5. Patrol gameplay

5.1. How to use

Note: [R2] on the remote control handle has the [pause/start] function for this gameplay. This section takes the X3 Mai Lun car as an example.

Depending on the environment, the parameters will be different; this function requires patient debugging to achieve good results.

5.1.1. Start

```
#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
```

One-click startup (robot side)

```
roslaunch yahboomcar_bringup patrol.launch
```

Started successfully, print log

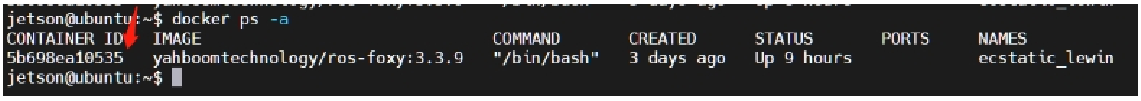
```
# Log of successful patrol function startup
Bring up rqt_reconfigure to control the Robot.
# Log of successful startup of the underlying driver
First IMU message received.
# Log of successful lidar startup
RPLIDAR S/N: 6A97EDF9C7E29BD1A7E39EF2FA44431B
[ INFO] [1631677752.206795121]: Firmware Ver: 1.29
[ INFO] [1631677752.208026726]: Hardware Rev: 7
[ INFO] [1631677752.210976099]: RPLidar health status : 0
[ INFO] [1631677752.808115075]: current scan mode: Sensitivity, max_distance:
12.0 m, Point number: 7.9K, angle_compensate: 2
```

At this time, after the above three parts are started successfully, the dynamic parameter debugging tool is started on the virtual machine side.

<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:

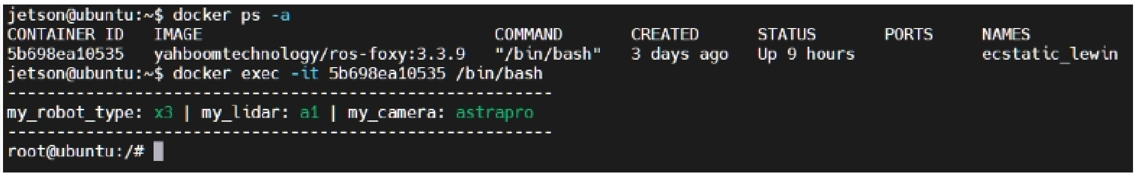
```
docker ps -a
```



```
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:

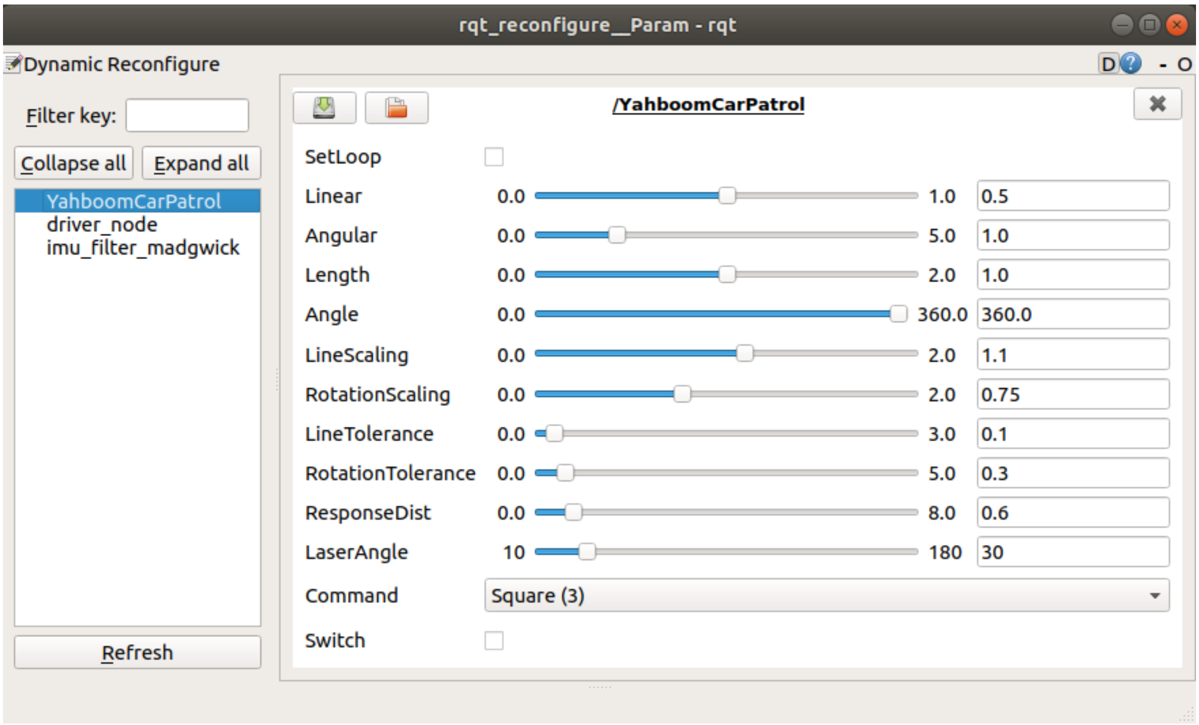
```
docker exec -it 5b698ea10535 /bin/bash
```



```
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.

```
rosrun rqt_reconfigure rqt_reconfigure
```



SetLoop □
Linear 0.0 1.0 0.5
Angular 0.0 5.0 1.0
Length 0.0 2.0 1.0
Angle 0.0 360.0 360.0
LineScaling 0.0 2.0 1.1
RotationScaling 0.0 2.0 0.75
LineTolerance 0.0 3.0 0.1
RotationTolerance 0.0 5.0 0.3
ResponseDist 0.0 8.0 0.6
LaserAngle 10 180 30
Command Square (3)
Switch □

Parameter analysis:

Parameters	Range	Parse
[SetLoop]	[False, True]	Loop patrol, default False.
[Linear]	[0.0, 1.0]	The running linear speed of the car
[Angular]	[0.0, 5.0]	The running angular speed of the car
[Length]	[0.0, 2.0]	The straight-line running distance of the car
[Angle]	[0.0, 360.0]	Rotation angle of the car

Parameters	Range	Parse
[LineScaling]	[0.0, 2.0]	Straight line distance scaling ratio, default 0.9
[RotationScaling]	[0.0, 2.0]	Rotation angle scaling, default 1.0
[LineTolerance]	[0.0, 1.0]	Allowable straight line distance error
[RotationTolerance]	[0.0, 5.0]	Allowed rotation angle error
[ResponseDist]	[0.0, 8.0]	If there are obstacles within the response distance, the car will stop moving; Remove the obstacles and the car will continue to complete the patrol task.
[LaserAngle]	[10, 180]	Lidar detection angle (left and right angles)
[Command]	Default [Square]	Patrol mode: [LengthTest, AngleTest, Triangle, Square, Parallelogram, Circle]
[Switch]	[False, True]	Patrol function [Start/Pause]

When debugging parameters, mainly debug [RotationScaling]. It can be debugged to [1.0, 1.1, 1.2, 1.3, 1.4], etc. according to the actual situation. Observe the effect. The circle is generally 0.9.

1), [LengthTest]: Run the test command straight, adjust the [LineScaling] and [LineTolerance] parameters so that the actual running distance of the car is close to the value [Length].

[LineScaling] The smaller the parameter, the greater the straight distance. The smaller the [LineTolerance] parameter is, the greater the front and rear vibration will be. Just debug multiple times and find the best data. Errors will always exist.

2), [AngleTest]: Rotation test command, adjust the [RotationScaling] and [RotationTolerance] parameters so that the true rotation distance of the car is close to the value [Angle].

[RotationScaling] The smaller the parameter, the larger the rotation angle. The smaller the [RotationTolerance] parameter is, the greater the left and right vibration will be. Just debug multiple times and find the best data. Errors will always exist.

After debugging [1)] and [2)], [LineScaling] and [LineTolerance], [RotationScaling] and [RotationTolerance] generally do not need to be adjusted.

5.1.2. 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_bringup] function package and modify the parameters corresponding to the [patrol.py] file, as shown below

```
class YahboomCarPatrol():
    def __init__(self):
        ... ...
        self.SetLoop = False
        self.Linear = 0.5
        self.Angular = 1.0
        self.Length = 1.0
        self.Angle = 360.0
        self.LineScaling = 1.1
        self.RotationScaling = 0.75
        self.LineTolerance = 0.1
        self.RotationTolerance = 0.3
        self.ResponseDist = 0.6
        self.LaserAngle = 20
        self.Command = "finish"
        self.circle_adjust = rospy.get_param('~circle_adjust', 2.0)
```

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

```
gen.add("SetLoop", bool_t, 0, "SetLoop", False)
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("Length", double_t, 0, "Length in limit", 1.0, 0, 2.0)
gen.add("Angle", double_t, 0, "Angle in limit", 360.0, 0, 360.0)
gen.add("LineScaling", double_t, 0, "Line Scaling", 1.1, 0, 2.0)
gen.add("RotationScaling", double_t, 0, "Rotation Scaling", 0.75, 0, 2.0)
gen.add("LineTolerance", double_t, 0, "LineTolerance", 0.1, 0, 3.0)
gen.add("RotationTolerance", double_t, 0, "RotationTolerance", 0.3, 0, 5.0)
gen.add("ResponseDist", double_t, 0, "ResponseDist in limit", 0.6, 0.0, 8.0)
gen.add("LaserAngle", int_t, 0, "LaserAngle in limit", 30, 10, 180)
```

Enter the [cfg] folder of the [yahboomcar_bringup] function package and modify the initial value of the corresponding parameter in the [PatrolParam.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.

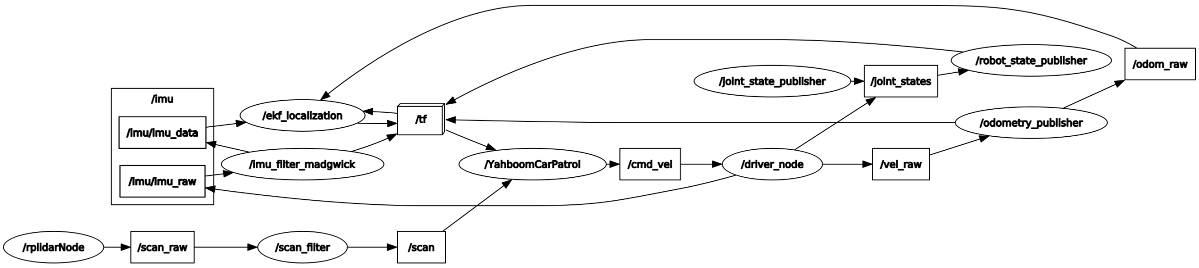
```
cd ~/yahboomcar_ws
catkin_make
source devel/setup.bash
```

5.1.3. Patrol function

- After successful startup, select the trajectory [Triangle, Square, Parallelogram, Circle] to be executed for patrol in [Command].
- The parameter [Length] can be adjusted according to needs. For example, the default value is 1.0, the command is Square, and the trajectory of the car is a square with a side length of 1.0.
- When adjusting the parameter [Linear], please note that the greater the speed, the greater the inertia, and the smaller the accuracy.
- Parameter [LaserAngle]: For example: the angle is 30°. At this time, the system only analyzes 30° on the left and right of the car (0° in front of the car)
- Parameter [Switch]: After setting, click the box behind [Switch] to start patrolling. By default, it is executed once. After the execution is completed, the check mark in the box disappears automatically.
- If you need to patrol in a loop, click the box behind [SetLoop] to continue patrolling, and the error will accumulate to become larger and larger.

Node view

```
rqt_graph
```



5.2. Source code analysis

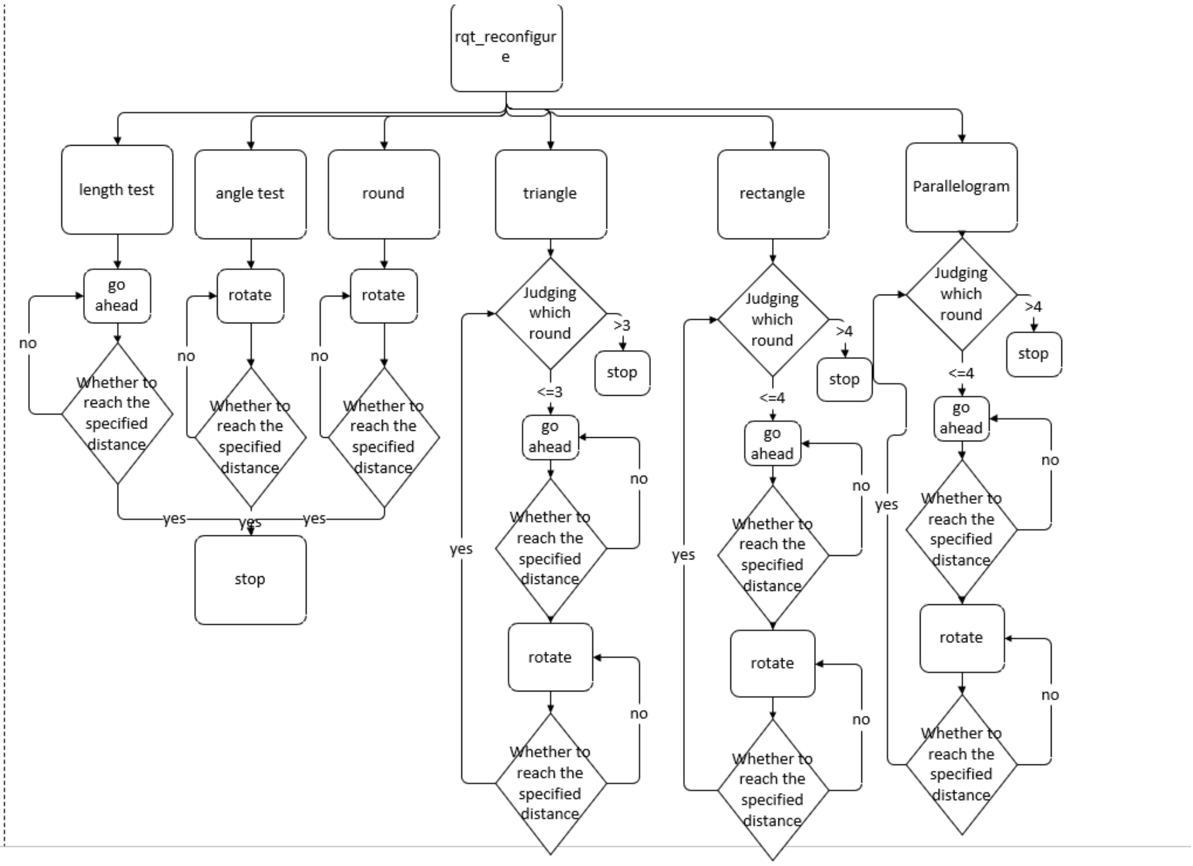
launch file

- patrol.launch

```
<launch>
     <!-- Start the underlying driver -->
     <include file="$(find yahboomcar_bringup)/launch/bringup.launch"/>
     <!-- Start lidar Start lidar -->
     <include file="$(find rplidar_ros)/launch/rplidar.launch"/>
     <!-- Start patrol node Start patrol node -->
     <node pkg="yahboomcar_bringup" type="patrol.py" name="YahboomCarPatrol"
required="true" output="screen">
         <param name="circle_adjust" type="double" value="2.0"/>
     </node>
</launch>
```

- circle_adjust parameter: It is a circle patrol parameter, which is a proportional factor for adjusting the circle radius.

patrol.py source code flow chart:


rqt_reconfigure
length test
angle test
round
triangle
rectangle
Parallelogram
go ahead
rotate
rotate
Judging which round
Judging which round
Judging which round
>3
>4
>4
stop
stop
stop
<=3
<=4
<=4
no
no
no
Whether to reach the specified distance
Whether to reach the specified distance
Whether to reach the specified distance
yes
yes
yes
stop
go ahead
go ahead
go ahead
no
no
no
yes
yes
yes
Whether to reach the specified distance
Whether to reach the specified distance
Whether to reach the specified distance
rotate
rotate
rotate
no
no
no
Whether to reach the specified distance
Whether to reach the specified distance
Whether to reach the specified distance