4. Lidar following

Introduction to radar following gameplay:

- Set the laser radar detection angle and distance.
- After turning on the car, the car follows the target closest to the car, and the screen prints tracking obstacles!! and issues an alarm. The car will keep a certain distance.
- When there is an obstacle in front of the car, it will keep moving backwards until there is no obstacle.
- The PID of the car's linear speed and angular speed can be adjusted to achieve the best car following effect.

4.1. Usage

Note: The [SWB] mid-range of the aircraft model remote control has the [emergency stop] function of this gameplay. Please put the aircraft model remote control in a convenient place for control. Pay attention to safety when playing! ! !

- To start control, you need to first turn the SWB button to the upper gear position (control command mode) to release the remote control

Turn off the self-starting chassis service

```
sudo supervisorctl stop ChassisServer
```

One-click start, after executing the command, the car starts to move.

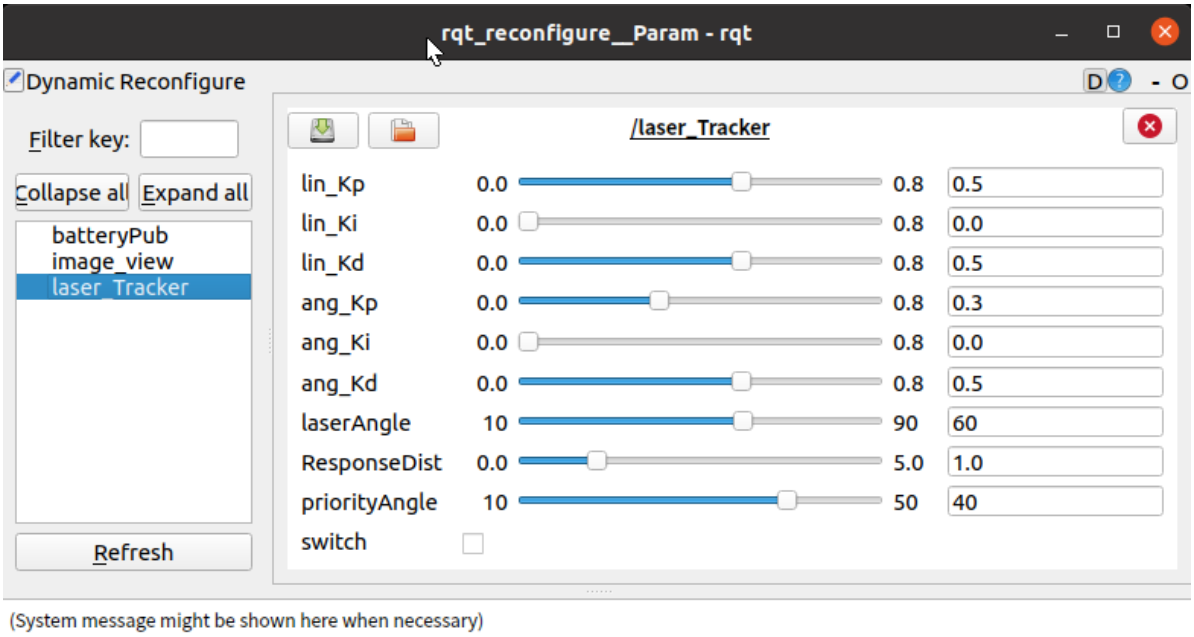
```
sudo supervisorctl restart LaserServer #start/stop Switch radar service (indoor version)
roslaunch yahboom_navrobo_laser laser_Tracker.launch
```

- The car has a backward collision avoidance function, and the radar needs to be started normally. If you run `rostopic echo /scan`

and the print is empty, the data cannot be obtained, and the startup is abnormal. Please restart the radar service command.

Dynamic debugging parameters

```
rosrun rqt_reconfigure rqt_reconfigure
```



Parameter analysis:

Parameter	Range	Analysis
【laserAngle】	【10, 90】	Laser radar detection angle (left and right angles)
【ResponseDist】	【0.0, 5.0】	Car following distance
【priorityAngle】	【10, 50】	Car priority following range (left and right angles)
【switch】	【False, True】	Car starts to pause

【lin_Kp】, 【lin_Ki】, 【lin_Kd】: Car linear speed PID debugging.

[ang_Kp], [ang_Ki], [ang_Kd]: PID debugging of the car's angular velocity.

In the box in front of [switch], click the value of [switch] to True, and the car stops. [switch] defaults to False, and the car moves.

The parameter [priorityAngle] cannot be larger than [laserAngle], otherwise it is meaningless.

- Parameter modification

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

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

```
class laserTracker:
def __init__(self):
rospy.on_shutdown(self.cancel)
... ...
self.lin_pid = SinglePID(0.5, 0.0, 0.5)
self.ang_pid = SinglePID(0.3, 0.0, 0.5)
```

```
self.ResponseDist = rospy.get_param('~targetDist', 1.0)
Server(laserTrackerPIDConfig, self.dynamic_reconfigure_callback)
self.laserAngle = 60
self.priorityAngle = 40 # 40 ``` [rqt_reconfigure] Initial value modification of debugging tool ```python gen.add("lin_Kp", double_t, 0, "Kp in PID", 0.5, 0, 0.8) gen.add("lin_Ki", double_t, 0, "Ki in PID", 0.0, 0, 0.8) gen.add("lin_Kd", double_t, 0 , "Kd in PID", 0.5, 0, 0.8) gen.add("ang_Kp", double_t, 0, "Kp in PID", 0.3, 0, 0.8) gen.add("ang_Ki", double_t, 0, "Ki in PID", 0.0, 0, 0.8) gen.add("ang_Kd", double_t, 0, "Kd in PID", 0.5, 0, 0.8)
gen.add("laserAngle", int_t, 0, "laserAngle", 60, 10, 90)
gen.add("ResponseDist", double_t, 0, "ResponseDst", 1.0, 0, 5)
gen.add("priorityAngle", int_t, 0, "priorityAngle", 40, 10, 50)
gen.add("switch", bool_t, 0, "switch in rosbot", False)
```

Enter the [cfg] folder of the [yahboom_navrobo_laser] function package and modify the initial values of the corresponding parameters in the [laserTrackerPID.cfg] file.

```
gen.add("lin_Kp", double_t, 0, "Kp in PID", 0.5, 0, 0.8)
```

Analyze the above example

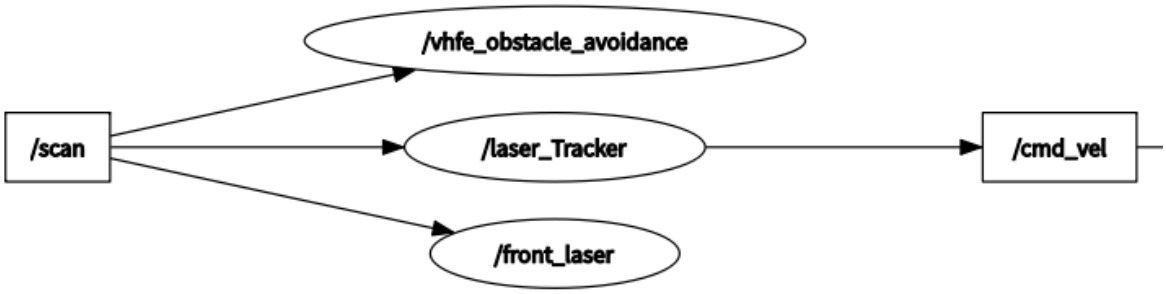
Parameter	Analysis	Corresponding parameter
name	Name of the parameter	"lin_Kp"
type	Parameter data type	double_t
level	A bit mask passed to the callback	0
description	A description parameter	"Kp in PID"
default	Initial value for node startup	0.5
min	Minimum value of parameter	0
max	Maximum value of parameter	0.8

Note: After the modification is completed, the environment must be recompiled and updated to be effective.

```
cd ~/YBAMR-COBOT-EDU-00001/
catkin build yahboom_navrobo_laser
source install/setup.bash
```

Node view

```
rqt_graph
```



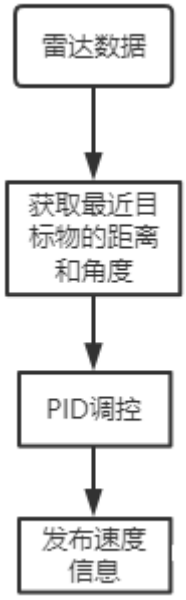
4.2, source code analysis

launch file

- laser_Avoidance.launch

```
<launch>
<!-- Chassis drive -->
<include file="$(find scout_bringup)/launch/scout_mini_robot_bringup.launch"/>
<!-- Start the lidar follow node -->
<!-- Activate lidar follow node -->
<node name='laser_Tracker' pkg="yahboom_navrobo_laser" type="laser_Tracker.py"
required="true" output="screen"/>
</launch>
```

laser_Tracker.py source code flow chart:



According to the position where the target appears, the car moves autonomously to face the target and always maintains a certain distance.