Ptero

Pterra

Team

Peter: Software

Swetha: Jetson, LiDAR, mapping

Cher: Drone/rover, Custom Hardware

Tiffany: LiDAR, mapping, logo

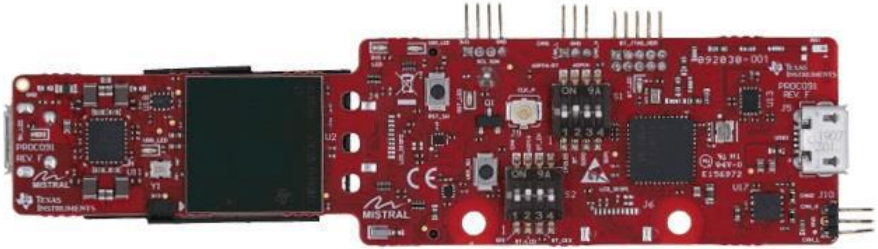
Jackie: Controller interface

Problem

Traditional obstacle avoidance is expensive with constrained solutions

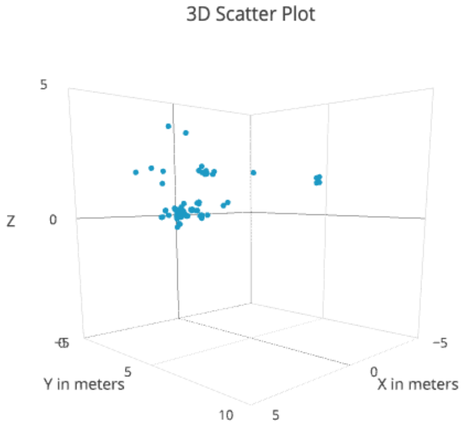
- LiDAR is generally expensive and usually moves
- Time-of-Flight sensors are very costly, and have a small field of view
- Multiple cameras provide good depth performance, but fail with poor light, dust, or smoke

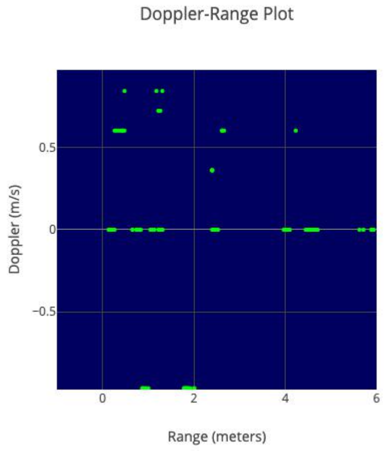
Can millimeter-wave RADAR fill in this gap?



mmWave RADAR

- Operates between 60- and 77-GHz
- 130 degree azimuth / elevation
- Returns 3D point clouds with doppler data
- Can see through smoke, dust, etc





mmWave RADAR

- Can see through:
  - Fabric
  - Drywall
  - Unmoving objects
  - And other objects…
- Data is incredibly noisy
- Sensor is very sensitive to parameter shifts

Goals

1. Demonstrate obstacle detection with mmWave RADAR sensors
2. Develop safe control algorithms which will remain safe with an uncertain connection to the vehicle
3. Demonstrate how RADAR can be used seamlessly with other sensors to improve safety

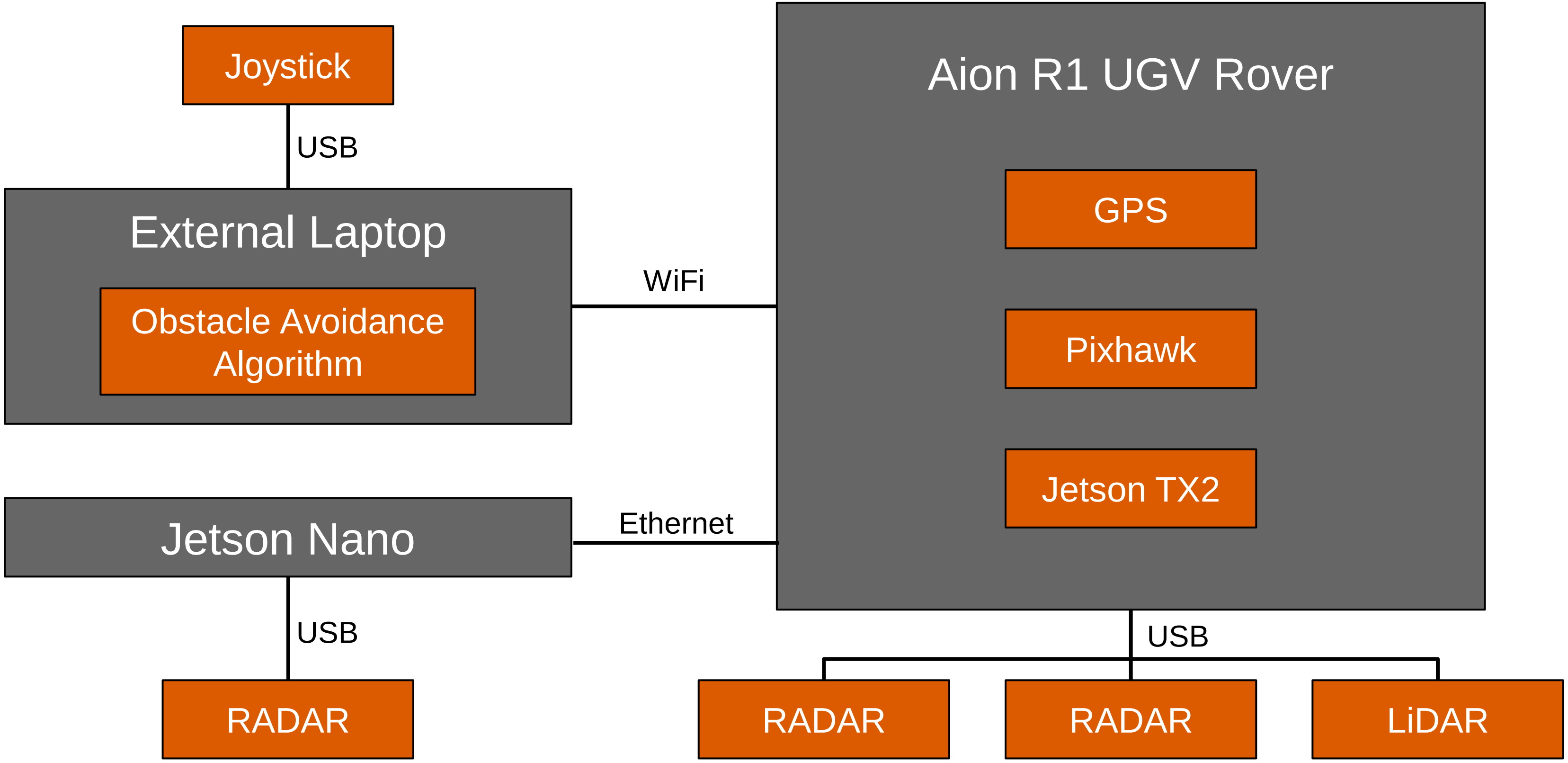
Constraints

- Sensor placements
  - We want as large a field of view as possible, so we decided to use 3 RADAR sensors to cover 86% of the system.
- Latency
  - Due to our goal of working through latency, we decided to allow minimal latency to enter the system.
  - Use WiFi for control
  - Operate the RADAR sensors at 10Hz.
- Time

Final Product

WaveSafe with Mounted Sensors

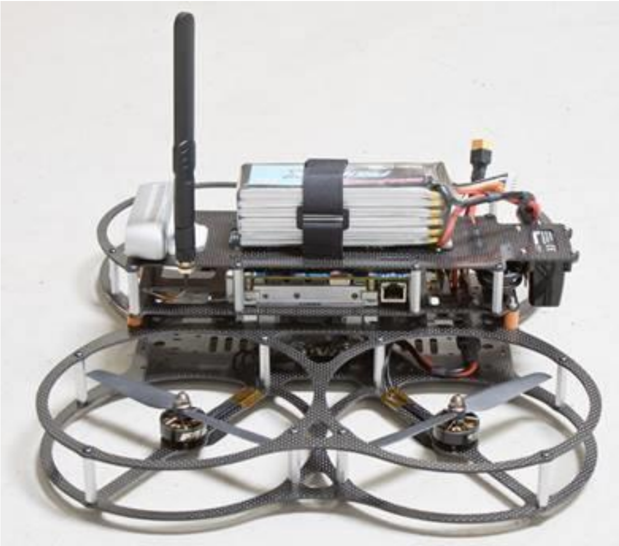
Block Diagram



Hardware

Quadcopter

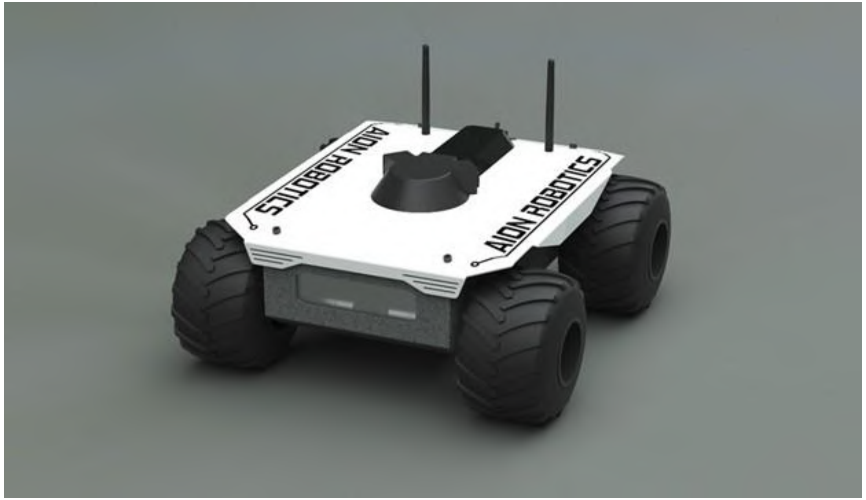
- Motor placement backwards
- Drifting
- Difficulty landing
- Inconsistent flight
- Lost hardware



**our algorithm will work with a functional drone

Aion Robotics Rover

- Robot Operating System (ROS)
- Nvidia Jetson TX2
- Pixhawk 2.1
- GPS
- Tablet controller



NVIDIA Jetson

- Jetson TX2 (on rover)
  - NVIDIA Pascal architecture
- Jetson Nano for testing
  - Low cost
  - Sensor compatibility

**Both include Jetpack SDK, which includes board support package (BSP), Linux OS, NVIDIA CUDA, cuDNN, and TensorRT libraries

TI mmWave RADAR

3x TI RADAR sensors connected over USB with custom mounting hardware

Slamtec RPLiDAR A1

- 360 degree
- 12-meter range
- 2-10 Hz sample rate
- 0.2 cm, 1 degree resolution

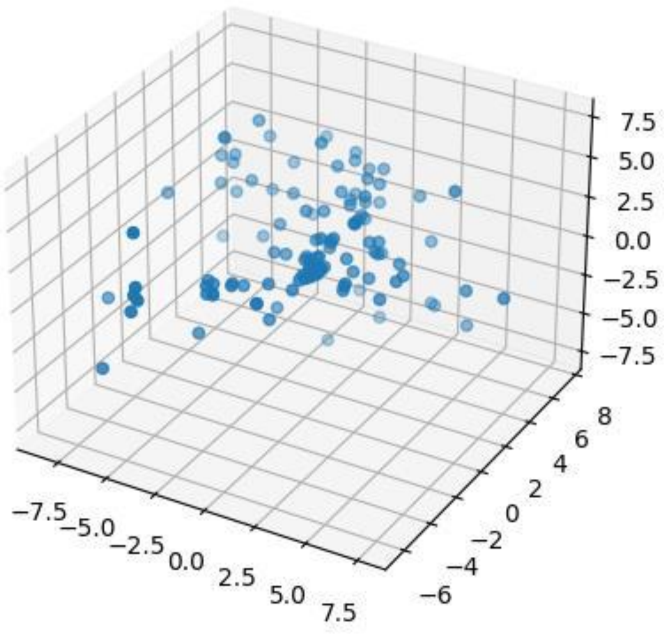
Software

RADAR

Fully asynchronous sensor management

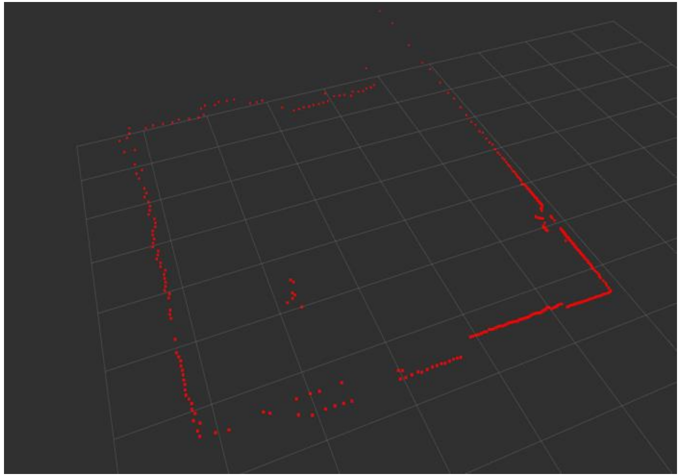
Sensor data is merged on-vehicle

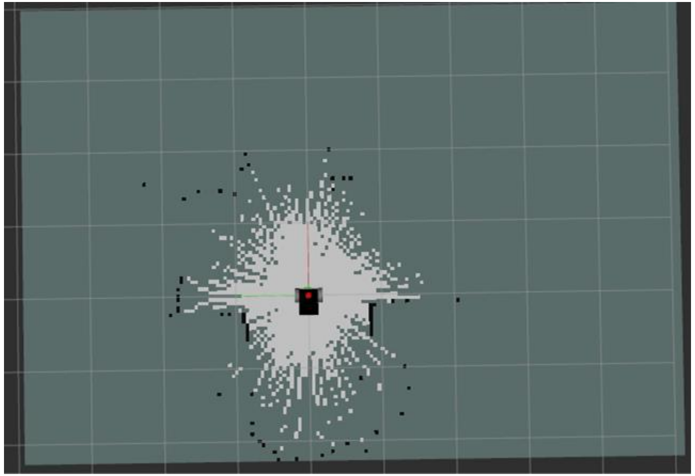
Data is filtered to remove noise before usage



LiDAR

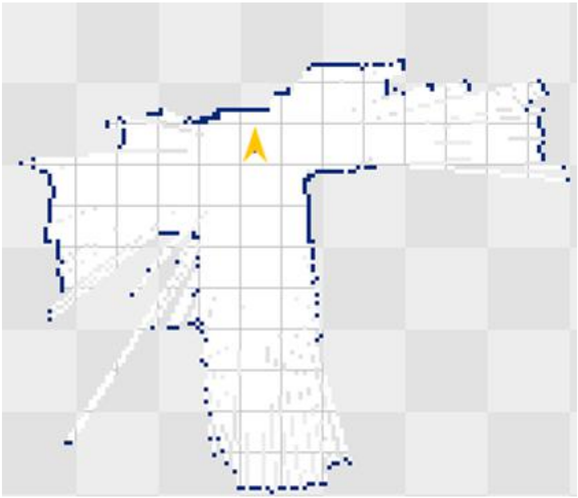
Hector SLAM creates a continuous map while vehicle is in motion using scan matching and pose estimates





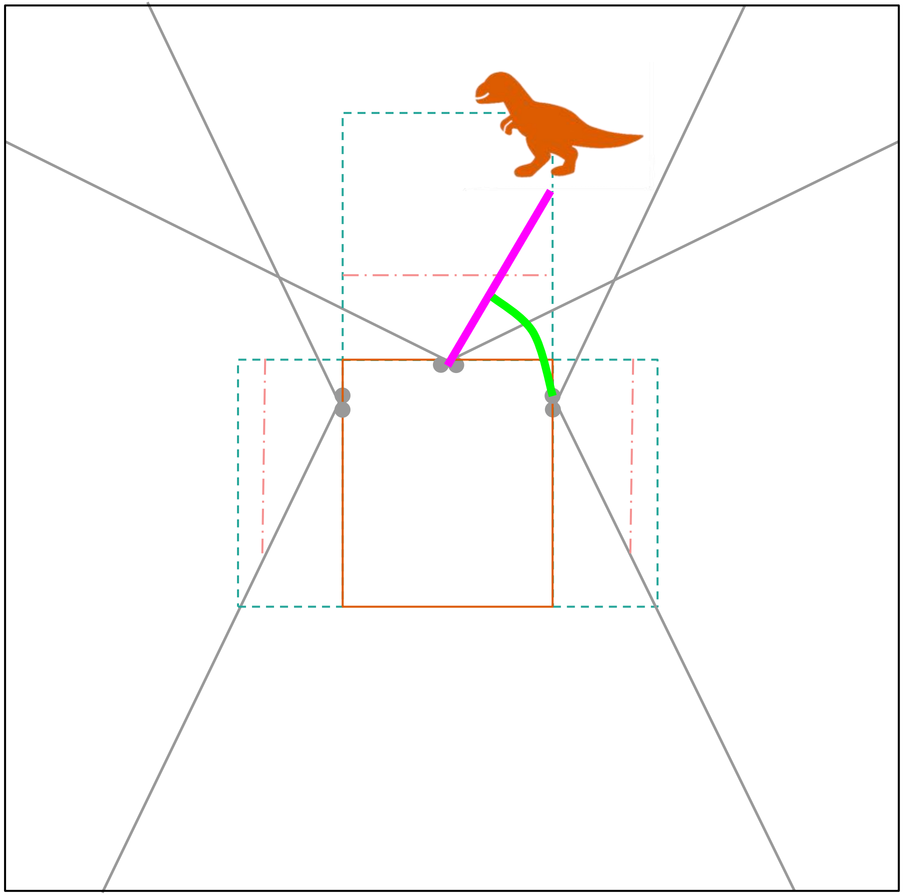
LiDAR with RADAR

- Interfaced LiDAR with ROS node and used data in addition to RADAR data
- LiDAR allows us to see behind the vehicle
- Able to compare performance



Obstacle Avoidance

- Based on the closest obstacle, we compute an angle tangent to it as a goal
- We iteratively modify the users input to change the angle of approach
- Slow down velocity based on distance



`pip install pymmWave`

- Published an open-source Python package for wide-usage
- Published optimized sensor interaction tools
- Provided a set of complex custom RADAR algorithms for pose estimation, continuous state estimation, and estimating movement with a single RADAR sensor.

Acknowledgements

Yogananda Isukapalli - Capstone Instructor

Boning Dong - TA

Trenton Rochelle - TA

Phil Tokumaru - Project Advisor

Scott Rasmussen - Drone Specialist

AeroVironment™

UC SANTA BARBARA
College of Engineering

Q&A