2. Radar obstacle avoidance

Introduction to radar obstacle avoidance gameplay:

- Set lidar detection angle and response distance
- After turning on the car, the car will drive straight when there are no obstacles.
- Determine the direction in which the obstacle appears on the car (front left, front right, right in front)
- Respond according to the position of the car where obstacles appear (turn left, turn right, turn left in a large circle, turn in a large right circle)

2.1. How to use

Note: The [R2] of the remote control handle has the [pause/start] function of this gameplay. Due to the problem of movement method, the case in this section does not support the Ackerman model. Different models will have different parameter ranges, but the principle is the same; take the X3 McLunner as an example.

```
#Raspberry Pi 5 master needs to enter docker first, please perform this step
#If running the script into docker fails, please refer to ROS/07, Docker tutorial
~/run_docker.sh
```

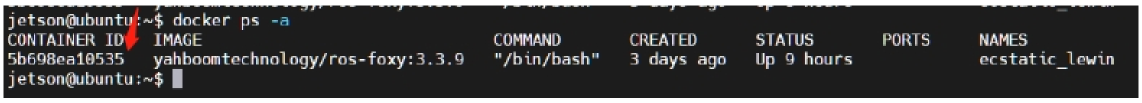
Start with one click (robot side). After executing the command, the car starts to move.

```
roslaunch yahboomcar_laser laser_Avoidance.launch
```

<PI5 needs to open another terminal to enter the same docker container

1. In the above steps, a docker container has been opened. You can open another terminal on the host (car) to view:

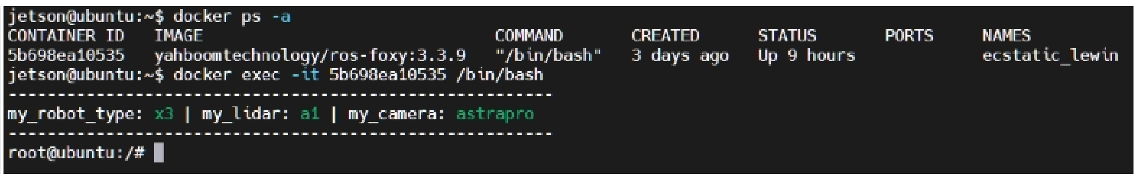
```
docker ps -a
```



```
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE                                   COMMAND       CREATED      STATUS     PORTS     NAMES
5b698ea10535   yahboomtechnology/ros-foxy:3.3.9        "/bin/bash"   3 days ago   Up 9 hours           ecstatic_lewin
jetson@ubuntu:~$
```

2. Now enter the docker container in the newly opened terminal:

```
docker exec -it 5b698ea10535 /bin/bash
```

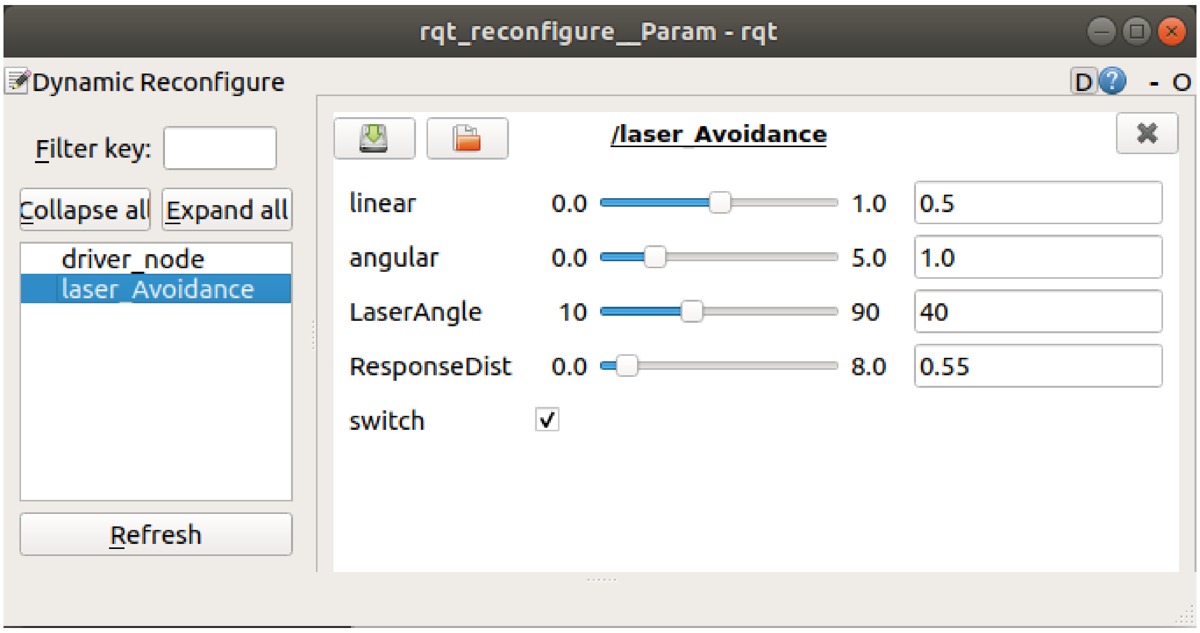


```
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE                                   COMMAND       CREATED      STATUS     PORTS     NAMES
5b698ea10535   yahboomtechnology/ros-foxy:3.3.9        "/bin/bash"   3 days ago   Up 9 hours           ecstatic_lewin
jetson@ubuntu:~$ docker exec -it 5b698ea10535 /bin/bash
-------------------------------------------------------
my_robot_type: x3 | my_lidar: a1 | my_camera: astrapro
-------------------------------------------------------
root@ubuntu:/#
```

After successfully entering the container, you can open countless terminals to enter the container.

Dynamic debugging parameters

```
rosrun rqt_reconfigure rqt_reconfigure
```



linear	0.0	1.0	0.5
angular	0.0	5.0	1.0
LaserAngle	10	90	40
ResponseDist	0.0	8.0	0.55
switch	✓		

Parameter analysis:

Parameters	Range	Parse
[linear]	[0.0, 1.0]	Car linear speed
[angular]	[0.0, 5.0]	Car angular speed
[LaserAngle]	[10, 90]	Lidar detection angle (left and right angles)
[ResponseDist]	[0.0, 8.0]	Car response distance
[switch]	[False, True]	Car movement [Start/Pause]

In the box in front of [switch], click the value of [switch] to True, and the car will stop. [switch] Default is False, the car moves.

- Parameter modification

When the parameters are adjusted to the optimal state, the corresponding parameters are modified into the file, and no adjustment is required when using again.

According to the optimal parameters of the [rqt_reconfigure] debugging tool, enter the [scripts] folder of the [yahboomcar_laser] function package and modify the parameters corresponding to the [laser_Avoidance.py] file, as shown below

```
class laserAvoid:
    def __init__(self):
        rospy.on_shutdown(self.cancel)
        ... ...
        self.linear = 0.5
        self.angular = 1.0
        self.LaserAngle = 40
        self.ResponseDist = 0.55
```

[rqt_reconfigure] Modification of the initial value of the debugging tool

```
gen.add("linear", double_t, 0, "linear in robot", 0.5, 0, 1.0)
gen.add("angular", double_t, 0, "angular in robot", 1.0, 0, 5.0)
gen.add("LaserAngle", int_t, 0, "LaserAngle", 40, 10, 90)
gen.add("ResponseDist", double_t, 0, "ResponseDist", 0.55, 0, 8)
gen.add("switch", bool_t, 0, "switch in rosbot", False)
```

Enter the [cfg] folder of the [yahboomcar_laser] function package and modify the initial values of the parameters corresponding to the [laserAvoidancePID.cfg] file.

```
gen.add("linear", double_t, 0, "linear in robot", 0.5, 0, 1.0)
```

Take the above article as an example to analyze

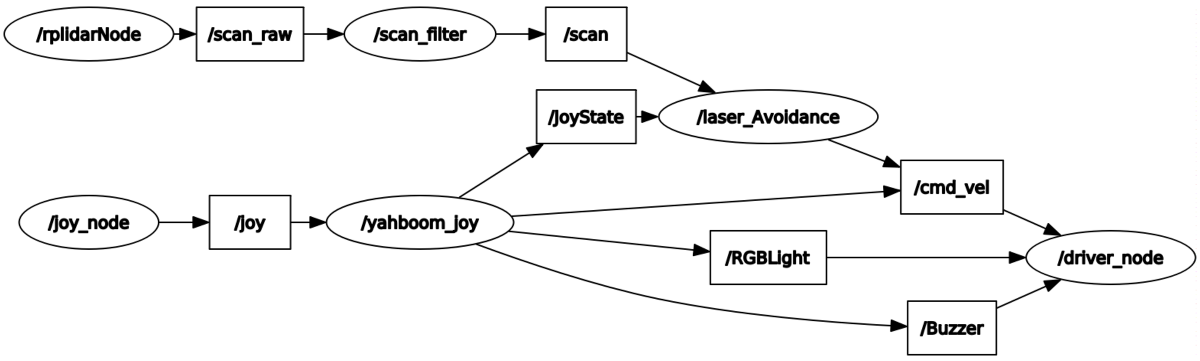
Parameters	Analysis	Corresponding parameters
name	The name of the parameter	"linear"
type	parameter data type	double_t
level	a bitmask passed to the callback	0
description	A description parameter	"linear in robot"
default	Initial value for node startup	0.5
min	parameter minimum value	0
max	parameter maximum value	1.0

Note: After modification, you must recompile and update the environment to be effective.

```
cd ~/yahboomcar_ws
catkin_make
source devel/setup.bash
```

Node view

```
rqt_graph
```



【laser_Aviodance】node analysis

- Subscribe to lidar data

- Subscribe to handle information
- Publish car speed

2.2. Source code analysis

launch file

-base.launch

```
<launch>
     <!-- Start lidar node -->
     <!-- Activate the lidar node -->
     <include file="$(find rplidar_ros)/launch/rplidar.launch"/>
     <!-- Start the car chassis drive node-->
     <!-- Start the car chassis drive node -->
     <include file="$(find yahboomcar_bringup)/launch/yahboomcar.launch"/>
     <!-- Handle control node -->
     <!-- Handle control node -->
     <include file="$(find yahboomcar_ctrl)/launch/yahboom_joy.launch"/>
</launch>
```

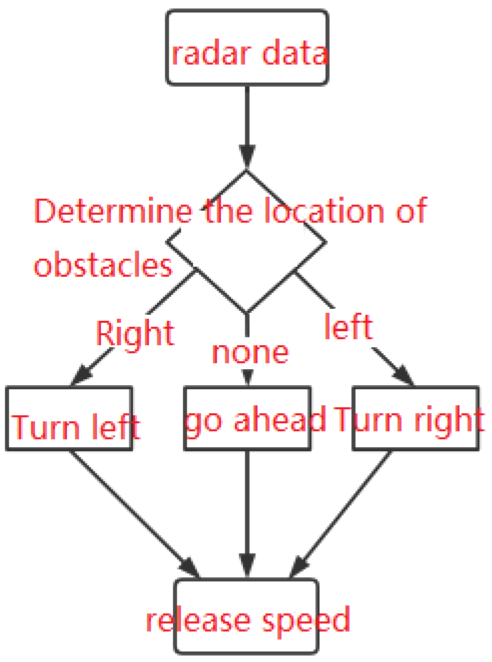
- laser_Avoidance.launch

```
<launch>
     <!-- Launch base.launch file -->
     <!-- Launch the base.launch file -->
     <include file="$(find yahboomcar_laser)/launch/base.launch"/>
     <!-- Start lidar obstacle avoidance node -->
     <!-- Activate lidar obstacle avoidance node -->
     <node name='laser_Avoidance' pkg="yahboomcar_laser"
type="laser_Avoidance.py" required="true" output="screen"/>
</launch>
```

laser_Avoidance.py source code parameter analysis:

Parameters	Default value	Judgment
self.front_warning	Default is 0	When the value is greater than 10, it means there is an obstacle ahead.
self.Left_warning	Default is 0	When the value is greater than 10, it means there is an obstacle in front of the left.
self.Right_warning	Default is 0	When the value is greater than 10, it means there is an obstacle on the right front.

Programming flow chart:



The car moves forward autonomously and avoids surrounding obstacles. For specific program design, please see the source code of [laser_Avoidance.py].