Team C9: GrubTub

Michael Li, Sebastian Montiel, Advaith Sethuraman

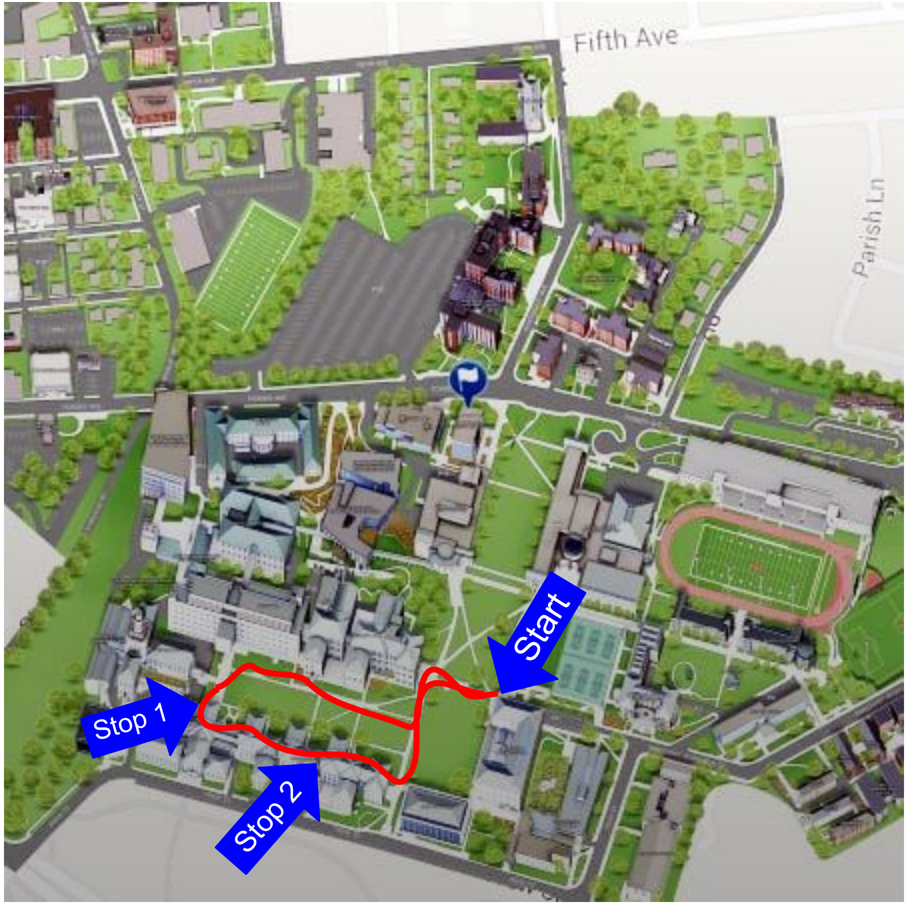
2/22/2021

Motivation

- Students get hungry, but don't want to go outside
- Intra-campus delivery is very convenient (don't need to walk to open road)
- Current approaches
    - Nuro: Autonomous, but meant for roads with limited human/sidewalk interaction
    - Starship: Fully autonomous, but not good at pedestrian avoidance
    - Kiwi: Semi-autonomous, with a person setting waypoints constantly (.2 Hz)

Use Case

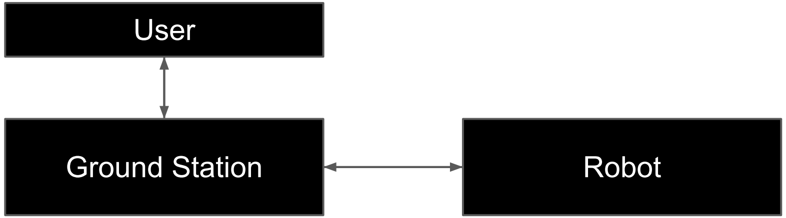
- **Customer:** Hungry College Students
- **Problem:** On Campus Food Delivery
  - Small, controlled version of generic food delivery
- **Solution:** Autonomous wheeled robot
  - Can schedule and carry out multiple placed orders
  - Travels on sidewalks and avoids obstacles/pedestrians
- **ECE Areas:**
  - Signals, Software, Hardware

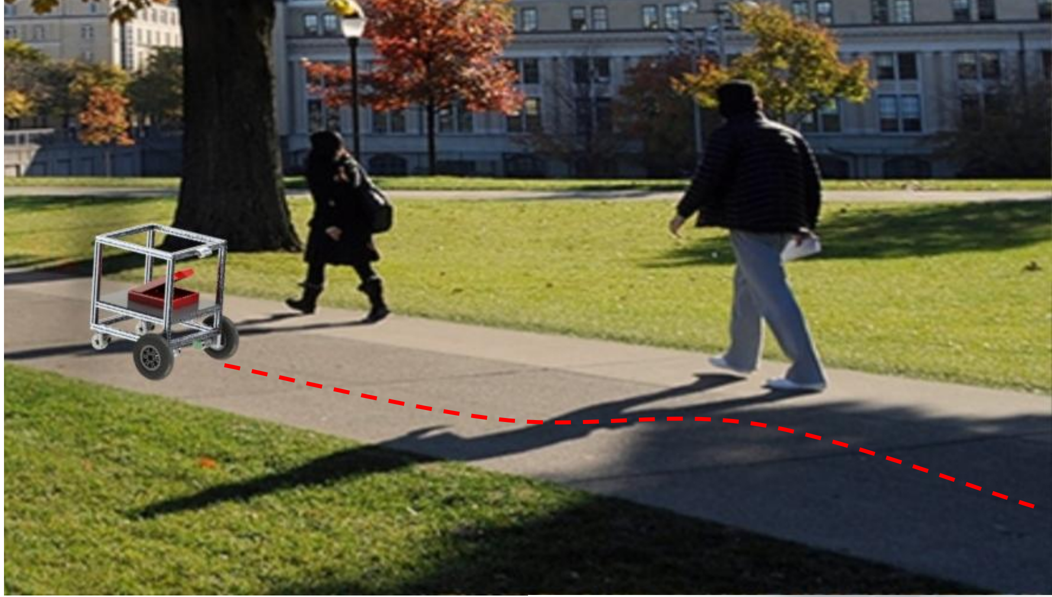


Requirements

- Delivers food within campus roads
  - Able to handle slopes and bumpy terrain
  - > 80% of time on sidewalk rather than the grass
- At least 30 minutes of battery life
  - Around 6 deliveries' worth (CFA to HH is 5 min)
- Must arrive within 1m at least 68% of the time (Gaussian with μ=1m), within 3m 100% of the time (customer comfort)
- Must hold and transport multiple deliveries
  - 2kg >>> 3*average meal weight, 1 ft^3 can fit 3 takeout containers
- Must not collide with pedestrians or other objects
  - Stay at least .25 m away from all other objects at any given point in time
- Minimal human intervention
  - Less than 1 human intervention per 50 meters
  - Metric based on average trip distance/num of human interventions
  - Ensures reliable autonomy
- Food delivery in a timely fashion
  - less than HRTT (human round trip time) measured in seconds
  - Handle multiple orders/deliveries
- Food must be intact during delivery
- Robot must connect to ground station for emergency interventions and tracking

Solution Overview

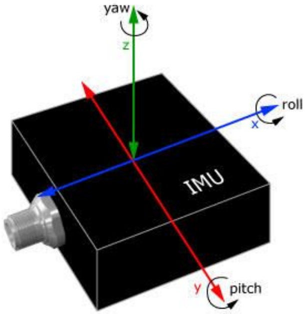


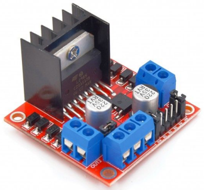


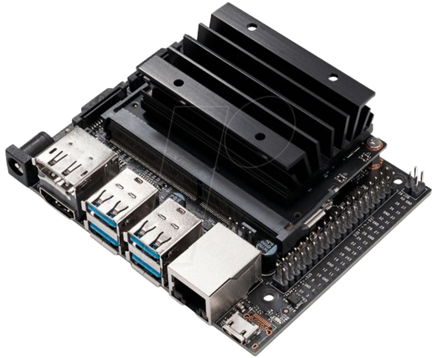
- **Mechanical:**
  - A robust, box-shaped robot chassis with 2-wheel differential drive
- **Hardware:**
  - Jetson Nano, Motors/Encoders, GPS, IMU, Lidar OR Camera, Wi-Fi Module
- **Software:**
  - Path Planning, Localization, Collision Avoidance, Controls, Networking

Solution: Hardware

- Jetson Nano (possibly TX2)
- Motors/Encoders
- L298N Motor Driver
- Wi-Fi Module
- Sensors
    - GPS
    - IMU
    - Motor Encoders
    - Lidar OR Stereo Camera

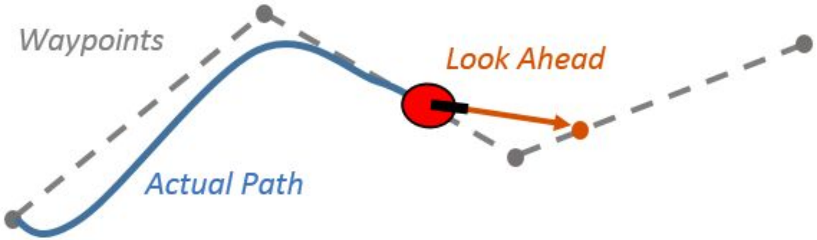


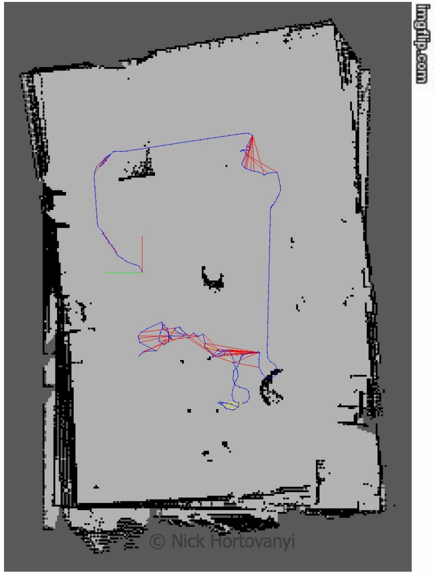




Solution: Software

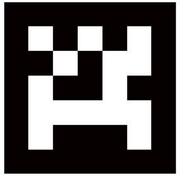
- Using ROS as framework
- **Obstacle Avoidance**
  - Use LIDAR data to detect objects nearby
  - Avoidance using custom policies (Reciprocal Velocity Obstacles, Reinforcement Learning)
- **Planning**
  - Multi-order scheduling (per our requirements)
  - Novel optimization algorithm to find delivery order to meet time requirements
    - Multi-order scheduling on a known graph is likely NP-hard
- Localization
  - RTABMAP using 2D Lidar (best case)
  - ORB-SLAM/VO using RGBD camera (middle case)
  - Fiducial Markers + Particle Filtering (worst case)
- Controls
  - Low level PID control loop for driving motor
  - Pure Pursuit for following paths
- Communication
  - ROS uses an abstraction on top of IP, running in a local network



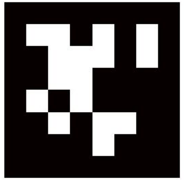


Major Technical Challenges

- Running SLAM with low error (1m FDE) at reasonable FPS (24+) on embedded board
  - Need to experiment with various methods and compare stereo camera vs. 2D LIDAR
  - Visual methods affected by lighting, LIDAR may work badly outdoors
  - **Mitigation:** Use fiducial markers to manually localize within campus
- Communications between robot/ground station
  - Inconsistent Wifi quality as the robot roams campus
  - **Mitigation:** Use a cellphone with a hotspot onboard the robot
- Creating and assembling the chassis
  - Need to build the robot from scratch, which is an involved mechanical engineering effort
  - **Mitigation:** Could buy an existing chassis or robot base like iRobot Create 2



Tag25h9



Tag36h11

Testing, Verification, and Metrics

Requirement	Testing Method	Metrics
Robot must deliver food on campus roads	Test robot driving up/down every slope on campus Test robot driving/path planning on sidewalk areas	0% failure rate on ascending/descending slopes with 2kg payload > 65% of time (s) on sidewalk than the grass (assuming robot goes half as fast on grass)
At least 30 minutes of battery life	Test robot with 2kg payload until battery voltage reaches low value.	Battery lasts > 30 mins, based on empirically determined human travel time of TSP.
Robot must arrive close to the drop-off point	Test robot and measure || ground truth goal location - end location ||	Assume Gaussian with $\Sigma = (1m)*\mathbf{I}$, better than landing within 1 std. dev randomly (>= 68%) Within 3m 100% of the time $\|\|door - road\|\|_2$
Robot must transport multiple deliveries at once	Attempting to deliver increasingly heavy orders weighed beforehand	Must hold and transport 2kg food at 1 ft^3
Food must be intact during delivery	Test the number of times food is damaged from delivery	Food must be intact 100% of the time.

Testing, Verification, and Metrics

Requirement	Testing Method	Metrics
Robot must not collide with pedestrians or other objects	Have pedestrians walk near the robot while it is in transit. Use LIDAR to inspect the minimum distance measurements in-transit	Check that the minimum distance to an obstacle is at least .25 m at all times
Minimal human intervention	Give robot orders to random locations and empirically record the number of interventions/distance	Check that the interventions/distance is at most 1 intervention/50 meters (significantly better than Kiwi’s 1 intervention/5 meters)
Robot must deliver food in a timely fashion	Given robot many random orders at once with varying payload weights and track the delivery time for each order	Each delivery time must be less than HRTT (human round trip time) measured in seconds
Robot must connect to ground station for emergency interventions and tracking	Have the robot do a full tour of campus while continuously pinging the ground station	Tail latency should be less than 750ms, enough time for robot to report emergency position, and enough time for intervention control to intervene before it goes off the sidewalk. (Assuming robot travels 2 ft/s on a 3 ft sidewalk)

Tasks and Division of Labor

- Advaith:
  - Design/CAD robot chassis
  - Localization algorithm
  - Path following (pure pursuit, LQR)
  - Collision Avoidance (ORCA, RL)
  - Low-level controls (PID)
- Sebastian:
  - Communication / network configuration + testing
  - Hardware driver bringup (motor controller pwm, imu, lidar/camera, gps)
  - *Collision Avoidance (ORCA, RL) - secondary*
- Michael:
  - Multi-order algorithm
  - Sensor data cleaning (kalman filtering)
  - Integration
  - *Communication / network configuration - secondary*
  - *Localization - secondary*
- Everyone:
  - Design/CAD robot chassis
  - Bill of Materials
  - Building chassis (aluminum, dc motors, acrylic plates, boards, battery)
  - Parts testing (sensors, motors, power supply)
  - Hardware driver bringup (motor controller pwm, imu, lidar/camera, gps)
  - Testing

Schedule

GrubTub	start	end
Proposal and Planning	**02/18/21**	**03/17/21**
Website Setup	02/18	02/19
Proposal Presentation Slides	02/21	02/21
Proposal Presentation (feat. Michael)	02/22	02/23
Finalize robot CAD Design	02/21	02/21
Test Camera/LIDAR	02/21	02/21
Finalize UI Design	02/25	02/26
Finalize parts list	02/23	02/24
Design Presentation (feat. Advaith)	02/26	03/03
Design Report	03/12	03/17
Design and Implementation	**02/20/21**	**04/13/21**
Milestone #1: Proof of Concept/C...	**02/20/21**	**03/14/21**
Robot line following in simulation	02/20	02/21
Determine nodes on the map/build...	02/20	02/20
Brute force multi-order algo simulat...	02/21	02/22
Multi-order algorithm proof of conc...	02/24	03/02
Integrate IMU to ROS	02/22	02/22
Proof of concept for global/local pl...	02/24	02/26
Order Structural Components	02/25	03/09
Order Electronic Hardware	02/25	03/09
Basic power test	03/11	03/12
Assemble Chassis	03/11	03/14
Milestone #2: Basic Integration	**03/10/21**	**03/25/21**
Integrate Wifi to Jetson Nano and c...	03/13	03/14
Integrate Camera to Jetson Nano	03/13	03/14
Get localization algorithm running ...	03/10	03/10
Build static map of campus	03/11	03/12
Get live perception feed to Jetson	03/15	03/15
Integrate motor power supply and ...	03/16	03/18
IMU data preparation	03/19	03/21
Integrate GPS to Jetson Nano	03/19	03/21
GPS data preparation	03/22	03/23
Implement low level PID controller ...	03/20	03/21
Integration test #0: Sensors and A...	03/24	03/24
Write joystick controller	03/23	03/23
Integration test #1: Basic joystick ...	03/25	03/25
Milestone #3: Final Implementat...	**03/03/21**	**04/13/21**
Get robot to localize within map	03/26	03/27
Independent local collision avoidan...	03/28	04/01
Independent path following on rob...	03/30	04/02
Get heartbeat back to groundstati...	03/26	03/26
Implement multi-order algorithm	03/03	03/06
Get groundstation to send global p...	03/27	03/27
Implement robot state machine	03/22	03/27
Validate state machine	03/28	03/28
Implement groundstation state ma...	03/27	03/29
Integration test #2: Follow basic pa...	04/03	04/03
Integrate local planners	04/05	04/06
Integration test #3: Follow basic pa...	04/07	04/10
Interim Demo	04/12	04/12
Write groundstation UI	03/16	03/17
Integration test #4: Follow optimal...	04/12	04/13
Testing and Benchmarking	**04/12/21**	**04/15/21**
Ensure battery life meets 30 minute ...	04/12	04/14
Test robot driving up/down every slo...	04/12	04/14
Test robot driving/path planning on s...	04/12	04/14
Human Interaction Testing (5 tria...	**04/12/21**	**04/14/21**
Test pedestrian avoidance	04/12	04/14
Test delivery times	04/12	04/14
Test number of human interventio...	04/12	04/14
Basic Field Testing (20 trials)	**04/12/21**	**04/15/21**
Test robot final displacement error	04/12	04/15
Test food intactness	04/12	04/15
Test load capacity	04/12	04/15
Test groundstation intervention la...	04/12	04/15
Final Report	**04/12/21**	**05/02/21**
Final Presentation Slides	04/29	05/01
Final Presentation (feat. Sebastian)	05/02	05/02
Record videos	04/12	04/15
Edit video	04/24	04/29
Final Video Submission	05/02	05/02
Final Report	05/02	05/02