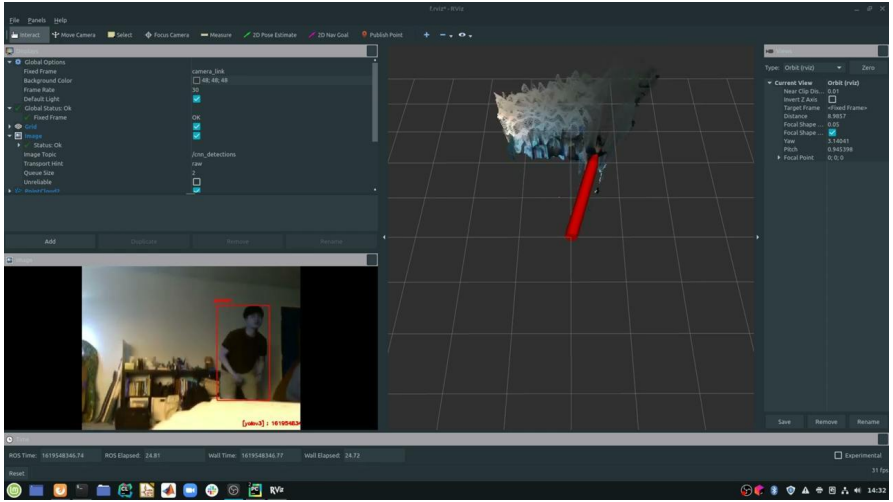
Team C9: GrubTub

Final Presentation

Advaith Sethuraman, Michael Li, **Sebastian Montiel**

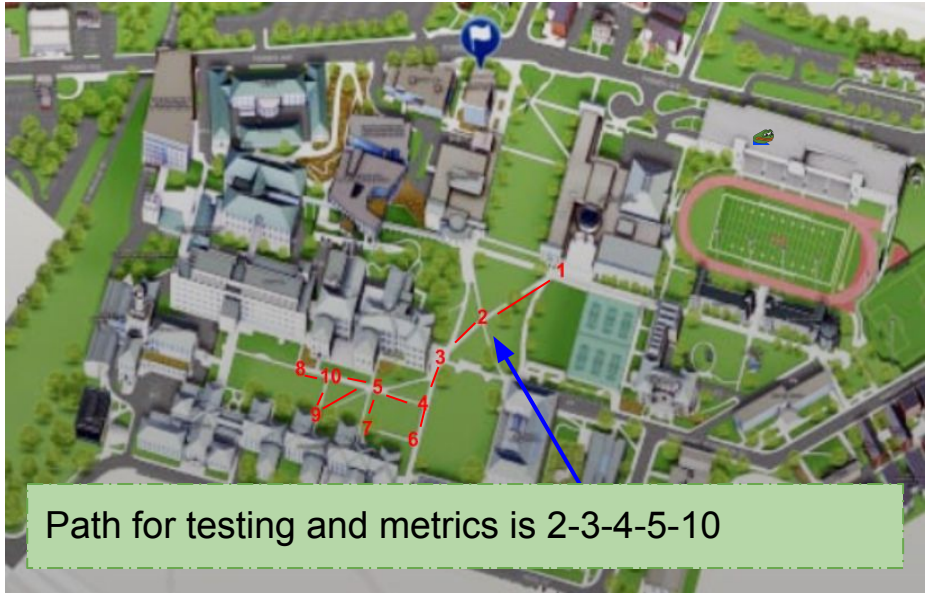
5/3/2021

Highlight Reel

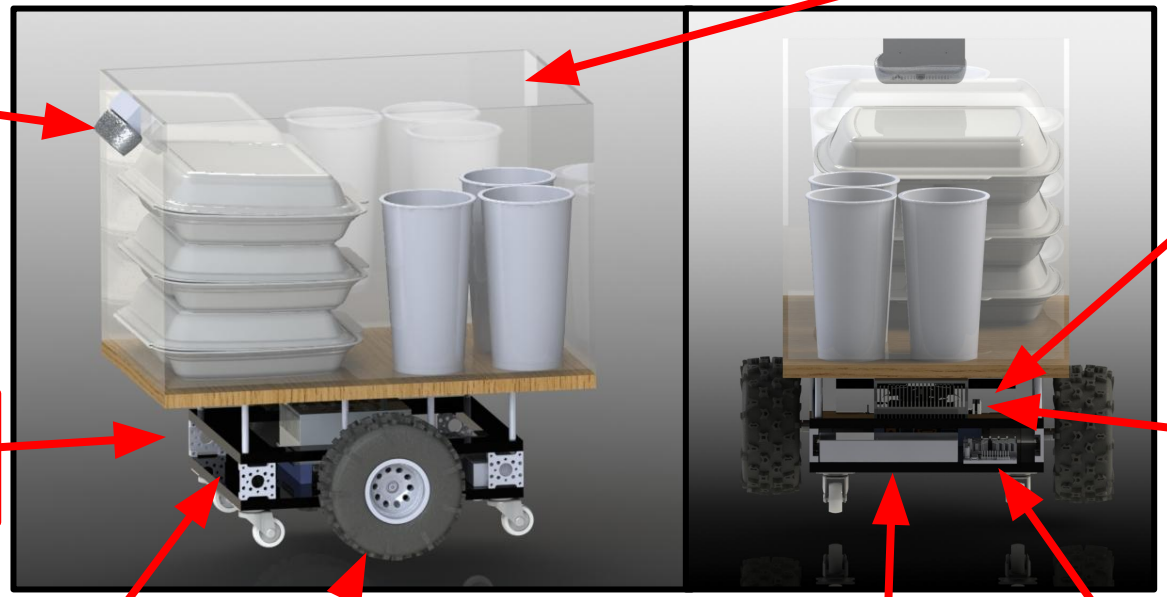


Application Area

- On-campus food delivery
- Delivers food within campus roads
  - **No slopes**
- Must hold and transport multiple deliveries
- Must not collide with pedestrians
- Minimal human intervention
- Food delivery in a timely fashion
- Food must be intact during delivery



Mechanical Solution: Expectation



16” x 11" x 10" (1.02 ft^3) Payload

Intel Realsense D415

Adafruit IMU

Prowler chassis

Nvidia AGX Xavier

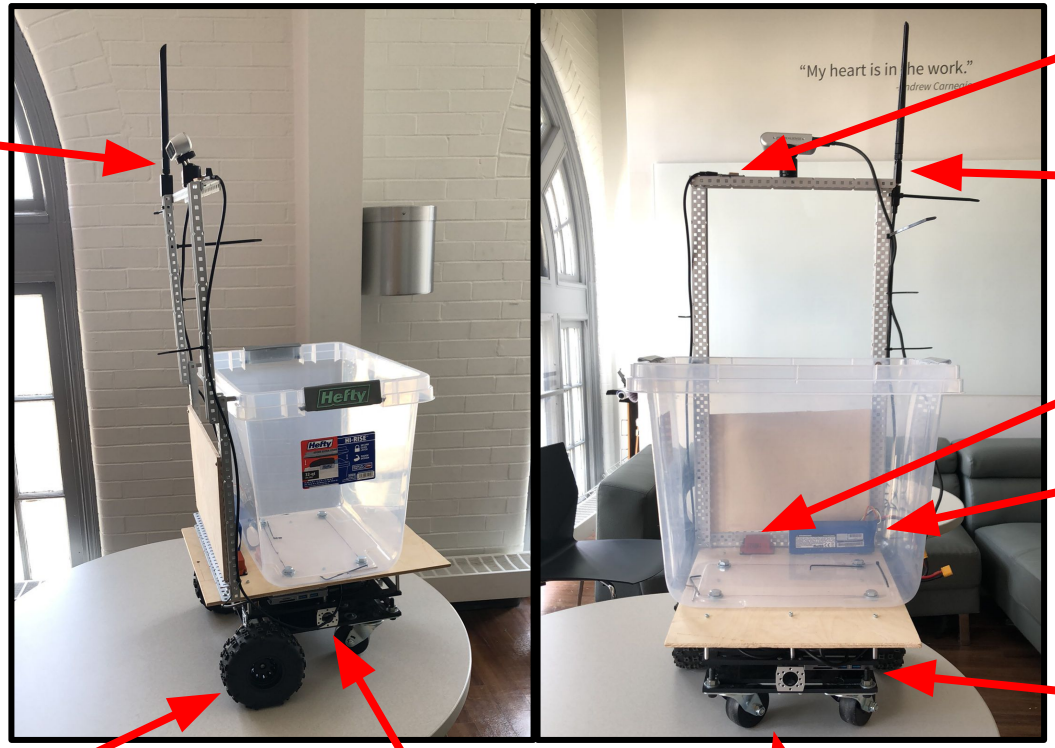
Electronics Bay

5” Rubber wheels

2*5000mAh LiPo Battery

Roboclaw 2x15A Controller

Mechanical Solution: Reality



Intel RealSense D435

Adafruit Ultimate GPS

WiFi antenna

Xsens Mti-200 IMU

2*5000mAh LiPo Battery

Electronics Bay (Xavier, RoboClaw, USB hubs)

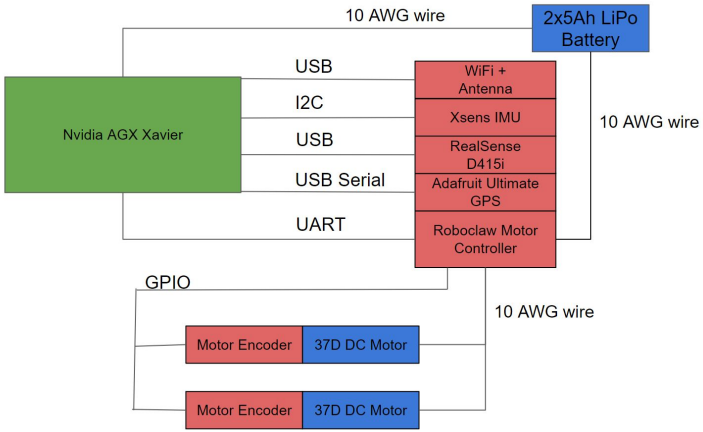
5" Rubber wheels

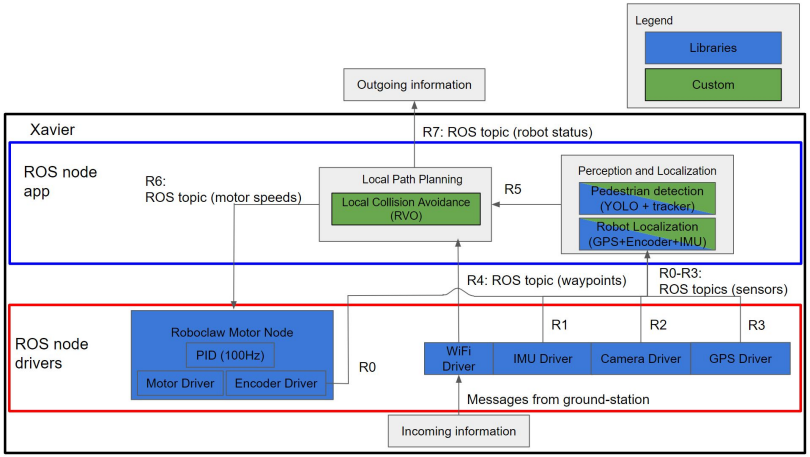
Prowler chassis

3.5" Caster wheels

Hardware and Software Design Changes

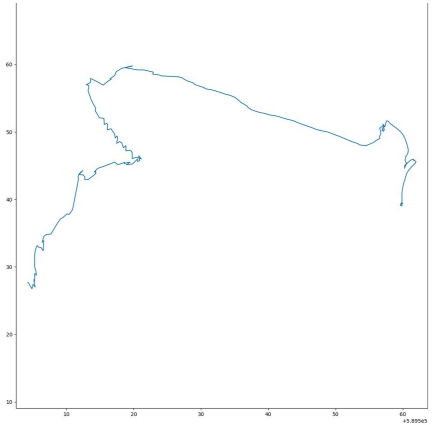
- Adafruit BNO055 IMU → Xsens Mti-200
- RTABMAP visual SLAM → non-visual localization with a hardcoded map



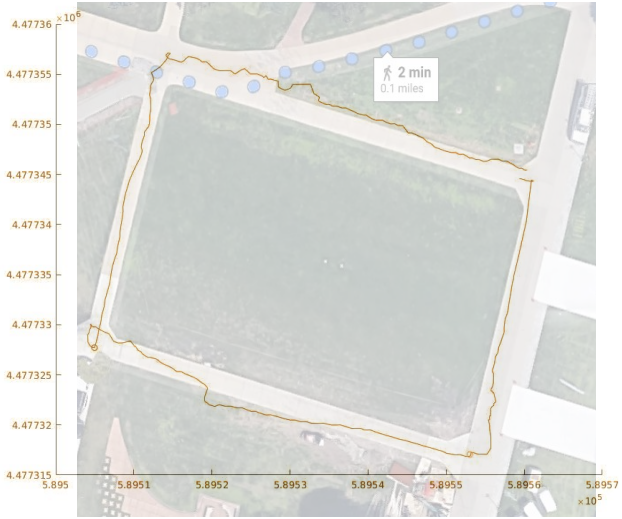


Localization Challenges

Raw GPS Data



Filtered Localization Data



Human Operated Localization Data

- GPS noise under trees and large buildings (and in general)
  - GPS typically reports it has < 1 HDOP (Horizontal Dilution of Precision)
- Wheel slip when it goes into the grass → encoders and IMU disagree
- Tuning (for the attempt at P/PID controls)

Other Challenges/Tradeoffs

- Inconsistent WiFi on campus
- Libraries with bugs (RoboClaw)
- The Dutch Hackers
- Poorly tuned controls
- Visual SLAM vs. GPS/IMU/Encoder
- RVO controller vs. PID vs. LQR

I now turn my attention to finding the root cause of random sporadic failures when the **Roboclaw ROS driver** makes calls into Roboclaw API. The most

> https://www.cmu.edu/iso/governance/procedures/compromised-computer.html
>
> ---
> We have detected your computer checking into a known Command and Control
> server for crypto mining.
>
> See the sample traffic below. This computer's network connection has been

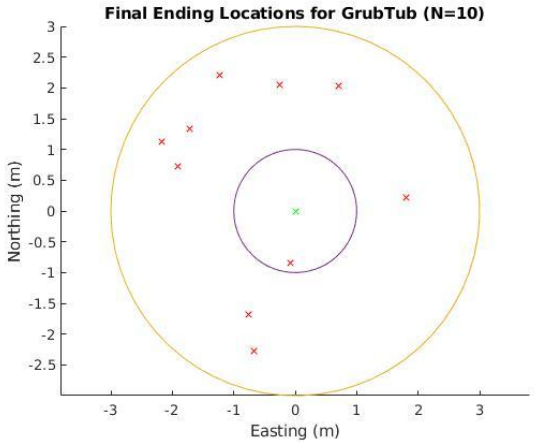
Testing, Verification, and Metrics

Requirement	Testing Method	Metrics	Result
Robot must deliver food on campus roads	Ran 10 trials on test path and logged time on the grass	> 65% of time (s) on sidewalk than the grass	With intervention, stays on sidewalk ~75% of the time
At least 30 minutes of battery life	Sum total time from sensor bags running on one charge	Battery lasts > 30 mins	Ran for 39 minutes
Robot must arrive close to the drop-off point	Measure FDE from known UTM coordinate	Within 1m 68% of the time Within 3m 100% of the time	3m success - 100% 1m success - 10%
Robot must transport multiple deliveries at once	We put objects in the tub with known weights	Must hold and transport 2kg food at 1 ft^3	Robot could carry 2.1kg
Food must be intact during delivery	Inspection of food after deliveries	Food must be intact 100% of the time.	Robot always arrived with the food undamaged.

Testing, Verification, and Metrics

Requirement	Testing Method	Metrics	Result
Robot must not collide with pedestrians or other objects	Have pedestrians walk near the robot while it is in transit.	Minimum distance between robot base_link and any human >= .25m	Our one live trial got within 5cm of Michael
Minimal human intervention	Ran trials and counted number of interventions	Interventions/distance <= 1 intervention/50 meters	2-3 Interventions @ ~160 meter path (2/160-3/160)
Robot must deliver food in a timely fashion	Ran trials and timed the path	Each delivery time must be less than HRTT (human round trip time)	The path's HRTT is ~4 minutes. It takes the robot ~6 minutes
Robot must connect to ground station for intervention/tracking	Attempting to use the robot in various parts of campus	Tail latency < 750ms, need time for emergency control takeover	Average latency < 500ms, but inconsistent WiFi leads to an infinite latency tail

Quantitative Test Results



Trial #	Grass time (s)	Grass Percent	1M accuracy (m)	3m accuracy (m)	Food Intactness (Bool)	Interventions/Meter	Total Interventions	HRTT Time (s)	Robot Time (s)	% Over HRTT	Path Length (m)
1	121	30.63%	0.85	0.85	Intact	0.0125	2	240	395	64.58%	160
2	125	32.05%	2.17	2.17	Intact	0.01875	3	240	390	62.50%	160
3	102	27.79%	1.85	1.85	Intact	0.0125	2	240	367	52.92%	160
4	149	35.56%	2.37	2.37	Intact	0.025	4	240	419	74.58%	160
5	91	21.62%	2.04	2.04	Intact	0.0125	2	240	421	75.42%	160
6	142	36.04%	1.82	1.82	Intact	0.0125	2	240	394	64.17%	160
7	97	24.62%	2.53	2.53	Intact	0.01875	3	240	394	64.17%	160
8	104	27.01%	2.45	2.45	Intact	0.01875	3	240	385	60.42%	160
9	153	36.52%	2.06	2.06	Intact	0.00625	1	240	419	74.58%	160
10	135	34.88%	2.15	2.15	Intact	0.0125	2	240	387	61.25%	160
Average	121.9	30.67%	2.03	2.03	Intact	0.015	2.4	240	397.1	65.46%	160
Success Threshold	N/A	65%	1	3	Intact	0.02	N/A	N/A	240	0%	N/A
Metric Success Rate	N/A	100.00%	10.00%	100.00%	100.00%	90.00%	N/A	N/A	0.00%	0.00%	N/A

The End is Near!

5/4

Get Controls Running Properly on Odom

Keep tuning the PID and try LQR controls.

5/5

Attempt Running Controller using GPS

We attempt to run our controller and drive the robot in the noisier UTM frame.

5/6

Integrate High Level Software

Easy to do, but requires a finalized controller first.

5/7

Integrate Pedestrian Avoidance

Combine perception and avoidance policy and test