tree: An algorithmic foundation for probabilistic robot mapping, in Algorithmic Foundations of Robotics IX: Selected Con- tributions of the Ninth International Workshop on the Algorithmic Foundations of Robotics, D. Hsu, V. Isler, J.-C. Latombe, and M. C. Lin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 157173, ISBN: 978-3-642-17452-0.

[10] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, Isam2: Incre- mental smoothing and mapping using the bayes tree, The International Journal of Robotics Research, vol. 31, no. 2, pp. 216235, 2012.

[11] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm", DTIC Document, 1992.