

What is the problem: Lane following while obeying signs and avoiding obstacles in the lane

Your approach: Combine previous labs that used the lane detection to stay in the lines as well as object detection to avoid the obstacles and sign detection to see what the sign is and obey what it says.

Environment/World:

`ros2 launch turtlebot3_gazebo turtlebot3_autorace_2020.launch.py`

A clearly detailed division of sub-tasks among the group members:

- Joey Hyun - Sign Detection
- Cameron Schwindaman - Lane Following
- Tasmia Rahman - Obstacle detection

A brief description of what your ROS package will do along with the overall tentative ROS architecture:

We will combine previous labs to create a ROS package that will control the turtlebot to make sure it stays in the lanes, avoids objects, and sign detection to obey the signs.

The ROS name will be `autonomous_driving` with all of the basic things that are in a ROS package including a launch file.

We are going to use real-time detection using sensor data, with our fallback being to use a pre-map of the environment.

Sign_detector_node:

- Subscribes: `/image_raw`
- Publishes: `/signs/detected`
- Algorithm: Ultralytics YOLO (YOLOv11), Used Roboflow to label/annotate traffic sign images and organize datasets.
- Fallback : use a color-based detection instead: red for stop signs, yellow for caution, etc.

Lane_detector_node:

- Subscribes: `/image_raw`
- Publishes: `/lane_vel`
- Algorithm: Use PID to follow the lane
- Fallback: Use pure pursuit instead of PID as the lane following algorithm

Obstacle_detector_node:

- Subscribes: `/scan`
- Publishes: `/objects_detected`
- Algorithm: detect obstacles in real time and makes list of crumbs to `/objects_detected`
- Fallback: try a different stopping distance if the object is not stopping in time or if the object is not detecting properly, lower the speed so that the node has enough time to process `/scan`.

Decision _making_node:

- Subscribes: /signs/detected, /lane_vel, /objects_detected
- Publishes: /cmd_vel

Timeline:

First implementation of nodes 11/15 - 11/25

Presentation done by 11/25 - 12/01

Practice presentation done by 12/01 - 12/03

Final report done by 12/03 - 12/10