The report of P545 mid-term lab

Yubin Zou, Pei Zhang

Introduction

This report mainly introduces the methods I use in my mid-term lab. These methods include how to solve the delay of cart’s action, and the control of speed and steering. At last, the experimental result is given to demonstrate the efficiency of these methods.

Methods

1. The model of cart in my driver.
   The model my driver uses is below:
   
   ![Diagram of waypoint and heading direction]
   
   Where $\alpha \in [0, 360)$

   The conversion section is below:

   ```python
   x = waypoint_latlon[1] - current_latlon[1]
y = waypoint_latlon[0] - current_latlon[0]
desired_direction = distance.distance([[0,0], [x,y]].forward_azimuth
heading = float(self.compass_p['heading'])

if desired_direction + heading - 90 > 360:
    angle_to_target = desired_direction - 90 + heading - 360
else:
    angle_to_target = desired_direction - 90 + heading

if angle_to_target >= 360:
    angle_to_target = angle_to_target - 360
   ```
2. Make a turn in advance

In my driver, the cart will make a turn in advance, because there is a significant delay of steering, when the distance from the cart to the target waypoint is 4 meters, the driver will send a signal of steering in advance.

```python
if (desired_distance < LBO + 4):
    print "The", self.current_waypoint_ind, "th waypoint has already been reached!"
    self.last_waypoint = waypoint_latlon
    self.current_waypoint_ind += 1  # go to the next waypoint
    if(self.current_waypoint_ind == waypoints_num):
        self.current_waypoint_ind = 0
    if(self.last_waypoint == self.start_latlon):
        self.laps_counter += 1
    if self.laps_counter == 4:
        self.jdriver_s['percent_throttle'] = 0.0
        self.jdriver_s['turn_radius_inverse'] = 0.0
        return

This is the section of this method. When the distance to the target waypoint is less than LBO + 4, this waypoint is seen to have already been reached, and then set the next waypoint to be the target waypoint.

3. Speed up and slow down

For a more efficient and quick navigation, the speed that my driver sets is various at different situations. Consider the distance to the current target waypoint is L1, the distance to the last target waypoint is L2, the angle to the current target waypoint is $\alpha$.

The real rules that my driver uses are below

<table>
<thead>
<tr>
<th>Speed</th>
<th>situations</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>($0 &lt; L1 &lt; 8$) OR ($0 &lt; L2 &lt; 3$ AND ($\alpha &gt; 5$ OR $\alpha &lt; 355$))</td>
</tr>
<tr>
<td>60</td>
<td>(ALL of other situations)</td>
</tr>
</tbody>
</table>

The section of this method is below:

```python
if 0 < desired_distance < 8 or (0 < leaving_distance < 3 and (angle_to_target > 5 or angle_to_target < 355)):
    self.jdriver_s['percent_throttle'] = 40.0
else:
    self.jdriver_s['percent_throttle'] = 60.0
```

4. Steering control

In my driver, the conversion relationship between steering value and angle to the target waypoint is liner. Consider the steer throttle as S and the angle to waypoint is $\alpha$, then the chart below describes it.

<table>
<thead>
<tr>
<th>S</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>($W/90$)*0.5</td>
<td>[0.90]</td>
</tr>
<tr>
<td>0.5</td>
<td>[90,270]</td>
</tr>
<tr>
<td>((360-W)/90)*0.5</td>
<td>[270,360]</td>
</tr>
</tbody>
</table>
The graph is

```
The code section is below:

def compute_turn_rad(self, angle_to_target, steer_val_max = 0.5):
    if 0 < angle_to_target and angle_to_target < 90:
        #steer_val = -(6.1728e-5)*(angle_to_target-90)**2 + steer_val_max
        steer_val = (angle_to_target/float(90)) * steer_val_max
    elif 270 <= angle_to_target and angle_to_target < 360:
        #steer_val = (6.1728e-5)*(270-angle_to_target)**2 - steer_val_max
        steer_val = -((float(360) - angle_to_target)/float(90)) * steer_val_max
    elif 90 <= angle_to_target and angle_to_target <= 180:
        steer_val = steer_val_max
    else:
        steer_val = -steer_val_max
    return steer_val
```

**Experimental results:**

The waypoint list is

```
self.waypoint_lation_list = [[1,39.161917,-86.5221206333,1.5,3.0],
[2,39.1818975,-86.521724,1.5,3.0],
[3,39.182143,-86.5217033333,1.5,3.0],
[4,39.182199,-86.5220985,1.5,3.0],
[5,39.1819156667,-86.522309,1.5,3.0],
[6,39.1819645,-86.522398,1.5,3.0],
[7,39.1820415,-86.5223095,1.5,3.0],
[8,39.1821313333,-86.5223926667,1.5,3.0],
[9,39.1822116667,-86.522302,1.5,3.0]]
```
I run this driver on the cart, the path the cart runs is