

2. Lidar obstacle avoidance fun gameplay

2. Lidar obstacle avoidance fun gameplay

- 2.1 Description of program function
- 2.2. Program Code reference path
- 2.3. Program Startup
 - 2.3.1. Car start command
 - 2.3.2. Modifying parameters on the VM side
 - 2.3.3. View the topic communication node graph
- 2.4. Core Source Code Analysis

2.1 Description of program function

After the program starts, the car will move forward, when there is an obstacle in the detection range, it will adjust its attitude, avoid the obstacle, and then continue to move forward.

Enable/pause this function by starting the Dynamic Parameter Adjuster on the virtual machine and clicking [Switch].

In addition, the [L1] button on the handle locks/turns on the car's motion controls. When motion control is enabled, the function is locked; This function can be turned on when the motion control is locked.

2.2. Program Code reference path

After SSH connection car, the location of the function source code is located at,

```
#python文件#python file
/userdata/yahboomcar_ws/src/yahboomcar_laser/yahboomcar_laser/laser_avoidance.py
#launch文件#launch file
/userdata/yahboomcar_ws/src/yahboomcar_laser/launch/laser_avoidance_launch.py
```

2.3. Program Startup

2.3.1. Car start command

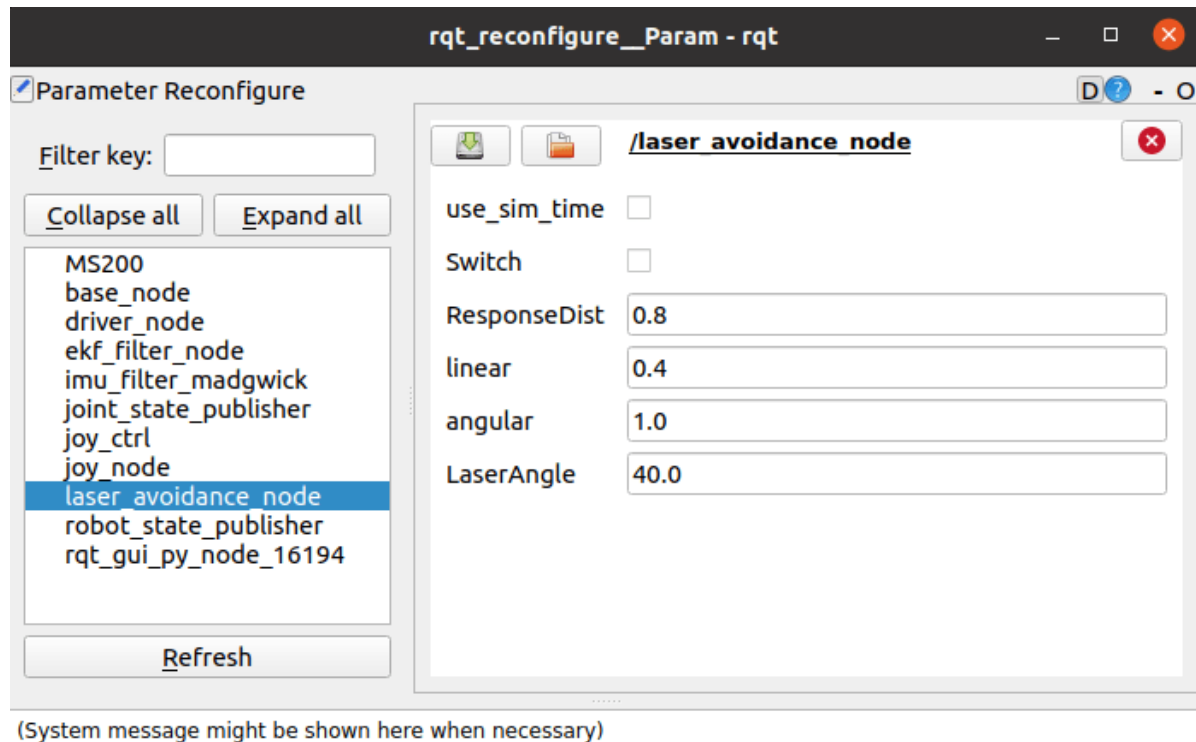
After SSH connects to the car, terminal input,

```
ros2 launch yahboomcar_laser laser_avoidance_launch.py
```

2.3.2. Modifying parameters on the VM side

On the virtual machine, open the dynamic parameter adjuster, open the terminal input,

```
ros2 run rqt_reconfigure rqt_reconfigure
```



The meaning of each parameter is as follows.

parameter name	parameter meaning
Switch	Play switch
ResponseDist	Obstacle detection distance
linear	The linear velocity
angular	Angular velocity
LaserAngle	Radar detection Angle

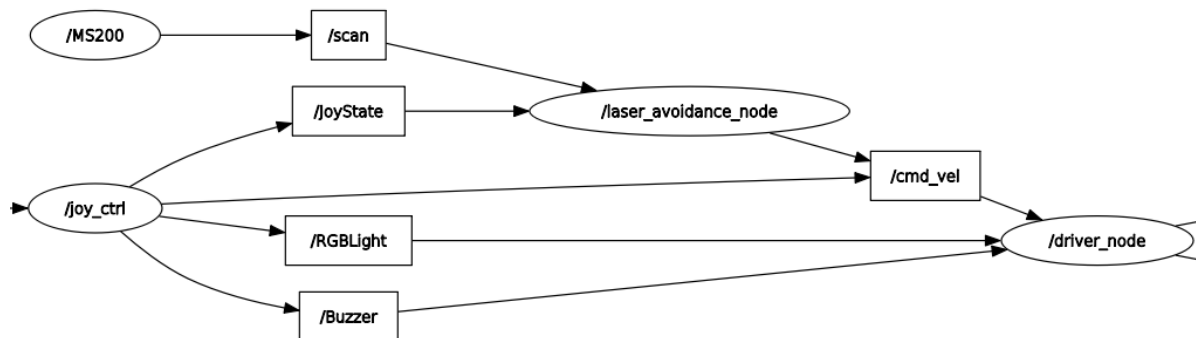
The above parameters can be adjusted. By checking or unchecking [Switch], the radar obstacle avoidance function can be enabled or suspended.

The other four need to be set when the decimal, modify, press the enter key or click the blank space can be written.

2.3.3. View the topic communication node graph

Virtual machine terminal input,

```
ros2 run rqt_graph rqt_graph
```



2.4. Core Source Code Analysis

Mainly look at the radar callback function, which explains how to obtain the obstacle distance information at each Angle.

```
def registerscan(self, scan_data):
    if not isinstance(scan_data, LaserScan): return
    self.right_warning = 0
    self.left_warning = 0
    self.front_warning = 0

    ranges = np.array(scan_data.ranges)
    for i in range(len(ranges)):
        if ranges[i] < self.ResponseDist: #range[i]就是雷达扫描的结果，这里指的是距离信息
            #range[i] is the result of radar scanning, here refers to the range information
            angle = (scan_data.angle_min + scan_data.angle_increment * i) * 180 / pi #雷达的信息的angle是弧度制，这里要转换成角度进行计算
            # The angle of radar information is a radian system, which should be converted into an Angle for calculation
            if angle > 180: angle = angle - 360 #angle是根据雷达的结构来设定判断范围的
            #angle is based on the structure of the radar to set the judgment range
            if -self.LaserAngle < angle < -20:
                self.right_warning += 1
            elif abs(angle) <= 20:
                self.front_warning += 1
            elif 20 < angle < self.LaserAngle:
                self.left_warning += 1

    if self.Joy_active == True or self.Switch == False:
        if self.Moving == True:
            self.pub_vel.publish(Twist())
            self.Moving = not self.Moving
        return

    self.Moving = True
```

