Design Review 5

Formula 1/10 Autonomous Vehicle

Team 303 Members:

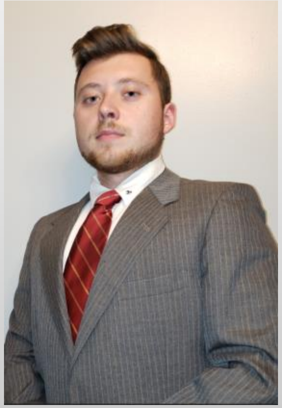
Cody Vanderpool, Michael Calisi,
Derek Swenson, Steven Roy, Nicholas Stiles

Department of Electrical and Computer Engineering

FAMU-FSU
College of Engineering

Meet The Team

Cody Vanderpool
PROJECT MANAGER



Steven Roy
MECHANICAL ENGINEER

Derek Swenson
SOFTWARE ENGINEER

Nicholas Stiles
SOFTWARE ENGINEER

Michael Calisi
ELECTRICAL ENGINEER

Project Overview

The Competition

- The Formula 1/10 competition gives students an opportunity to learn about perception, planning and control for autonomous vehicles [1].

Fig. 1: An autonomous 1/10th scale vehicle. [1]

- Teams from around the globe build vehicles and design algorithms for autonomy before racing one another.
- Robot Operating System (ROS) is commonly used to implement autonomous navigation.

Project Summary

- Design and build a 1/10th scale car that can analyze its surroundings and navigate around obstacles autonomously.

- Requirements:
  - Avoid walls and other obstacles without human input.
  - Make decisions in real time.
  - Operate at a safe and controlled speed.
  - Ability to switch between autonomy and remote control.
  - Adhere to the rules and guidelines of the F1/10 Autonomous Racing Competition Rulebook.
  - Theme based on Mel Brook’s 1987 cult classic *Spaceballs*.

Stakeholders

- Sponsor:
  - Dr. Jerris Hooker
- Advisor:
  - Dr. Shayne McConomy
- The Formula 1/10 Autonomous Racing Competition
- FAMU-FSU College of Engineering

Recap: Fall Semester

- ➢ Vehicle will run using the NVIDIA Jetson TX2 which will operate using Robot Operating System (ROS).
- ➢ Selected a combination of a LIDAR and a ZED stereoscopic camera to collect both a 360° 2D slice of wall distances and a 3D mesh of environment in front of vehicle.
- ➢ Selected all necessary parts and a configuration that abides by competition rules.
- ➢ Began learning ROS as well as testing on the ZED camera and collecting data.

Physical Build of the Vehicle

Vehicle Build

- Stripped down Traxxas 1/10th scale Ford Fiesta Rally Car chassis.
- Mounting brackets and platforms 3D printed in PLA.
- All components and sensors mounted with hex standoffs using pre-existing holes.

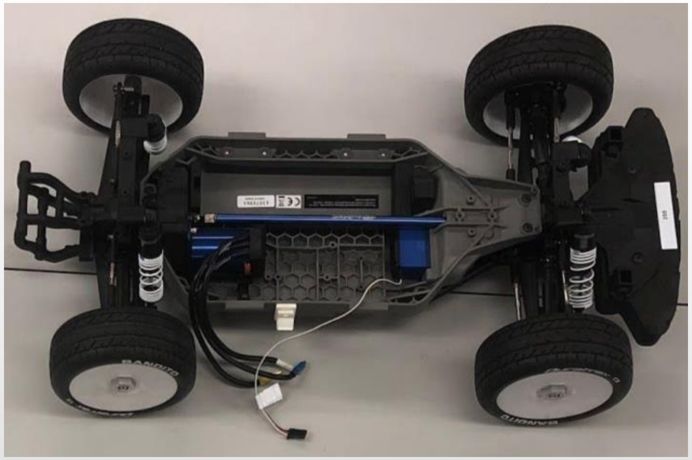


Fig. 2: Traxxas RC Rally Car chassis.

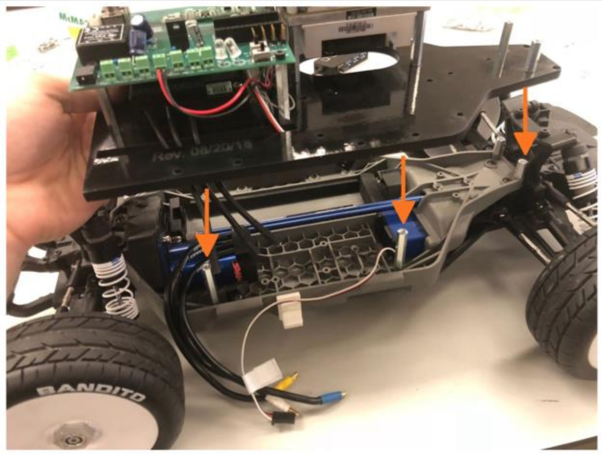


Fig. 3: Mounting of Jetson and Powerboard.

Powerboard Assembly

Board Specifications:

1. 110mm x 77.30mm
   1.6mm thick
1. Weight: 3.2 ounces
2. 48 total
   components
3. Input:11.2v
4. Output: 3v, 6v , 8v

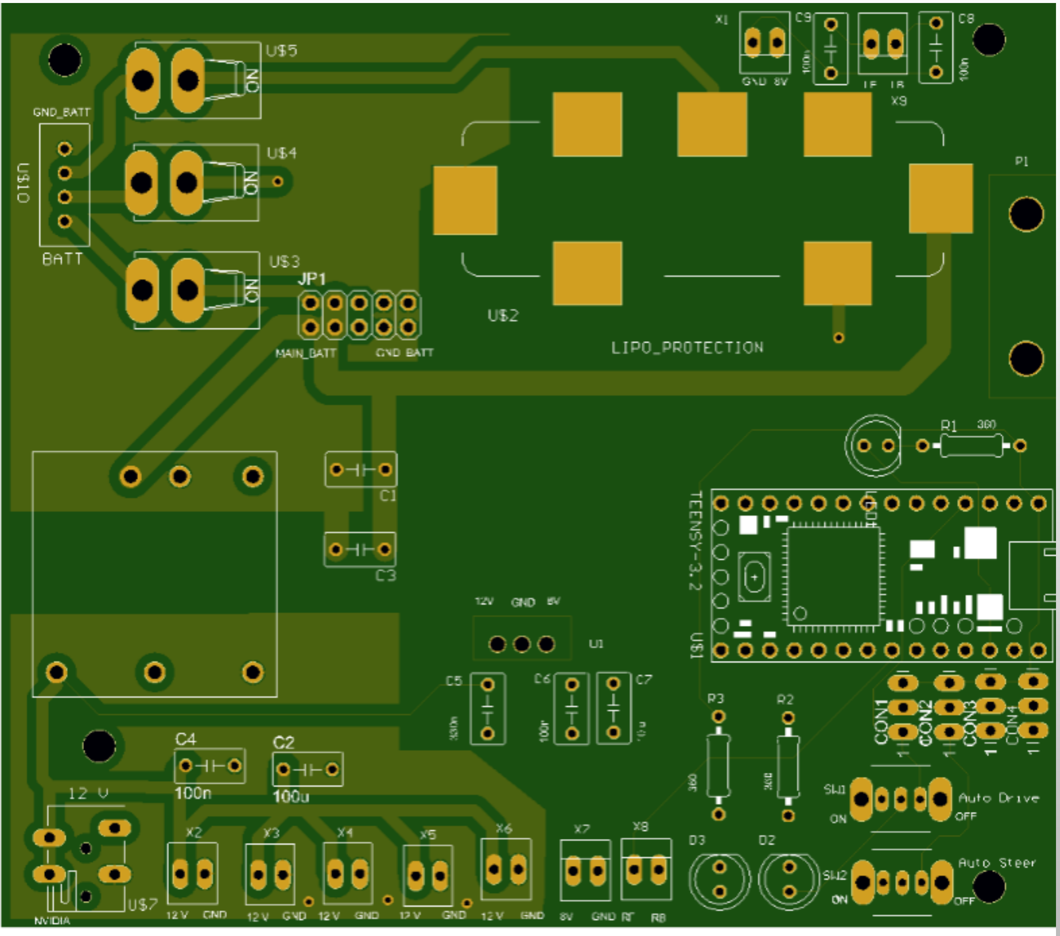


Fig. 4: Power board model

Main Components

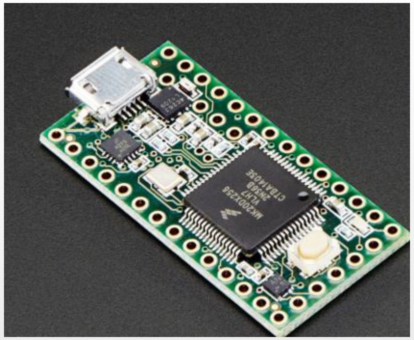


Fig. 6: Teensy 3.2


Fig. 7: 12V DC transformer

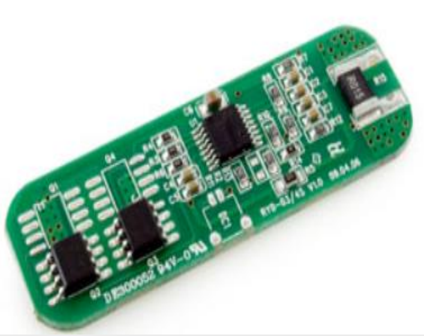


Fig. 8: LiPO protector

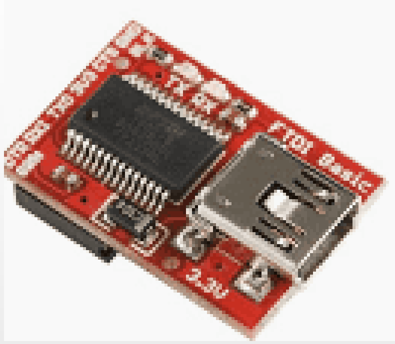


Fig. 9: Breakout Board

- Teensy 3.2-Shield: Jetson - Speed Controller - Servo communication.
- 12v DC.v.DC transformer: Steps down power from 11.2v to 8v for Teensy 3.2 and IMU.
- 3s LiPO protection circuit: Kills power before voltage depletion point and prohibits current fluctuations to processing components.
- Sparkfun Breakout board: converts the 6 pin header to usb for the communication between the Jetson tx2 and the IMU.
- HBR 11.2v LiPo battery: provides all power


Fig. 10: 11.2V LiPO battery

Powerboard Assembly

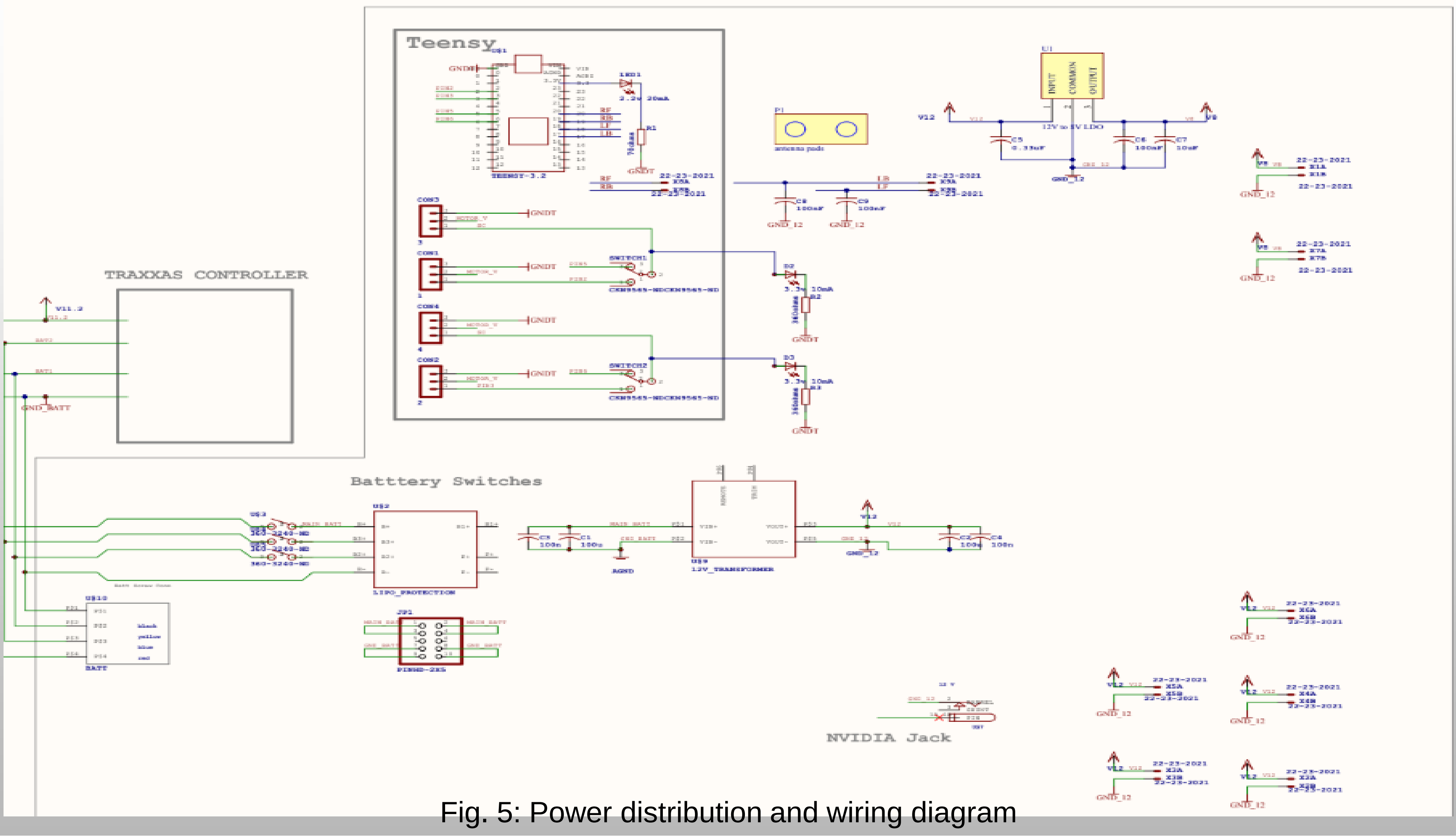


Fig. 5: Power distribution and wiring diagram

Controls Overview

Autonomous Control Considerations

- A human driver makes decisions based on sensed vehicle-road interactions such as:
    - velocity
    - acceleration
    - forces
    - noise
    - obstacles
    - position
    - vibrations
    - roll
    - slip

- Simplified Model Considers:
    - Distance and Direction to objects
    - Position, Velocity, and Acceleration

- Controllable Variables:
    - Steering angle (**δ**)
    - Motor speed (ω)

How Do We Maintain Trajectory?

- ➢ Algorithm uses LIDAR data to maintain equidistance from walls
- ➢ Camera will give distance to objects ahead
- ➢ Slow down when a turn is detected and accelerate after turn

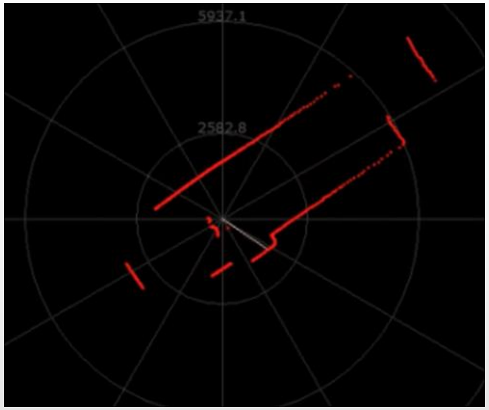


Fig. 11: LIDAR creates a 2D map of surrounding surfaces.

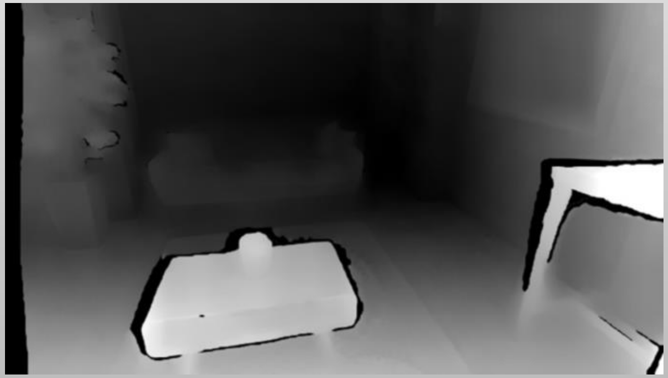


Fig. 12: ZED Stereoscopic camera maps distance in grayscale.

PID Controller Objectives

- ➢ Keep vehicle driving along centerline of track
- ➢ Manage steering attack response times
- ➢ Modulate speed to be fast on straightaways and slow down coming into turns

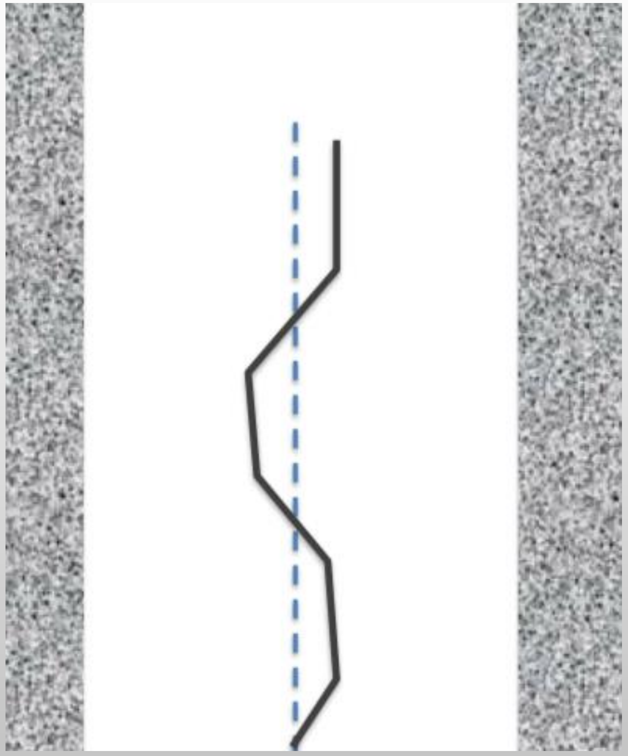


Fig. 13: The goal is to drive the error in lateral distance from the centerline to zero. [1]

PID Controller

- Two separate models
  - ‘Driver’ model controls error in trajectory
  - ‘Bicycle’ model simulates vehicle physics

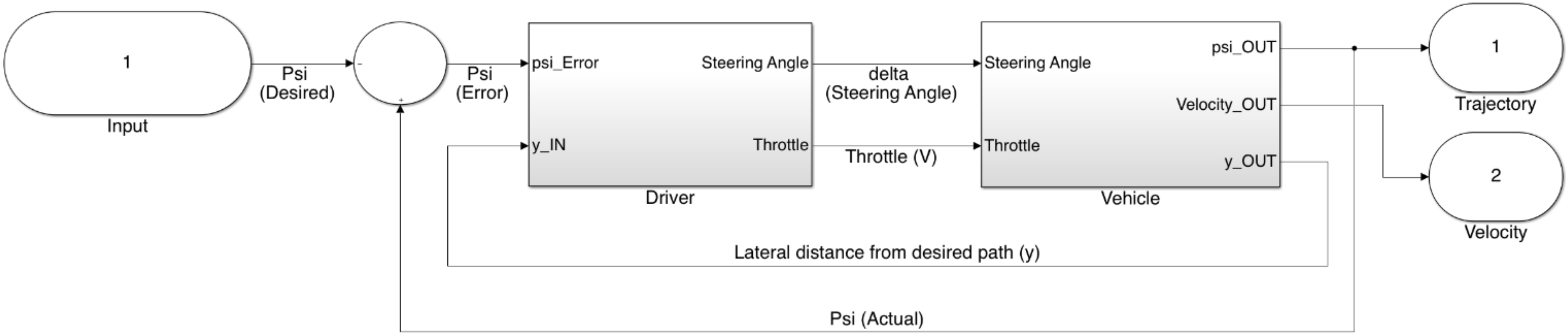


Fig. 14: Simulink model of PID controller.

Genta Driver Model

- A simple driver model that incorporates predictive behavior.
- Variables:

τ - Time delay
δ - Steering angle
$\dot{\delta}$ - Steering angle rate of change
Kd - Gain
ψ - Actual yaw angle
ψ_1 - Desired yaw angle
y - Lateral distance from desired trajectory
L - Preview distance

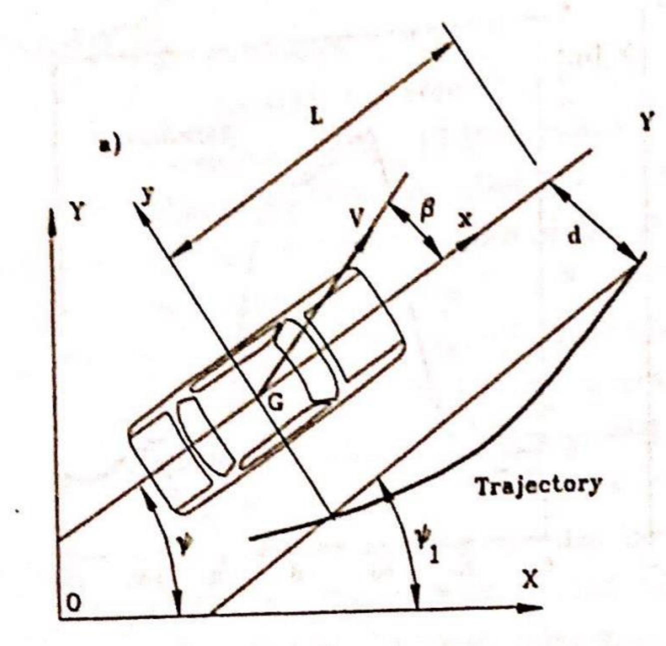


Fig. 15: Genta Driver Model. [2]

$$\tau\dot{\delta}(t) + \delta(t) = -K_d[\varphi(t) - \varphi_1(t) + \frac{y(t)}{L}]$$

Bicycle Vehicle Model

- ➢ Simplifies vehicle kinematics to one wheel per axle
- ➢ Comprised of linearized equations of motion
  - ∑F_y , ∑F_x , ∑M_z
- ➢ Transform into two unknowns:
  - Yaw rate
  - Sideslip angle (lateral vel.)
- ➢ Variables of note:

v - velocity
β - sideslip angle
δ - steering angle
ψ - yaw angle

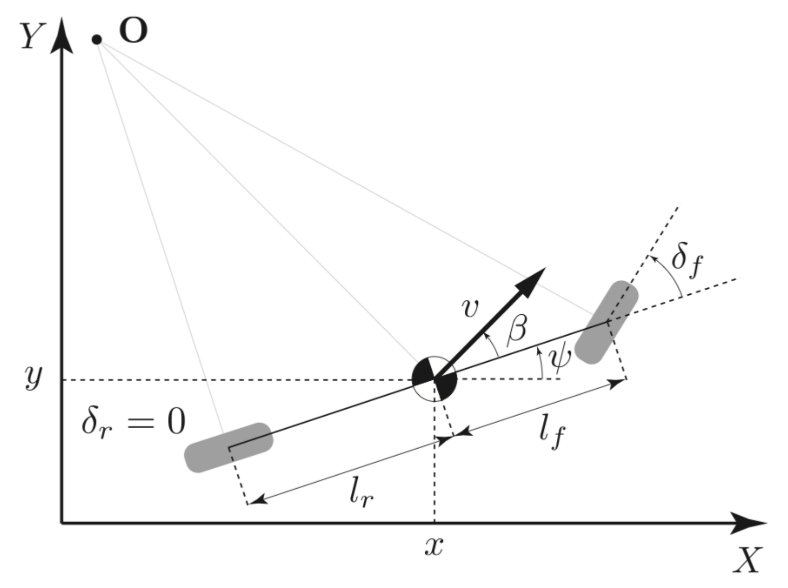


Fig. 16: Bicycle Vehicle Model. [3]

Data and Mapping Overview

Data Collection

- ➢ ZED cameras point cloud data will be used in order to create an accurate depth map of the environment
- ➢ IMU will control localization while the RPLIDAR will keep track of distances out of ZED camera frame
- ➢ Will use the rviz ROS library in order to visualize created map

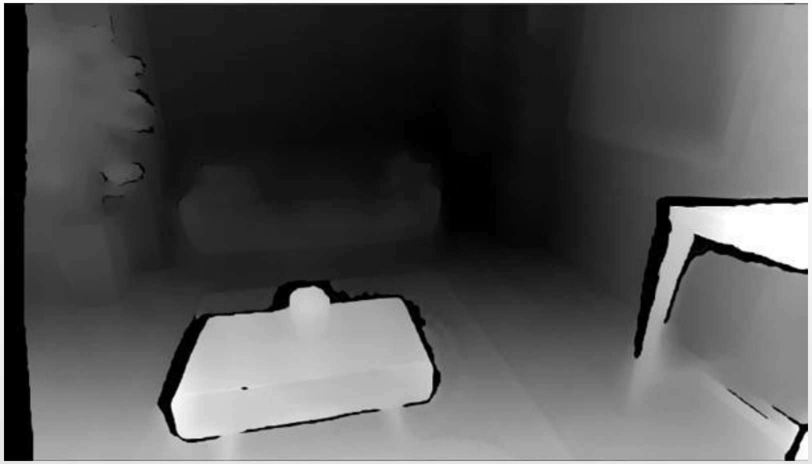


Fig. 17: ZED point cloud data.

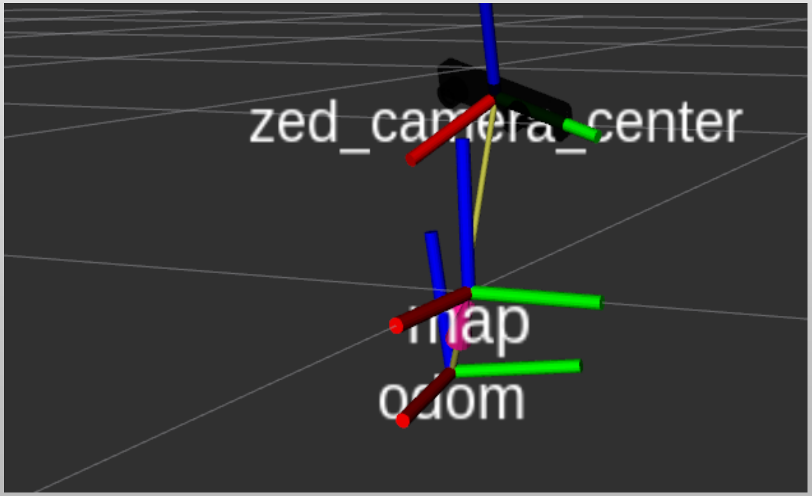


Fig. 18: Rviz model of camera and IMU.

Localization

- Mapping of the environment will be generated using the Hector SLAM library, which utilizes ROS bag files
- Since Hector SLAM is computer intensive, real time localization will use MIT Particle Filter Localization
- Updates at a rate of 30Hz and utilizes GPU, whereas other options only use CPU

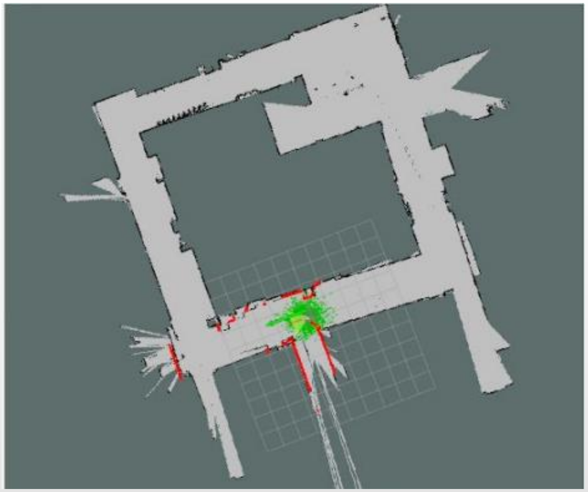


Fig. 19: Hector SLAM generated map.

Fig. 20: Particle Filter Localization.

Simulation

- ➢ Test algorithms in a controlled environment before we bring it into the real world so that we minimize risk of crashing.
- ➢ ROS ‘Gazebo’ simulator software
- ➢ Loads a world as a .DAE file and loads the car. It has a physics engine that can determine when the car crashes into a wall.

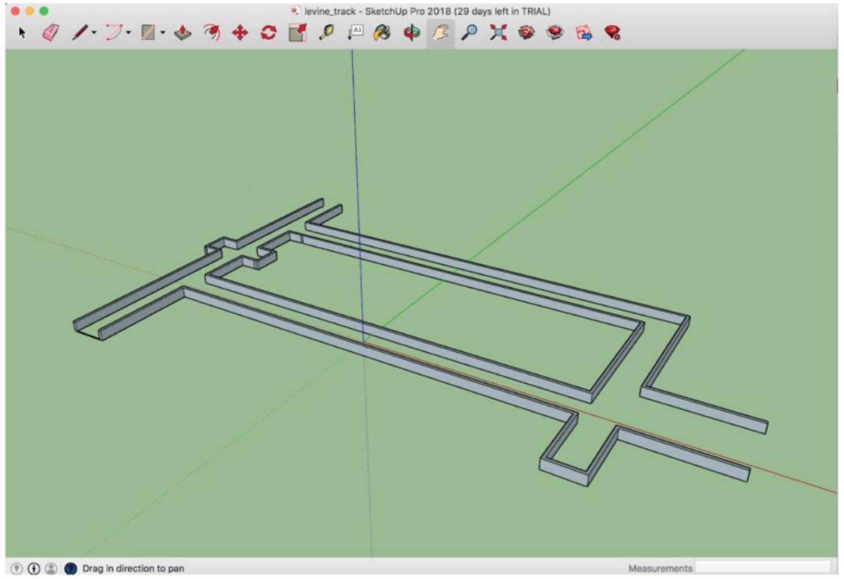
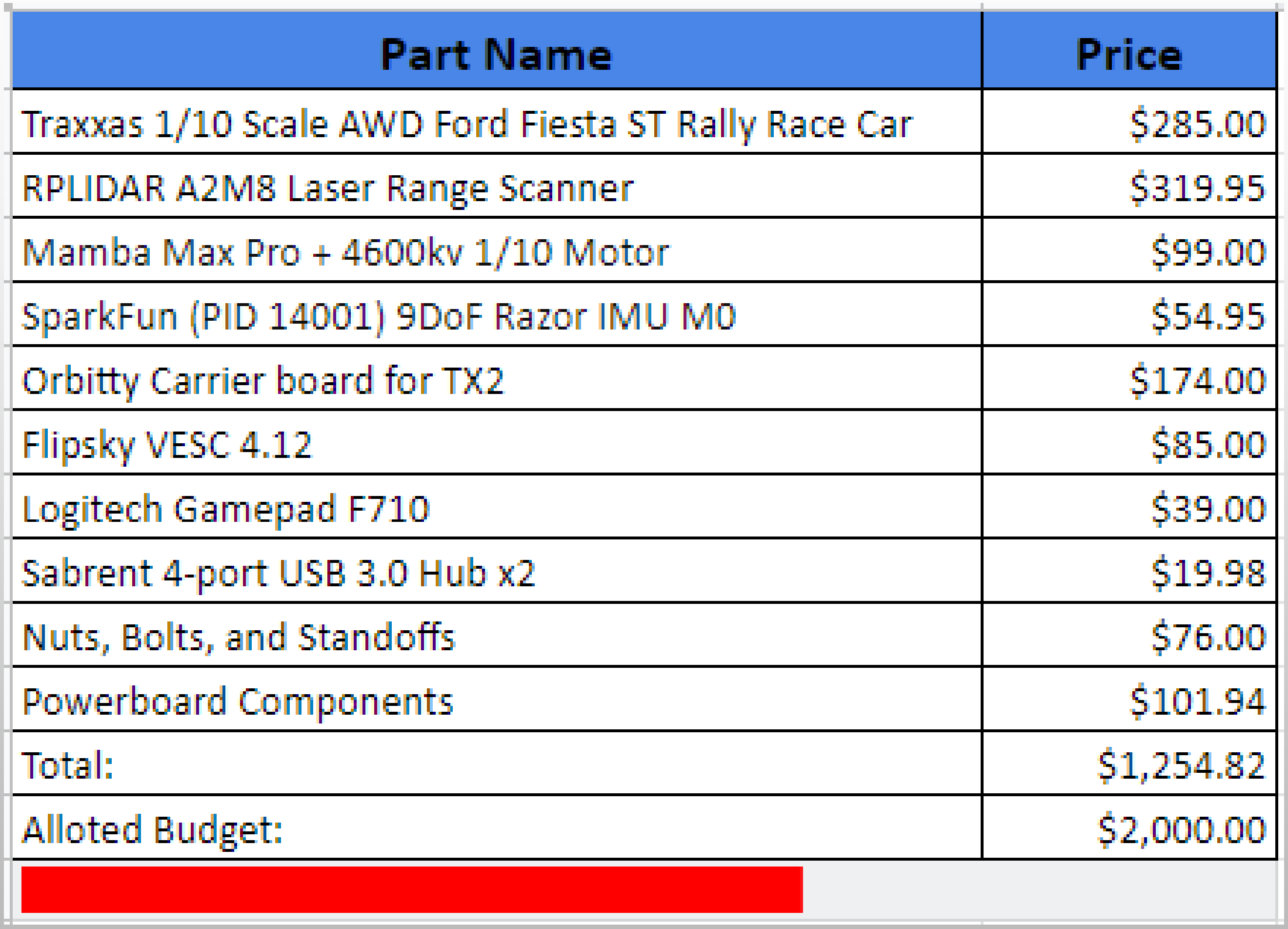


Fig. 21: Gazebo 3D model of hallway.

Looking Forward

Budget Update



Part Name	Price
Traxxas 1/10 Scale AWD Ford Fiesta ST Rally Race Car	$285.00
RPLIDAR A2M8 Laser Range Scanner	$319.95
Mamba Max Pro + 4600kv 1/10 Motor	$99.00
SparkFun (PID 14001) 9DoF Razor IMU M0	$54.95
Orbitty Carrier board for TX2	$174.00
Flipsky VESC 4.12	$85.00
Logitech Gamepad F710	$39.00
Sabrent 4-port USB 3.0 Hub x2	$19.98
Nuts, Bolts, and Standoffs	$76.00
Powerboard Components	$101.94
Total:	$1,254.82
Alloted Budget:	$2,000.00
[redacted]	

Progress Update

- Vehicle Assembly: 80% Complete
- Main Concerns:
  - Construction, Implementation and Fine Tuning of the Driver and Vehicle models is a difficult task.
  - Translating data from sensors to necessary variable inputs for control system.
  - Implementation of travel between offices will leave little time for testing

Next Steps

- Mount powerboard and connect all components
- Calibrate VESC PID gains
- Map gamepad controls to motor outputs
- Integrate OpenCV in order to path-find
- Run simulations using Gazebo collecting data in order to determine best algorithm
- Begin testing in real environment
- Implement speed controller once path-finder has been refined

References

[1] ‘About F1/10’. [Online]. Available: http://f1tenth.org/about [Accessed: 12- December- 2018].

[2] G. Genta, Motor Vehicle Dynamics: Modeling and Simulation. Danvers, MA: World Scientific Publishing Co. Pte. Ltd., 1997, p. 205-324.

[3] J. Kong, M. Pfieffer, G. Schildbach, F. Borrelli, ‘Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design’. [Online]. Available: https://borrelli.me.berkeley.edu/pdfpub/IV_KinematicMPC_jason.pdf [Accessed: 10- February- 2019].

Questions?