Carla Lab

Team: Virtual Fast and Keyboard Furious

In this lab, we have implemented

1. Pure Wall Following
2. Wall following with Instructions

To run this package you will require the f110-skeletons-spring2020 package in catkin_ws.

To run pure wall following:
roslaunch virtualfastkeyboardfurious_wall_following pure_wall_following.launch

To run Wall following with Instructions:
roslaunch virtualfastkeyboardfurious_wall_following instruction_wall_following.launch

Pure Wall Following is addressed in previous Individual Assignment.

In instruction_wall_following, we make use of 5 components.

1. Finding Gaps
2. Finding the type of junction based on types of gap perceived by the robot
   i. 'T' : when there are two gaps.
   Ii. 'CROSS': crossroad type interaction when there are three gaps.
   iii 'NO_JUNCTION': Straight tube
3. Turning according to Instruction and Junction
4. Monitoring Turn initiation and completion.

Finding Gaps:

Similar to the previous assignment, we publish all gaps. Currently the best gap is displayed.
Parameter Changes:

1. Additionally we add gap angles to gaps messages. Gap angle helps us identify which side of the robot the gap lies: left, right or in the middle ('center')
2. We also restrict maximum range 10m to find gaps within robot's vicinity and not far away.
   #Gap Finding Params
   MIN_GAP_SIZE : 1.0
   MAX_GAP_SIZE : 10.0

3. Currently the best gap is displayed by RViz.

Finding Type of Junction:

Here, gaps length and gap angle help identify if there is no junction or ‘T’ or ‘Cross’ junction. Based on that we initiate the robot to instruction following procedure.
Parameter Explanation:

1. First we check that the gap from list of gaps are within bounds of max and min gap_size and.
2. Default junction type is no Junction, meaning no gaps at turning were found.
3. Then we check if there is gap at left, right or center of the robot based on gap angle meeting these limits.
   R_GAP_ANGLE_LIMITS : -20
   L_GAP_ANGLE_LIMITS : 20

4. For ‘cross’: gap is on three sides.
5. For ‘T’ 2 gaps can be a combination eg: Center or left.

Turning According to Instruction:

Based on junction type, here we execute the instruction from the csv file.
We access the csv file and obtain direction and velocity information.
Parameter Explanation:

1. If cross or T junction, then we are bound to turn. So we calculate the initial heading of robot.
   a. We set the ros parameter ‘INSTRUCTION_STATUS’ as ACTIVE.
   b. We execute the explicit instruction
2. If no junction, then default
   a. Default velocity and direction is executed.
3. Given ‘INSTRUCTION_STATUS’ active_turn_monitoring() is a function that is called to monitor turn execution and completion.

Turn Monitor System

active_turn_monitoring() is function that checks if turn is completed while following explicit instructions and setting ‘INSTRUCTION_STATUS’ to INACTIVE so the robot can follow default instructions.

There are two types of turn: Left and RIght. After turning, the robot heading changes. If center following is the instruction then CENTER_COUNTER keeps count of how many center executions will be considered before returning to default.

Parameter Explanation:

1. delta_heading >= TURN_COMPLETE_ANGLE. Where delta heading is change in heading from beginning of turn to current heading. Here we see if this change is greater than 60 degrees.

PID Control.

This was set after experimenting within the simulator as explained in the previous assignment.
KP : 0.5
KD : 0.01