CPS Lab Report

Dian Yuan, 03749889

Sept. 3, 2022

1 Day 3 - "Follow the gap"

1.1 General idea

The car should always steer where the distance between him and its environment is the biggest. However, since we don't want the car driving backwards alone the track, we should omit the data detected by the lidar for the rear. We can use Eqs.(1) to find the index corresponding to the maximum gap.

$$\text{index}_{\max} = \arg \max_{270 \leq i \leq 810} \text{ranges}[i] \tag{1}$$

1.2 Index correction

However, considering that the car itself is not a prime, there are times when we do not want the car to go exactly in the direction of the maximum gap. As shown in Fig.1, in this case, the car is likely to hit the inner track.

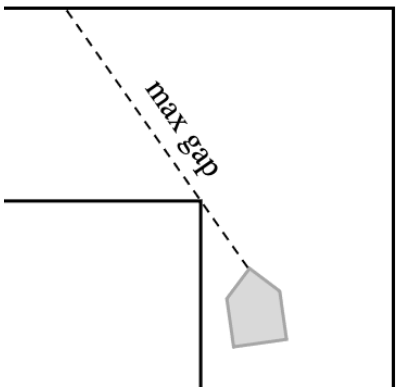

Figure 1: Collision is possible if car follows the direction of maximum gap during the curve

To avoid this, we would like to correct the index of the maximum gap, which should be reduced appropriately for left turns and increased for right turns on the contrary. Here we use the correction

of Eqs.(2), for the left and right curves, each has a correction of 7°, and for the straight, the absolute value of the angle is reduced so that the horizontal movement on the straight can be reduced.

$$\text{index}'_{\max} = \begin{cases} \text{index}_{\max} - 21 & \text{index}_{\max} > 570 \\ \text{round}(0.3 \cdot \text{index}_{\max}) + 378 & 510 \leq \text{index}_{\max} \leq 570 \\ \text{index}_{\max} + 21 & \text{index}_{\max} < 510 \end{cases} \tag{2}$$

Now we are able to get the maximum gap and the corresponding direction.

$$g_{max} = \text{ranges}[\text{index}'_{\max}] \tag{3}$$

$$orientation_{max} = -\pi + \frac{\pi}{540}\text{index}'_{\max} \tag{4}$$

Then the control law of the car can be expressed as

$$\text{steering\_angel} = k \cdot orientation_{max} \tag{5}$$

where k is the angle gain and the steering_angle should not exceed the set maximum steering angle.

1.3 Collision detection

The control law shown in Eqs.(5) is an open-loop control, and the car may not actually be moving in exactly the direction we set. Therefore, this method does not guarantee that the car does not collide. In order to drive the car safely, we also need to detect the shortest distance of the car from the track. Similar to maximal gap, we have

$$\text{index}_{\min} = \arg \min_{270 \leq i \leq 810} \text{ranges}[i] \tag{6}$$

$$g_{min} = \text{ranges}[\text{index}_{\min}] \tag{7}$$

$$orientation_{min} = -\pi + \frac{\pi}{540}\text{index}_{\min} \tag{8}$$

We want to get the "force" in the opposite direction when the car is too close to an obstacle on one side, and the "force" is greater when the gap is smaller or the obstacle is closer to the front of the car. It is natural to think of inverse proportional functions. We propose a collision control law as shown in Eqs.(9).

$$\text{steering\_angel} = -\frac{0.1}{g_{min} \cdot orientation_{min}} \quad (\text{when } g_{min} < 0.7) \tag{9}$$

Noting that the absolute value of $orientation_{min}$ reflects the angle of the nearest obstacle and its sign indicates the direction, this equation just meets our needs.

Combined with the collision detection mechanism, the car can travel safely on the track at speed of up to around 4.5m/s.

1.4 Speed up

Although the collision detection mechanism is added, in some sharp corners, it is not guaranteed that the car will pass at a higher speed, so the speed of the car needs to be dynamically adjusted in order to achieve the fastest lap speed. As is our daily driving habit, slowing down at curves. Therefore, we designed different control strategies according to the different situations that may be encountered in a racing car. First we introduce a new distance, which represents the current gap directly in front of the car.

$$g_{front} = ranges[540] \tag{10}$$

When $g_{front} > 5.5$ and $|orientation_{max}| < 0.07$, we believe that the car is in a straight path and facing the right direction, in the situation we set the speed to maximal 7m/s and steering_angel to zero.

When $g_{min} \leq 0.25$ or $g_{front} < 2.0$, at this point the car is about to hit an obstacle, tje steering_angle will be set to the maximum gap direction (left or right), and the speed will be limited within 5.5m/s.

In other cases, depending on the value of $orientation\_max$, different angle_gain and speed will be set. Tab.1 summarizes all the control strategies. If multiple states are satisfied at the same time, the table indicates the priority of the judgment from top to bottom.

Table 1: Control laws

State	**Control laws**
STRAIGHT: $g_{front} > 5.5$ and $\|orientation_{max}\| < 0.07$	speed = 7m/s steering_angle = 0
MAX_TURN: $g_{min} \leq 0.25$ or $g_{front} < 2.0$	speed = min(5.5m/s, previous speed) |steering_angle| = 0.4189
COLLISION: $g_{min} < 0.7$	speed = min(5.5m/s, previous speed) steering_angel $= -\frac{0.1}{(g_{min} \cdot orientation_{min})}$
BIG_TURN: $orientation_{max} > \frac{\pi}{5}$	speed = 4.9m/s steering_angle $= 0.35 \cdot orientation_{max}$
LITTLE_TURN: rest situations	speed = 5.5m/s steering_angle $= 0.27 \cdot orientation_{max}$

With this set of control laws, our car should be able to run the fastest lap time on the track within 18s.

2 Day 4 - "Online driving mode switching"

To enable online mode switching, we created a new node called mux, which acts as the central hub of the control system. Now, the keyboard node sends only key value read at this time under the /mode topic, and the follow_gap node continues to send the same information, except under the /follow_gap_drive topic. From a safety point of view, manual control should have a higher priority than autopilot. We used a variable "keyboard_flag" of type bool to achieve this. It is only set to zero when the keyboard presses "f", so that the control signal from /follow_gap_drive will be forwarded to the /drive topic, after which pressing any key that is not "f" will reset the flag back to true and the keyboard will take over control.

It should be noted in this node that the frequency of /follow_gap_drive published is 1000hz in line with the frequency of lidar signal update, so in order to make the follow_gap algorithm also have good control effect at this time, it is necessary to set the publish frequency of topic /drive in the mux node to 1000hz as well.

3 Day 5 - "Automatic Emergency Braking"

In this section we create a node named aeb, which will send a bool signal to the mux node indicating whether the car at the current position is safe or not. It also subscribes to the /drive topic published by mux to get information about the speed.In the mux node, the car will be controlled to emergency brake and return to a safe position based on the information from /aeb. The AEB system is only activated in keyboard mode.

3.1 Forward AEB

We can still use Eqs.(6)(7)(8) to obtain the minimum gap information during the forward motion. According to the definition of TTC, we can calculate the TTC of the car as

$$TTC = \frac{gap_{min}}{speed \cdot \cos\left(orientation_{min}\right)} \tag{11}$$

Considering the width of the car and the location of the lidar, we set the threshold of the forward TTC_f as 0.35, i.e. when $TTC < TTC_f$, the aeb node would send a message indicates unsafe to the mux node and mux would stop the car immediately. But relying on TTC alone does not guarantee that the car is completely safe. Consider the case shown in Fig.2, when the car is moving at a small angle to the track, the calculated TTC will be much larger than the actual TTC - at this point, the width of the cart seriously affects the TTC calculation. Therefore, we also add $g_{min} < 0.25$ to the criterion of determining whether the car is safe or not.

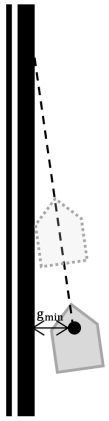

Figure 2: TTC Issue

3.2 Backward AEB

Similar to the forward AEB, the backward AEB requires finding the minimum gap behind. In this case we should find the minimum distance from index $[0, 270) \bigcup (810, 1080]$ of ranges. Considering that the lidar is installed at the front of the car, the threshold of backward TTC should be a little larger, we set it to 0.55.

3.3 Back to safe place

In order to facilitate the continued use of keyboard control and to avoid the car falling into a dead-end cycle of continuous AEB triggering, we want to be able to automatically return the car to a relatively safe position. The idea is to record the velocity at the moment the AEB is triggered and then give the car the same speed in the opposite direction until the TTC is greater than another larger threshold. After some experimentation, we set this threshold to 4.0.

4 Summary

The framework of the whole simulation experiment is shown in Fig.3. Once you launch all the nodes, you will find the car stationary at the starting point. You can either choose to use the "w, a, s, d" buttons on the keyboard for manual control - thanks to the AEB system this will be very safe, or press "f" the car will drive automatically. You can also switch back to keyboard control mode by pressing any key(except "f") in autopilot. However, due to the aggressive algorithm in follow_the_gap and the shortcomings of the algorithm itself, there is no guarantee that the car will complete a lap after the autopilot is triggered at any position of the keyboard control.

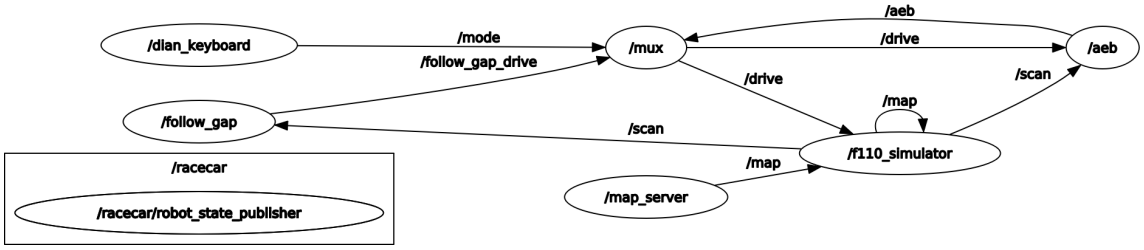

Figure 3: General project framework (ROS Graph)