

5. Patrol gameplay

5. Patrol gameplay

5.1. Usage

5.1.1, Startup

5.1.2, Parameter modification

5.1.3, Patrol function

5.2, source code analysis

5.1. Usage

Note: The [SWB] mid-range of the aircraft remote control has the [emergency stop] function of this gameplay. Please place the aircraft 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

The parameters will be different in different environments; this function requires patient debugging to achieve good results.

5.1.1, Startup

Stop the self-starting chassis service

```
sudo supervisorctl stop ChassisServer
```

One-click start (robot side)

```
sudo supervisorctl restart LaserServer #start/stop Turn on/off radar service  
(indoor version)  
roslaunch yahboom_navrobo_bringup patrol.launch
```

- It is necessary to ensure that the radar starts normally. If you run `rostopic echo /scan`

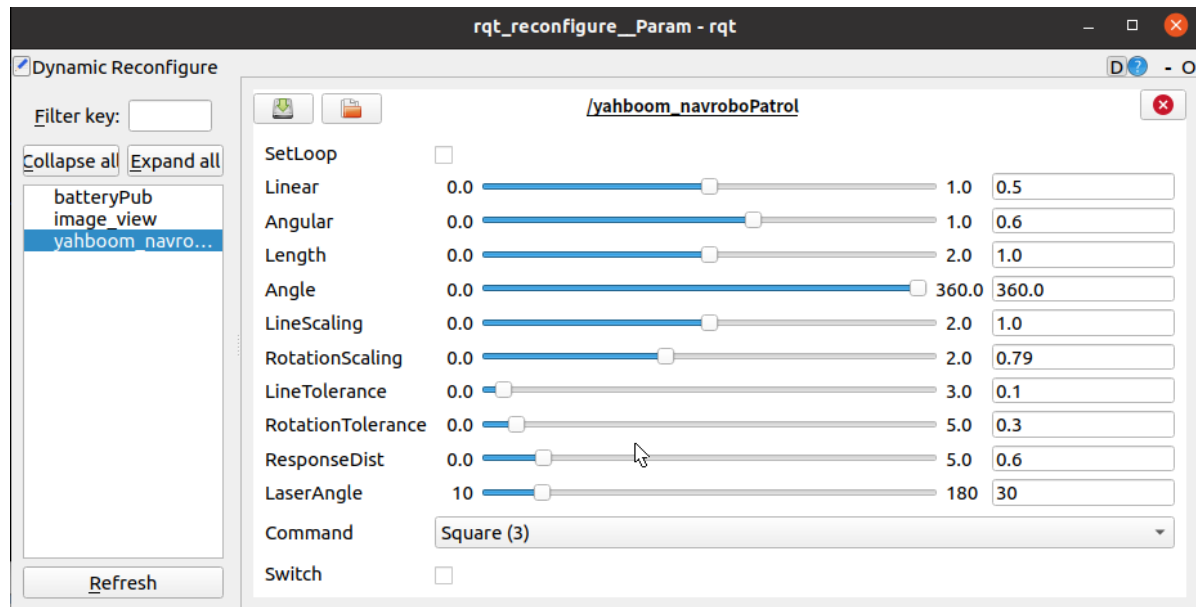
and the print is empty and the data cannot be obtained, the startup is abnormal. Please restart the radar service command.

Started successfully, print log

```
# Log of successful patrol function startup  
Bring up rqt_reconfigure to control the Robot.  
# Log of successful underlying driver startup  
IMU Type: Normal Port:/dev/imu_usb baud:230400  
[INFO] [1719409799.546426]: Serial port opened successfully...  
[ WARN] [1719451665.871742947]: Transform from imu_link to base_footprint was  
unavailable for the time requested. using latest instead.
```

At this point, when the above part is successfully started, start the dynamic parameter debugging tool.

```
roslaunch rqt_reconfigure rqt_reconfigure
```



Parameter analysis:

Parameter	Range	Analysis
【SetLoop】	【False, True】	Loop patrol, default False.
【Linear】	【0.0, 1.0】	Linear speed of the car
【Angular】	【0.0, 1.0】	Angular speed of the car
【Length】	【0.0, 2.0】	Linear distance of the car
【Angle】	【0.0, 360.0】	Rotation angle of the car
【LineScaling】	【0.0, 2.0】	Linear distance scaling ratio, default 0.9
【RotationScaling】	【0.0, 2.0】	Rotation angle scaling ratio, default 1.0
【LineTolerance】	【0.0, 3.0】	Allowable linear distance error
【RotationTolerance】	【0.0, 5.0】	Allowable rotation angle error
【ResponseDist】	【0.0, 5.0】	If there is an obstacle within the response distance, the car stops moving; Remove the obstacle, and the car continues to complete the patrol task.
【LaserAngle】	【10, 180】	Laser radar detection angle (angle on the left and right side)
【Command】	Default 【Square】	Patrol mode: 【LengthTest, AngleTest, Triangle, Square, Parallelogram, Circle】
【Switch】	【False, True】	Patrol function 【Start/Pause】

When debugging parameters, mainly debug 【RotationScaling】 , which can be debugged to [1.0, 1.1, 1.2, 1.3, 1.4] according to actual conditions, observe the effect, and the circle is generally 0.9.

1), 【LengthTest】 : Straight test command, adjust the 【LineScaling】 and 【LineTolerance】 parameters to make the actual running distance of the car close to the value 【Length】 .

The smaller the 【LineScaling】 parameter, the greater the straight distance. The smaller the 【LineTolerance】 parameter, the greater the front and rear vibration. After multiple debugging, you can get the best data. Errors always exist.

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

The smaller the [RotationScaling] parameter, the larger the rotation angle. The smaller the [RotationTolerance] parameter, the greater the left and right vibration. After multiple debugging, you can get the best data. Errors always exist.

After [1]) and [2]) debugging, [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, 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_bringup] function package and modify the corresponding parameters of 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] Initial value modification of debugging tool ```python
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, "Line Tolerance", 0.1, 0, 3.0)
gen.add("RotationTolerance", double_t, 0, "Rotation Tolerance", 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 [yahboom_navrobo_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)
```

Analyze the above example

Parameter	Analysis	Corresponding parameter
name	Name of the parameter	"Linear"
type	Parameter data type	double_t
level	A bit mask passed to the callback	0
description	A description parameter	"Linear in robot"
default	Initial value of the node startup	0.5
min	Minimum value of the parameter	0
max	Maximum value of the parameter	1.0

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

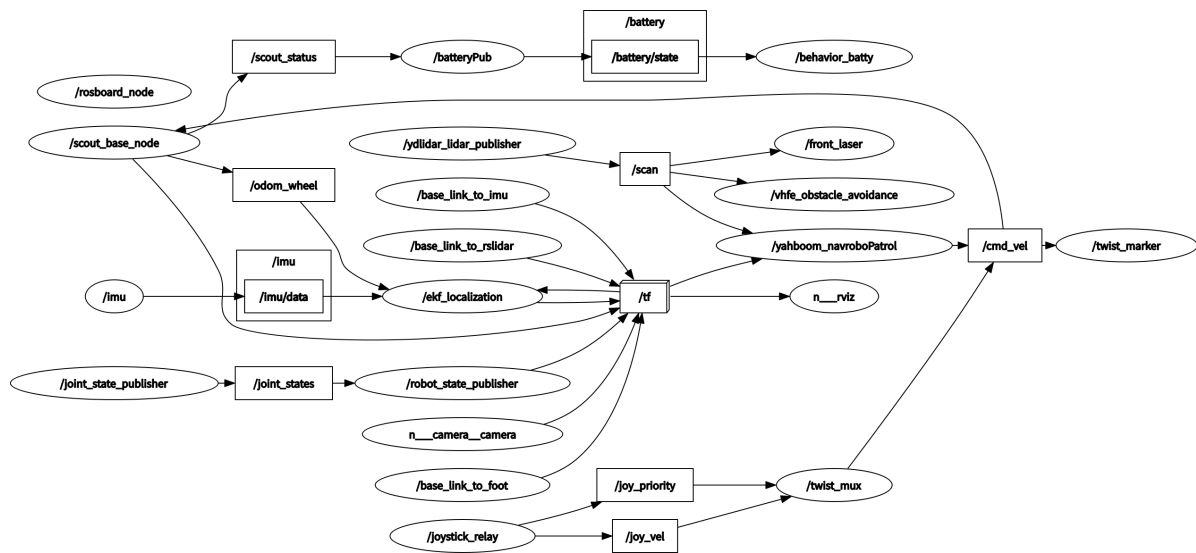
```
cd ~/yahboomcar_ws
catkin_make
source devel/setup.bash
```

5.1.3, Patrol function

- After successful startup, select the trajectory to be executed for patrol in [Command] [Triangle, Square, Parallelogram, Circle].
- 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 and the greater the inertia, the lower 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° directly in front)
- Parameter [Switch]: After setting, click the box behind [Switch] to start patrolling. It is executed once by default. After execution, the check mark in the box disappears automatically.
- If you need to patrol in a loop, click the box behind [SetLoop]. If you patrol continuously, the error will accumulate and become larger and larger.

Node view

```
rqt_graph
```



5.2, source code analysis

launch file

- patrol.launch

```
<launch>
  <!-- Chassis driver -->
  <include file="$(find
scout_bringup)/launch/scout_mini_robot_brpatrol.launch"/>
  <arg name="lidar_type" value="$(env RPLIDAR_TYPE)" doc="lidar_type
[4ROS,a1]"/>
  <!-- Start patrol node -->
  <node pkg="yahboom_navrobo_bringup" type="patrol_4ROS.py"
name="yahboom_navroboPatrol" required="true" output="screen" if="$(eval
arg('lidar_type') == '4ROS')">
    <param name="circle_adjust" type="double" value="2.0"/>
  </node>
  <node pkg="yahboom_navrobo_bringup" type="patrol_a1.py"
name="yahboom_navroboPatrol" required="true" output="screen" if="$(eval
arg('lidar_type') == 'a1')">
    <param name="circle_adjust" type="double" value="2.0"/>
  </node>
</launch>
```

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

patrol.py source code flow chart:

