Exercise 5: Using LIDAR to position the vehicle

Hongqing Liu, Weichen Liu, Jingwei Bao

University of Illinois Urbana Champaign,
hl85, wl45, jb52

Purpose

The primary goal of this exercise is to design a detecting and tracking system. The vehicle used LIDAR as sensor, then entered a complex area with cardboard boxes as obstacles and assign tasks. The specific tasks are: 1. Find the boxes in the LIDAR, 2. Drive between the boxes and in a circle around one box, and 3. Stop between the boxes.

Method

LIDAR Perception:

After obtaining point cloud data through LIDAR, there are three steps to filter out unwanted points:

1. For each frame, points that are more than 7 meters away from the sensor or within 1 meter of proximity to the sensor will be filtered out. The threshold 7 meter is because points too far is not necessary for detection, and the threshold of 1 meter is set to avoid detecting the vehicle itself.

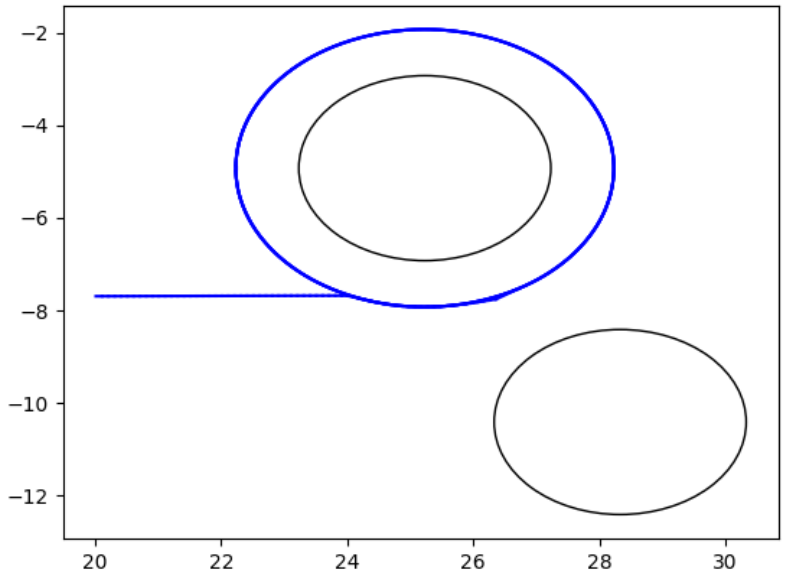
2. A search area is defined to specify the total area of the field being worked on (search_area = [[3,30],[-11,-1]]).

3. Points in the current point cloud with a z value lower than -1.5 are filtered out as they are considered part of the ground.

Then, we use the k-means method to cluster these remaining points and find the exact positions of two cardboard boxes within the field.

Path Planning: The path planning consists of two parts, composed of a circle and a straight line. Firstly, Rapidly-exploring Random Tree (RRT) method is used to achieve path following for the straight section, guiding the vehicle from any position to a location between two obstacles that allows passage. Then, the prototype path generated around the obstacle is seamlessly connected. As the

vehicle follows the waypoints, the basic logic is to find the closest point that has not been traversed. Therefore, at the end of the straight path, it naturally transitions to the waypoint closest to the circle path. The final generated path is shown in the following figure.



Waypoints Control: Control the vehicle to sequentially follow each way points of the plotted path.

Ensure the range of finding points: Force the system was only able to find waypoints in next 100 waypoints of current one.

Set variable look_ahead: The reason to set this variable is to deal with the problem of system delay. Since subscribing and publishing messages through ROS takes time, the system can only track points behind the vehicle when path planning is done. So look_ahead is designed to predict a brief position of vehicle when system processings are done, and it will search for waypoints within a radius of 0.3 meters around that position.

Heading control: Calculate the angle value between the current location and the planned next way point, and call the variable f_delta_deg. There are three

situations: f_delta_deg > 30 ,f_delta_deg < -30, or f_delta_deg between -30 and 30, we sequentially give command 2 (turn left), 0 (turn right), and 1 (stay current direction).

Speed Control: Control both acceleration and desired speed to make sure the velocity of the vehicle is relatively low, and brake when the vehicle is moving close to the end of the path.

Video Link

Scan points success: http://www.youtube.com/watch?v=3L7la2ZPhWglist=PL5IM62rCQfXrOYlQUh9B-AyZ2Xzz-sQe6index=10My YouTube Video
Outside view: http://www.youtube.com/watch?v=mW_gi3jvbtIlist=PL5IM62rCQfXrOYlQUh9B-AyZ2Xzz-sQe6index=9My YouTube Video
In-car view: http://www.youtube.com/watch?v=h49d3dRFL0slist=PL5IM62rCQfXrOYlQUh9B-AyZ2Xzz-sQe6index=11My YouTube Video