AA 274: Turtlebot Demo 2019

Team 4

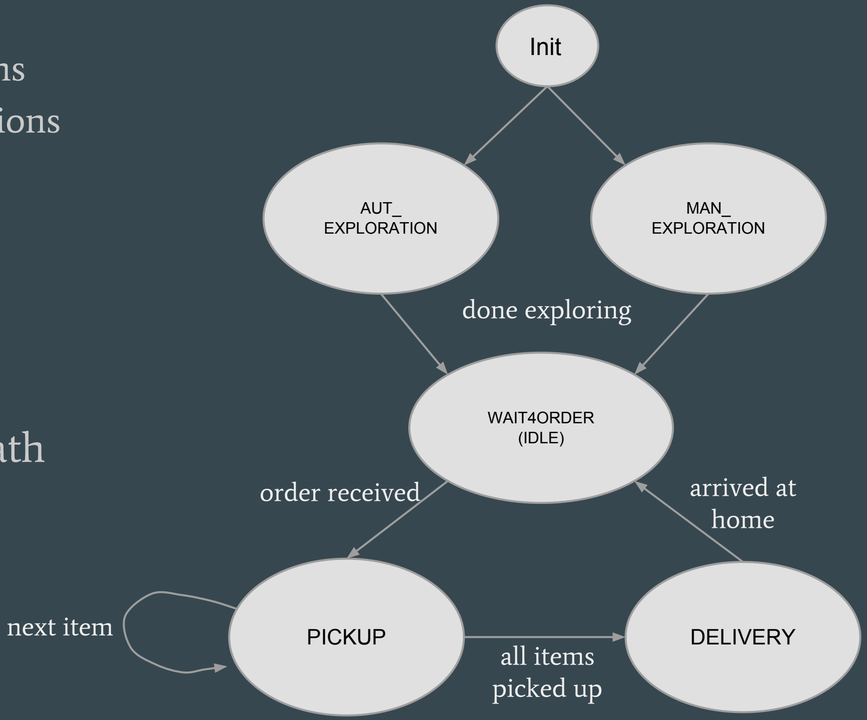
•••

Overview of the autonomy stack

- Hardware
    - Jetson
    - Raspberry Pi
    - Velodyne LIDAR
- Software (main nodes)
    - supervisor.py
    - gmapping
    - navigator.py
    - pose_controller.py
    - detector_resnet.py
    - Custom package: autonomous_exploration

Additions to reach baseline requirements

- Parallel state machine
- Other minor additions to **supervisor.py**:
  - Logging of detected food items and their locations
  - RViz Markers for food, robot and nav_goal positions
  - Subscriber to topic used to submit orders.
  - Subscriber to topic used to manually indicate when we are done exploring.
- Logic to prevent conflicting commands from **pose_controller.py** and **navigator.py**
- Back up when **navigator.py** fails to compute a path
- Minor tweaking of thresholds in **supervisor.py**, **navigator.py** and **pose_controller.py**



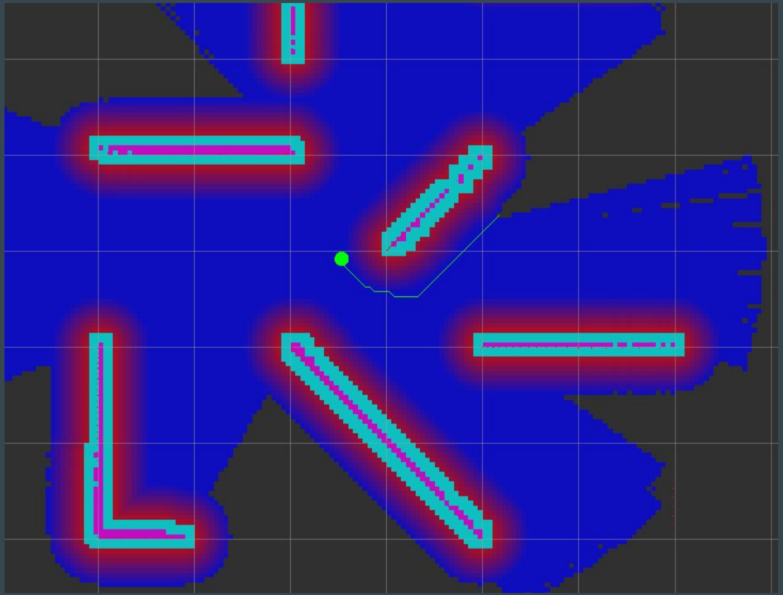
Extension: Autonomous Exploration Package

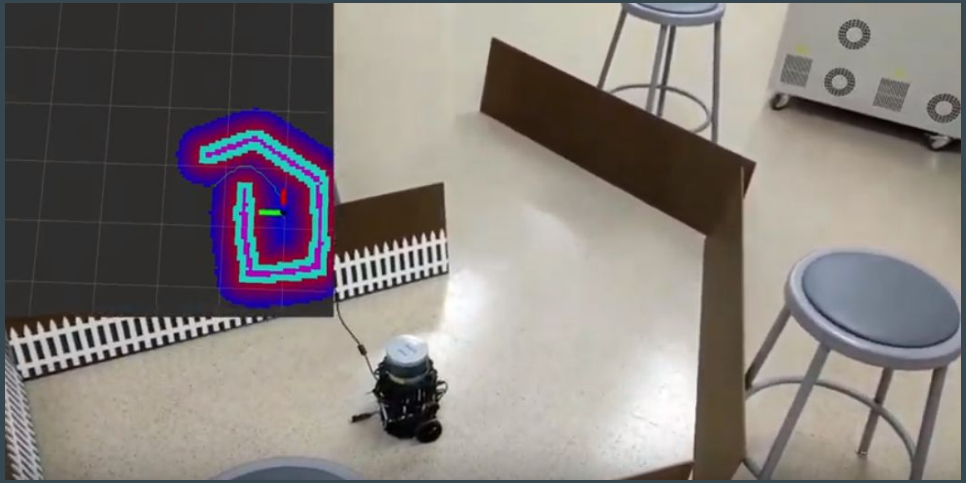
- Uses a cost map (based on gmapping) and a modified Dijkstra's algorithm to autonomously plan paths to unknown points.
- Two nodes:
  - **costmap_2d.py** - Subscribes to the map created by gmapping and publishes a cost map with high cost on and around obstacles.
  - **trajectory_planner.py** - Subscribes to the cost map, runs a modified Dijkstra's algorithm to find the shortest path to the closest unknown point and publishes this point (as well as the path).

    The node **navigator.py** subscribes to the published goal point and implements a path to this point.

Autonomous Exploration: Results

- For details, see HW4 submission.
  - Update since then: now using **navigator.py** instead of **simple_local_planner.py**





GitHub Repos

Project:
https://github.com/varununayak/asl_turtlebot/tree/project

Autonomous Exploration:
https://github.com/AlbinJagesten/autonomous_exploration