Georgia Institute of Technology
RoboJackets

Jessii

Project Manager
Dallas Downing | dallasd@gatech.edu
Software Subteam Lead
Alejandro Escontrela | aescontrela3@gatech.edu
Mechanical Subteam Lead
Yongjae Won | ywon30@gatech.edu
Electrical Subteam Lead
Ryan Waldheim | raw@gatech.edu

Members

Mechanical

Tomas Osses	tomas_osses@gatech.edu
Daniel Kilgore	dkilgore8@gatech.edu
Cameron Loyd	cloyd6@gatech.edu
Charles Li	cli651@gatech.edu
Akhil Sadhu	asadhu7@gatech.edu
Ryota Tsutsumi	rtsutsumi3@gatech.edu

Electrical

Alex Xu	rxu74@gatech.edu
Asha Bhandarkar	abhandarkar6@gatech.edu
Garrett Botkin	gbotkin3@gatech.edu
Noah Daugherty	noahpd@gatech.edu

Software

Justin Zheng	jzheng84@gatech.edu
Oswin So	oswinso@gatech.edu
Jason Gibson	jgibson37@gatech.edu
Sam Roquitte	samr@gatech.edu
Joshua Viszlai	viszlai@gatech.edu
Yosuke Yajima	yyajima@gatech.edu
Aaron McDaniel	amcdaniel39@gatech.edu
Kyle Keirstead	kkeirstead3@gatech.edu
Thai Tran	thaitran@gatech.edu

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

May 15th, 2019

1 Who We Are

1.1 Introduction

RoboJackets is the student organization for competitive robotics at the Georgia Institute of Technology. The Intelligent Ground Vehicle Competition (IGVC) team is one of the four competitive teams in RoboJackets. This year we wanted to improve on the designs of our robot from last year, Jessi, which placed second in the 2018 design competition. This year's robot, Jessii, is an iteration on Jessi's design, focusing on accessibility for users, safety improvements to the electrical system, and increased effectiveness of our software. Jessii features an improved E-Stop system and electrical monitoring system (Section 4.4). Our software uses an improved path planning algorithm (Section 2.1) and vision system (Section 2.2). This year she has a 3D LiDAR and three different cameras to improve obstacle detection (Section 5.2). She also uses a custom-built computer with increased computing power (Section 4.3.4). With these design improvements, Jessii is sleeker, quicker, and smarter than Jessi.

1.2 Organization

The RoboJackets team consists of three subteams and a project manager. The subteams correspond to the major disciplines of IGVC: mechanical, electrical, and software. Each subteam has a subteam lead, responsible for coordinating the details of the team's technical work and introducing new members to the team. Over the course of the 2018-2019 year, the team put in over 3000 hours working on Jessii. The chart below shows all our members, their area of work for the team, their class standing, and their major.

Position	Name	Standing	Major
Project Manager	Dallas Downing	Junior	Computer Science
Mechanical Lead	Yongjae Won	Junior	Mechanical Engineering
Mechanical	Tomas Osses	Junior	Mechanical Engineering
	Daniel Kilgore	Junior	Aerospace Engineering
	Kelvin Chong	Senior	Mechanical Engineering
	Cameron Loyd	Junior	Mechanical Engineering
	Charles Li	Freshman	Aerospace Engineering
	Akhil Sadhu	Freshman	Mechanical Engineering
	Ryota Tsutsumi	Freshman	Mechanical Engineering
Electrical Lead	Ryan Waldheim	Senior	Chemical Engineering
Electrical	Alex Xu	Junior	Computer Engineering
	Asha Bhandarkar	Freshman	Computer Engineering
	Garret Botkin	Freshman	Electrical Engineering
	Noah Daugherty	Junior	Electrical Engineering
Software Lead	Alejandro Escontrela	Sophomore	Aerospace Engineering
Software	Jason Gibson	Graduate	Computer Science
	Oswin So	Freshman	Computer Science
	Sam Roquitte	Freshman	Computer Science
	Justin Zheng	Senior	Computer Engineering
	Joshua Viszlai	Junior	Computer Science
	Yosuke Yajima	Graduate	Mechanical Engineering
	Kyle Keirstead	Freshman	Computer Science
	Thai Tran	Sophomore	Computer Science
	Aaron McDaniel	Junior	Computer Science

1.3 Design Process

Our design process for Jessii started with analyzing Jessi's successes and failures at the 2018 competition. The mechanical design was good, but suffered from lack of access to internal components and general bulkiness, making it difficult to service and transport. Thus, we wanted Jessii to be more modular and adjustable. We created a CAD model for Jessii that was continually developed throughout the year. The design also underwent two design reviews from RoboJackets members outside of IGVC to ensure it did not have any major flaws.

The electrical system underwent a similar design process. While there were no major failures at the 2018 competition, the system's design lacked resiliency. Thus, the design objective this year was to improve on the durability of the system through safety and diagnostic mechanisms, working with the mechanical team to create a more modular design, and implementing features requested from the software team. Throughout the year, members of the team designed PCBs to implement these features. Design reviews were held with other RoboJackets members in attendance to get feedback on the design and to point out potential improvements and weaknesses.

For our software, we spent several weeks in the fall reviewing the existing code base, trying to identify bugs and areas that needed more attention. The overall conclusion was that our localization and mapping strategies needed to be entirely overhauled for increased robustness. We implemented new algorithms with a foundation in probabilistic theory in order to handle sensor noise. We used GitHub to plan and review any changes made to the code.

2 Innovations

2.1 Field D*

To plan a path from Jessii's current position to the goal waypoint, Jessii makes use of the Field D* Any-Angle Path Planning algorithm [1]. Field D* can plan smooth and continuous paths which are often easier executed by differential drive robots. The smooth path generated by Field D* results in more optimal paths since it does not have to follow the restrictive state space enforced by standard grid search formulations. For these reasons, the Field D* algorithm was utilized by the recently-decommissioned Opportunity Rover for use in navigation across the Martian surface.

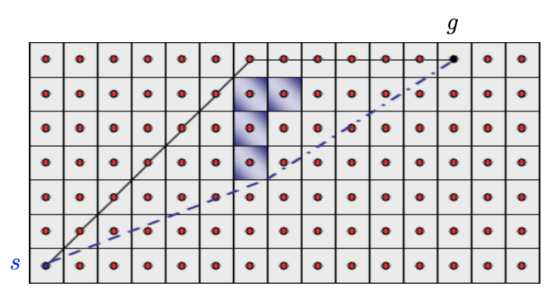

Figure 1: Difference between a path generated by a traditional path planner and one generated by Field D*. The solid line represents a path generated by A*. The dashed line indicates a continuous path generated by Field D*. Figure from [1]

Aside from planning paths that are easily navigated by differential drive robots, the Field D* algorithm makes search more efficient by reusing information from past searches. That is, instead of planning from scratch at each planning interval, the algorithm defines an underlying cost map in the form of a grid, where each cell represents an area on the map with a corresponding traversal cost. Upon the first iteration of planning, the algorithm performs a heuristic search with a cost function that allows for continuous changes in angle. The costs computed in the first iteration are stored in the cost map. This cost map is updated and re-used in subsequent planning steps resulting in highly efficient re-planning operations, valid in both static and dynamic environments.

Paired with the occupancy grid mapper, the Field D* path planning algorithm produces smooth and continuous paths through terrain unlikely to be occupied.

2.2 Convolutional Neural Network

Jessii uses a state-of-the-art U-Net Convolutional Neural Network for Image Segmentation [3], an architecture based off of the FCN (fully convolutional network), which we used last year, but modified in order to work with less training images and segment images much faster. Similar to the FCN architecture, pixel-wise classification is done as opposed to other techniques such as the sliding-window method. The key changes made from FCN are the U-shaped architecture which gives the neural network its name, where the typical contracting network is supplemented by an upsampling network. Importantly, the upsampling portion of the network has a large number of feature channels from the contracting portion, which helps the network propagate context information to higher resolution layers. Our use of this architecture is covered in more detail in section 5.2.

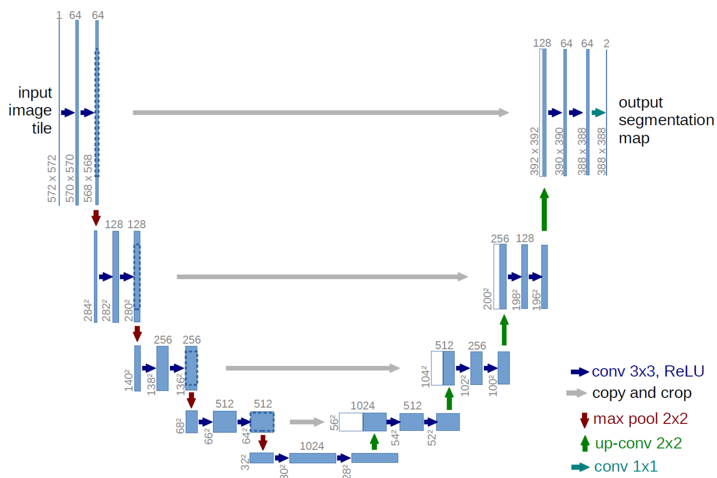

Figure 2: The architecture of the U-Net with the distinctive U-shaped layers. Figure from [3]

2.3 3D Image Projection

To obtain accurate projections of segmented images for mapping, Jessii computes directly the location of each pixel in the image by interpolating between 3D LiDAR scans of the ground, yielding a better estimation of the position of each pixel in the world when compared to a flat ground plane assumption or when fitting a ground plane.

3 Mechanical Design

3.1 Overview

Our main objective this year was to make a modular, easy-to-service robot. After producing an easy to manufacture robot in the previous year, we turned our attention to ease of use and design flexibility. Jessii's design consists of three parts: the base, the electronics tray, and the mast. The base supports the weight of the robot and houses the payload, batteries, and drive train. These components need to be low on the frame to prevent tipping and provide a low center of gravity for stability. The electrical systems are mounted on the electrical tray, which is located under a water-resistant, slide-out cover. The cover is also able to be completely removed if extensive electrical work needs to be done in the body of the robot. The mast refers to the top portion of the aluminum frame that supports the monitor, GPS antenna, IMU, cameras, safety light, flood light, and button box. The mast is collapsible and removable for ease of transportation. The robot's structure is mostly built out of 8020 Inc.'s T-slotted aluminum framing, which was chosen for its ease of assembly and manufacture.

3.2 Modularity

3.2.1 Sensor Mounts

In order to provide a convenient testing platform for the software subteam, we designed customized mounts for the cameras and the LiDAR. These mounts are easy to swap out if changing the angle or position of the sensor is necessary. The cameras can be rotated either horizontally or vertically to optimize Jessii's field of view. They can also move laterally along the top of the mast. The LiDAR mount was manufactured at a 10 degree angle from the horizontal so the robot can better detect the ground plane. The two side cameras were added to aid in detecting lines to the side of the robot. The cameras are covered with a 3D printed shroud that prevents moderate rain from reaching the cameras and reduces glare from the sun.

3.2.2 Mast Connectors

To allow fast and easy removal and reattachment of the modular mast, every wire that runs up to the mast connects to the electronics tray via panel mount connectors installed on the bottom of the electronics tray. Since the panel mounts are at the bottom of the tray, they are resistant to rain. This design also allows the entire top and sides of the cover to slide out and makes the mast much easier to remove.

3.2.3 Sliding Cover

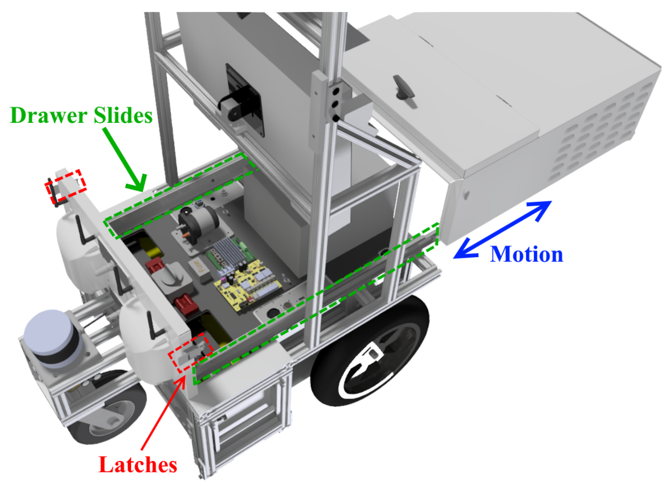

Figure 3: Sliding cover in the fully opened position.

Last year's cover was fixed to the chassis, making it difficult to service the electrical system even with access panels. To accommodate the electrical and software members, the entire cover can slide out from under the mast to reveal the electrical tray. For user safety, latches engage in the half-open and fully-open positions, preventing accidental closures. In the fully closed position, the sealing foam is compressed and the cover is held in place with two locking draw latches that require simultaneous pushing and pulling actions to open. This reduces the chance of accidental opening. If needed, the cover can be removed entirely. This is achieved with locking drawer slides inside the cover. There is a quick-release tab at the end of the drawer slides that allow them to be decoupled completely, resulting in the cover disconnecting from the electronics tray and base assembly. We also kept the original access panels in place to make it easier to perform quick electrical fixes.

3.3 Weatherproofing

Since the robot is intended for outdoor navigation, we decided it should have basic resistance to rain and other inclement weather. Our goal was to make the robot comply with IP43 for resistance to dust particles at least 1 mm in size and liquid falling from up to 60 degrees from vertical. To achieve this goal while still maintaining adequate cooling and ventilation for the internal electronics, we put covers on the input fans so they can only draw air from below. The fan cover was designed to reduce turbulence and allow maximum airflow while still fitting in a constrained area. The fans draw in cool air from the front, and the hot air exhausts through the louvers on the sides of the cover. The downward facing opening of the louvers prevents water entry from moderate rain. In case some amount of water gets in, there are drains on the bottom of the side walls of the cover.

The two access panels on the cover are designed to be easily operated while still being weatherproof. To achieve this, we put ethylene propylene diene monomer (EPDM) rubber foam seals on the inside of the access panel and installed aluminum lips on the mating surface of the cover. The lips compress the foam and form a watertight seal once the access panel is closed. The access panels have clamping latches to ensure adequate sealing. The external sensors were chosen to be water-resistant whenever possible; however, the cameras, the IMU, and the monitor required weatherproofing. The camera mounts block out rain by surrounding the camera and having openings for the lenses and the wires at an angle of at least 60 degrees from vertical. For the IMU, we 3D-printed an enclosure to block out water. The monitor is encased in plexiglass so that the rain cannot interfere with the electronics.

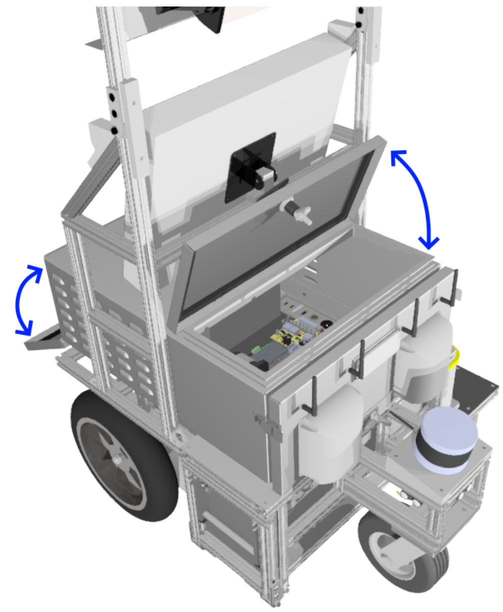

Figure 4: Both access panels opened, showing the EPDM foam rubber seals and the lips.

4 Electrical Design

4.1 Overview

The electrical design of Jessii focused on addressing various objectives decided after last year's competition. A major objective of Jessii's electrical system was to create a more resilient system through various fail-safe mechanisms, remote diagnostics, and more robust components. Thus, various components were upgraded, a new custom PCB focused on providing live diagnostic information was included, and an improved E-Stop system was implemented. Another objective was to increase the modularity of the design. We accomplished this by optimizing the PCB designs to reduce the electronics footprint size. Other notable improvements to the design include implementing features beneficial to the software team including the construction of a custom computer with hotswap capability.

4.2 Power Distribution

The main power of the robot is a 24V system supplied using two 12V lead-acid batteries in series. An internal disconnect (rated for 180A) switches the battery power. The motors, GPS receiver, computer, and flood light are powered from this main rail. The 24V rail is stepped down into 12V and 19V rails using two voltage regulators.

The majority of components are powered from 12V, while the monitor is powered from 19V. The encoders are powered from the logic board.

A new feature added to the power distribution system this year was hotswap capability to allow uninterrupted use of Jessii's computer system while transitioning from battery power to an external source and vice versa. A LTC4416 microcontroller controls two sets of MOSFETs to create a power switching circuit. The microcontroller continuously monitors the status of two power sources: the primary (the power from an external supply, such as an outlet) and a secondary (the batteries on board the robot). When the primary power source is present, it is used to power the computer while disconnecting the computer from the secondary. If not present, then the secondary source is used to power the computer.

4.3 Electronics Suite

4.3.1 Logic Board

The main printed circuit board of the electrical tray is the logic board. The board is the primary interface between the electrical and software systems. The secondary use of the board is to maintain the state of the robot. It does this by reading the status of the E-Stop and handles the encoder interrupts to calculate estimated wheel speeds. It then takes this information and other status variables and relays them to the computer. The software stack uses this information to inform autonomy decisions and outputs the resulting motor commands for the logic board to execute. Furthermore, the logic board implements a watchdog timer that will automatically stop the motors if they have not seen a new command in a given period of time. This system also controls the safety light to decouple that from a potential software crash, or computer brownout.

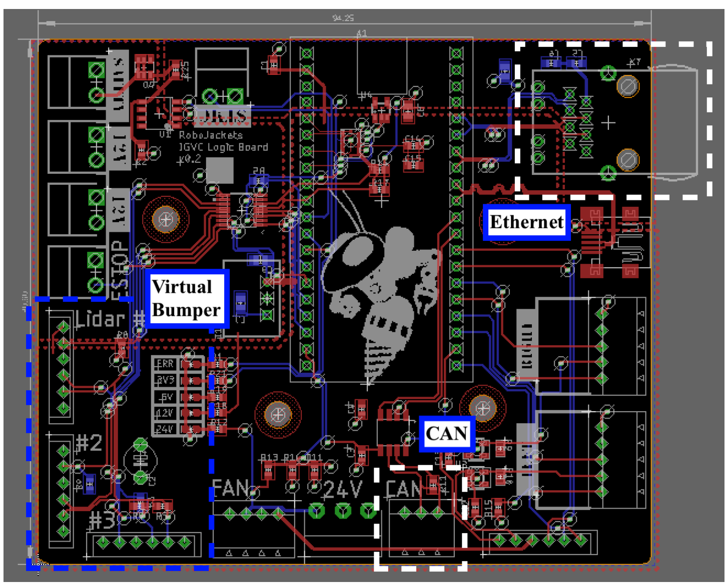

Figure 5: The Logic Board

The logic board handles the control of motors and maintains the speed using a combination of a feedforward and feedback controller

$$u = u_{feedforward} + u_{feedback} \tag{1}$$

A block diagram of the control loop is shown in Figure 6. The feedforward controller is implemented with a linear model of the motor

$$u_{feedforward} = K_v \cdot SP \tag{2}$$

where K_v is the feedforward coefficient, and SP is the set point. K_v was found by performing a linear fit on a plot of the velocity from the encoders and the motor commands. The feedback controller is implemented using a PID controller

$$u_{feedback} = K_p e(t) + K_i \int_0^t e(t')dt' + K_d \frac{de(t)}{dt} \tag{3}$$

where K_p, K_i and K_d are the gains for the proportional, integral and derivative terms respectively, and $e(t)$ is the error term $e(t) = PV - SP$, with the process variable, which is the velocity of the motors, measured by optical encoders mounted on the wheels.

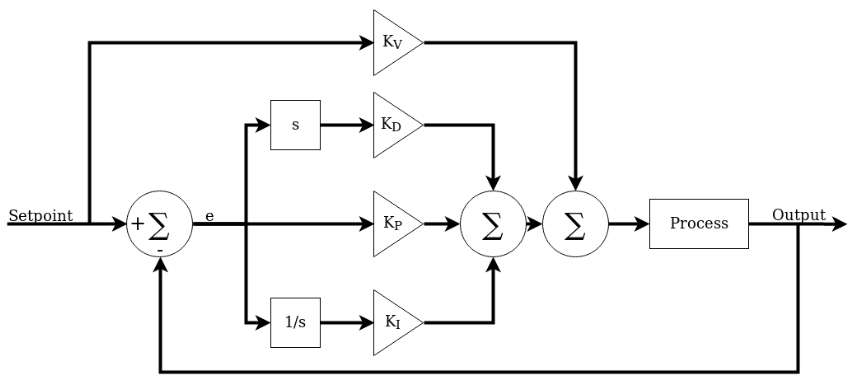

Figure 6: A block diagram of the control loop on Jessii, with the top being the feedforward path and the bottom being the feedback PID path.

The feedforward path was added to the motor control loop in order to reduce the response time of the system, as the slow response of the system was an issue that deteriorated the ability of the robot to follow paths.

The logic board is an iteration on the motor board on our previous robot Jessi. Previously the motor board handled communication via a virtual Serial interface utilizing Mbed's USBSerial library. We decided to switch to a custom Ethernet communication protocol for robustness as it handles the opening and closing of communication windows better.

In preparation for the future, the logic board contains an additional feature for it to be the testing ground of electrical system evolution: a CAN bus controller. The CAN bus is a bus standard designed to allow microcontrollers and devices within a vehicle to communicate without a host computer. This implementation will prevent a shutdown of the entire microcontroller communication network when one of the nodes dies. This feature is intended to support a wider diagnostic system with more custom made PCBs monitoring robot runtime status.

4.3.2 Diagnostic Board

The diagnostic board is a new addition to the electrical system of Jessii. It is a custom-designed and assembled PCB aimed at logging robot runtime state as well as live diagnostic information. It is capable of determining the connection of encoders by monitoring the voltage drop across a shunt resistor. This was implemented due to an encoder failure in the past that was difficult to debug.

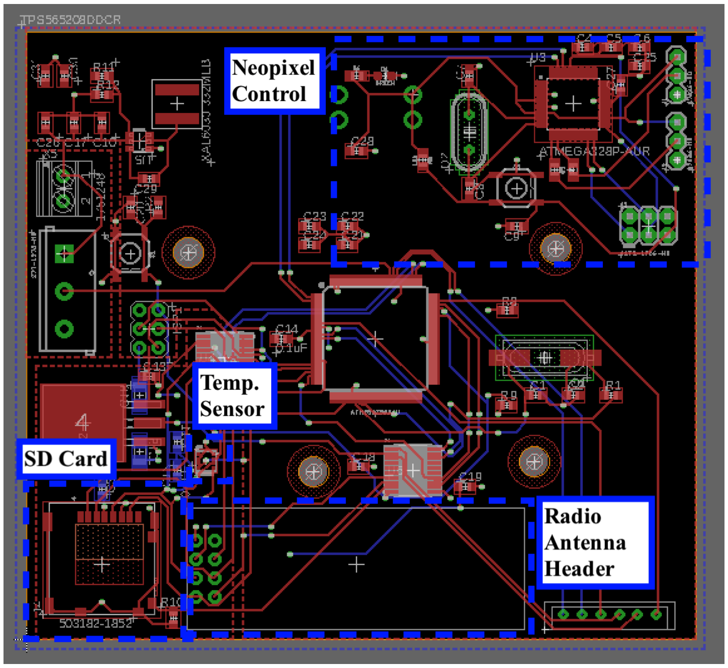

Figure 7: The Diagnostic Board

The diagnostic board also monitors the temperature in the electrical tray and adjusts the speed of the ventilation fans accordingly. Finally, the diagnostic board also interfaces with a Hall effect current sensor to monitor the overall current draw of the robot. It can log the data into a local SD card slot or send to a remote workstation via the onboard radio antenna. It further functions as a control center for the Neopixel underglow which in the future will reflect various states, for example, if there is a certain error or if the robot recognizes an obstacle.

4.3.3 Sensors

Jessii utilizes several different sensors for obstacle detection and localization. The Velodyne Puck VLP-16, a 3D LiDAR, is used for primary obstacle detection. Three Logitech c920 1080p cameras on the top of the robot are used for detecting lines. For localization, Jessii uses the Hemisphere R330 receiver with an A21 antenna for the GPS. This year, we have chosen to use two YostLabs 3-Space Micro USB IMUs: one on the mast to minimize electromagnetic interference for magnetometer readings, the other between the motors to minimize lever arm effect on accelerometer and gyroscope readings.

Jessii also contains two CUI 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 logic board and provide the current speed of each motor.

4.3.4 Custom Computer

To improve Jessii's computational capabilities, we added a custom built computer to replace the previously used one. The computer has an Intel Core i7-8700 processor and a Nvidia GTX 1060 GPU along with 32GB of RAM, which provides a significant upgrade in computational power compared to previous years. See Section 5.1 for more details.

4.4 Safety Devices

4.4.1 Wireless E-Stop

The wireless E-Stop module is another custom designed PCB on the robot. For this iteration, while the functionality of the system remains, all E-Stop circuitry is now on the PCB, increasing the modularity of the design. Two sister boards, differing only in their programming and component placement, make up the system. With one lacking certain connections, one board, located in the electronic tray is able to trigger a 24V signal used in the E-Stop. This connects to an antenna on the front of the robot. The other board is in a 3D printed remote case with a stop button. An ATmega328p responsible for generating and receiving messages handles the programming while the wireless component is handled with two NRF24L01 radios which interface with the ATmega328p through SPI (Serial Peripheral Interface). The board was adopted this year, with its chief difference being a new relay component the engage the solenoid to handle the fly-back voltages generated during state transitions. Another noticeable change is the addition of the previously off-board hardware-implemented state keeper that allows for the simultaneous use of a E-Stop button with a push button start, without the need of a microcontroller.

Each E-Stop system does not override the other. If the physical switch is disabled and the remote on is enabled the entire system is disabled and vice versa. This requires both E-Stop operators to entirely agree that it is safe to enable the robot before it can move.

4.4.2 Preemptive E-Stop

The virtual bumper system is a new safety measure added to Jessii this year. The bumper is a fail-safe redundant system aimed at providing obstacle proximity alert separate from the software navigation pipeline. The system comes into action when an obstacle comes too close to the robot for it to avoid. In our implementation, this is our preemptive E-Stop system, in case the human operator cannot stop Jessii before a collision.

The system is implemented using 3 Optical Distance Measurement Sensor (LIDAR-Lite v3HP), installed in the front of the robot with an angular separation of 30 degrees. The sensor has a resolution of 1 cm and an update frequency of 1kHz. The logic board communicates with the sensors via I2C protocol (see logic board figure).

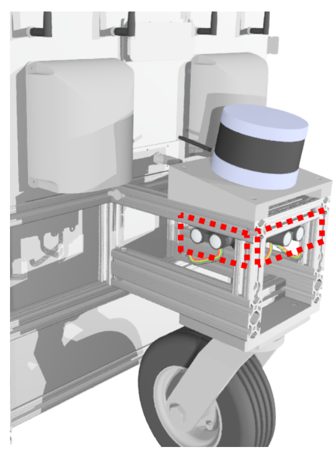

Figure 8: The location of the optical sensors for the preemptive E-Stop system.

5 Software Strategy

5.1 Overview

The software stack on Jessii is written in Robot Operating System (ROS), a message-passing robotics middleware that lends itself to clearly defined interfaces between independent processes called nodes. ROS runs all nodes simultaneously and uses a publisher subscriber architecture to transfer data. Since ROS controls interface between nodes we can use multiple languages depending on the framework desired. For example, our neural network model runs in Python using PyTorch, while the rest of the code base uses C++. A flow diagram showing the various nodes in our system and how they interact with each other is shown in Figure 9.

Our computer is running an Intel i7-8700K 3.7GHz 6 core processor, 32GB of RAM, and an Nvidia 1060 GPU. The Nvidia GPU was picked to allow CUDA support, a parallel computing library that allows using the GPU directly. This is required to run our neural networks in real time. The network sits on the GPU and runs the three different cameras in parallel. Our goal when designing this platform was to optimize for GPU processing since that was the bottleneck of the software stack previously.

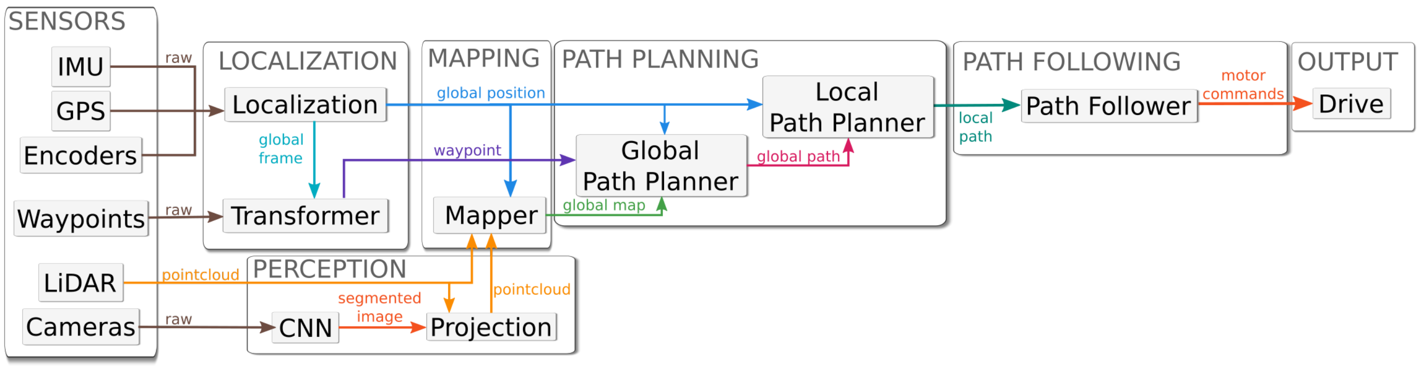

Figure 9: A flow diagram of the ROS nodes in Jessii.

Jessii uses improved path planning and mapping strategies, allowing the robot to efficiently plan and execute smooth global paths while keeping robot constraints in mind, as well as decrease the impact of projection and localization errors on the validity of the global map.

5.2 Obstacle Detection

Our obstacle detection is split between 3D LiDAR-based perception and vision-based perception.

5.2.1 3D LiDAR Detection

The 3D LiDAR has a range of 100m, with 30 degrees of vertical field of view giving Jessii the ability to detect obstacles even when on an inclined surface, as well as be able to detect the ground surface for projection. While detecting barrels and other 3D obstacles was the main purpose of the LiDAR in years past we wanted to extend to a 3D LiDAR to improve with our mapping strategy. The LiDAR was pitched 10 degrees forward in order to give a better view of the ground directly in front of us and allow us to project camera images onto the actual plane rather than making any assumptions.

5.2.2 Camera Based Detection

For detection of lines and potholes, this year we made use of the state-of-the-art U-Net CNN for Image Segmentation instead of a pretrained FCN8 to reduce model size and increase efficiency to cope with the increased loads of three total cameras. The model is general enough to detect both potholes and lines, as the two features are similar. Similar to the FCN8, the U-Net outputs a pixel-wise classification that can be visualized as a greyscale image (see Figure 10). The image segmentations are projected using a technique described in Section 5.4, and the resulting point cloud is used for mapping.

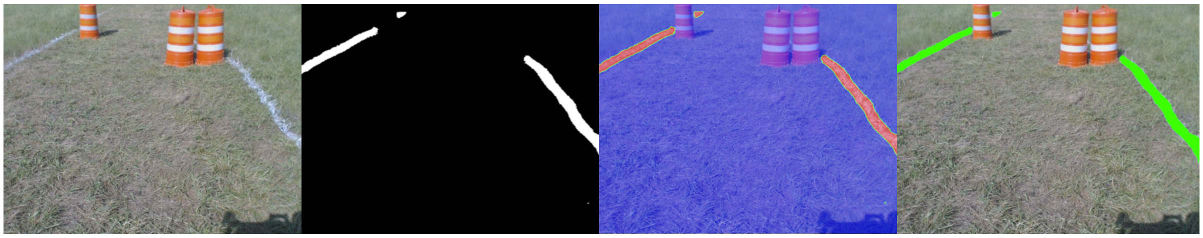

Figure 10: From left to right: Original image; Output of the neural network (minimum value of 0 is black, maximum value of 1 is white); Output of the neural network overlaid on the original image as a heatmap; The thresholded output of the neural network.

One problem previously faced was the blind angles to the immediate sides of the robot. To solve this, two side cameras pitched aggressively supplement the main detection camera by covering its blind angles, preventing the robot from driving too close to a line with an incorrectly estimated position.

5.3 Path Planning

Our path planning is split into global planning and local planning in order to efficiently generate paths that are locally feasible for Jessii but also globally accurate in a computationally efficient manner.

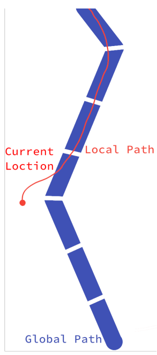

Figure 11: Global path from Field D* and the local path resulting from the control law along subsections of the global path

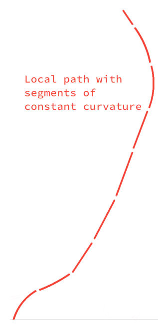

Figure 12: Local path with segments of constant curvature used as the input for motion profiling

This year, we have revamped our global planner to use the Field D* algorithm, an incremental search algorithm that generates smooth paths by generating paths that travel along edges of discretized grid cells. Field D* performs linear interpolations along edges of grid cells in order to obtain smoother paths compared to traditional search algorithms state formulations that reduce complexity by reducing the action space to transitions in a grid. This is done by assigning nodes to the corner of grid cells instead of the center and treating the cost of a point along the edge as a linear interpolation between the two surrounding nodes. Doing this results in three different potential optimal paths to traverse across a grid cell (see Figure 13). With this modification, the cost of any point within the grid cell can be calculated, and overall path costs are reduced, resulting in paths that can enter and exit nodes along any point on the edge of each node and not just through the four corners.

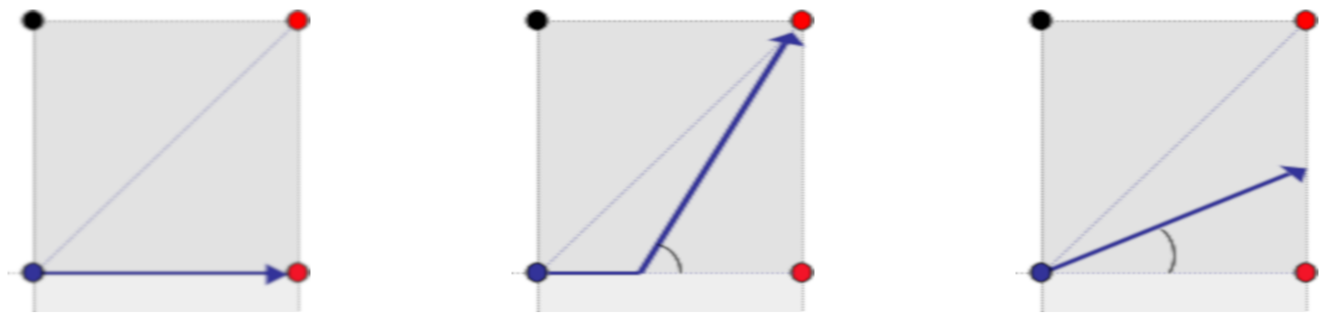

Figure 13: Possible optimal paths from the blue node to the rightmost edge. Figure from [1]

For our local planning, Jessii improves upon the differential drive control law used last year by switching to a Lyapunov stable path planner — where the robots distance from the path is guaranteed to be non-increasing – and performing motion profiling. For a given global path generated from global planning, the local planner first subdivides the global path into small sections, then uses a differential drive feedback based control law in order to generate a Lyapunov stable local path that is also feasible for the kinematics of a differential drive robot (see Figure 5). Afterwards, the generated local path is divided into tiny segments of constant curvature. The motion profiler then takes into account the current velocity and wheel acceleration constraints of Jessii and calculates a set of target velocities for each segment, ensuring that the robot is able to physically execute the planned trajectories (see Figure 12).

5.4 Map Generation

One of the largest improvements from last year was to mapping. Jessii improves on previous mapping techniques by performing sensor fusion between our vision and LiDAR based perception using an occupancy grid and exploiting the sensor model of each in order to extract information about both the presence and absence of obstacles.

An occupancy grid map is a grid-based map where each cell contains a probability of occupancy and assumes that occupancy probabilities are independent, obstacles are static, and the robot pose is known. Under these assumptions, the ratio between the occupied and unoccupied probabilities for a grid cell given sensor information regarding the occupancy of the grid cell and the pose of the robot is:

$$\frac{p(m_i|z_{1:t}, x_{1:t})}{1 - p(m_i|z_{1:t}, x_{1:t})} = \frac{p(m_i|z_t, x_t)}{1 - p(m_i|z_t, x_t)} \frac{p(m_i|z_{1:t-1}, x_{1:t-1})}{1 - p(m_i|z_{1:t-1}, x_{1:t-1})} \frac{1 - p(m_i)}{p(m_i)} \tag{4}$$

Where x is the robot state, m is the occupancy of the grid cell, and z is the sensor observation of the occupancy of the grid cell. This can be simplified by using log odds notation, where

$$l(x) = log\frac{p(x)}{1 - p(x)} \tag{5}$$

This turns the previous products of ratios into sums, which is more computationally efficient:

$$l(m_i|z_{1:t}, x_{1:t}) = l(m_i|z_t, x_t) + l(m_i|z_{1:t-1}, x_{1:t-1}) - l(m_i) \tag{6}$$

We perform RANSAC (Random Sample Consensus)[2] on the set of points which our 3D LiDAR returns to estimate and extract the ground plane. The points left after ground plane extraction represent obstacles such as barrels and are projected onto our 2D occupancy grid. In addition to mapping detected obstacles, we also exploit the LiDAR's sensor model and take into account angles where obstacles are not detected to infer areas that are unoccupied.

For our vision based projection, instead of relying on a flat ground plane assumption as we have done previously, pixels that are detected in our neural network are projected onto interpolated LiDAR scans of the ground, largely eliminating projection errors that would have resulted in an inaccurate global map. Pixels which are not detected to be lines are also projected in the same way but marked as areas where lines are absent. By incorporating both pieces of information in a probabilistic manner, Jessii is able to effectively deal with sensor noise, projection errors and mapping errors due to bad localization. This addresses the issues of the map being biased towards occupancy, which was one of the biggest problems in mapping we faced last year.

6 Failure Points

6.1 Self-Loosening

Standard operation of the robot results in vibrations in its frame. This poses a problem because the 8020 system relies on friction fasteners, which can easily become loosened by excessive vibration. To preemptively solve this problem, we applied threadlocker to every threaded fastener used on the chassis and mast of Jessii. We will have a service toolkit containing Allen keys and threadlocker on the robot itself during competition to quickly fix any loose fasteners.

6.2 Sensor Biases

While the IMU does perform online bias estimation of the magnetometer, this takes a non-negligible amount of time to converge. Also, the localization module does not perform online bias estimation for the gyroscope or accelerometer, which can result in our heading estimates drifting over time. Addressing this failure point would require keeping track of biases either with another Kalman Filter, or moving to another method of localization such as factor graphs.

6.3 Localization Robustness

Our localization module has an omnidirectional motion model. As such, in the event of significant errors in the IMU orientation, our localization module will allow for motions such as horizontal movement which are impossible given the non-holonomic constraints of a differential drive robot. However, given that we have a source of absolute position from the GPS and a relatively accurate accelerometer from the IMU, the Extended Kalman Filter (EKF) running in our localization module is able to compensate for a faulty IMU with accurate positions and accelerations.

6.4 Slippage

Jessii currently is not able to recognize and handle wheel slippage and will continue to power the wheel that has lost traction instead of decreasing torque to increase traction. This could result in Jessii being unable to regain wheel traction and getting stuck. To address this failure point, current sensing on the logic board could be used to detect when wheel slippage occurs to allow the robot to handle this situation.

6.5 False Positives in Neural Network

The output of the new neural network results in some false positives when used on the side cameras due to the different angle in which the cameras are mounted, which could result in sections of the map which are falsely labeled as lines and thus impassable if the false positives were consistent. Handling this resolution plan would require training the neural net with more data of images from the side cameras since the current neural net does not use a pretrained model. This failure point most likely comes from a lack of data used.

6.6 Failure to Handle Loop Closure

Jessii still does not have the capability to handle loop closure, which is the process of using information obtained from the same place at two different times to extract information about the accumulated error between the first and second visits. This failure could result in an error such as misalignment of the lane lines when returning to the starting location at the end of the run, which could negatively impact our path planning. Solving this would require moving to a different method of localization that supports loop closure, such as particle filter SLAM (Simultaneous Localization and Mapping) or factor graph SLAM.

7 Simulations

7.1 Gazebo

Gazebo, a 3D robot simulator, is an integral part of the software testing and prototyping process for Jessii. The joints of Jessii are detailed in a Universal Robotics Description Format (URDF) file, allowing for accurate modeling of the real-life dynamics of the system. Plugins to the simulator allow for accurate simulation of various sensors used, which enables testing of the entire software stack in simulation. The simulations consist of courses that increase in difficulty, starting from the qualification course and scaling up, allowing for a rigorous test of the behavior of the robot when new features are added to ensure that the code produces the expected behavior.

8 Initial Performance Assessments

Max Speed	4.89 mph
Acceleration	$5\,\mathrm{m\,s^{-2}}$
Ramp Climbing	25 degrees
Battery Life	6 hours standby 2 hours with motors running

9 References

[1] The Field D* Algorithm for Improved Path Planning and Replanning in Uniform and Non-Uniform Cost Environments

[2] Martin A. Fischler & Robert C. Bolles (June 1981). ”Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography” Comm. ACM. 24 (6): 381395. doi:10.1145/358669.358692.

[3] Ronneberger, Olaf; Fischer, Philipp; Brox, Thomas (2015). ”U-Net: Convolutional Networks for Biomedical Image Segmentation”. arXiv:1505.04597 [cs.CV]