5、Patrol game

Function package：~/transbot_ws/src/transbot_bringup

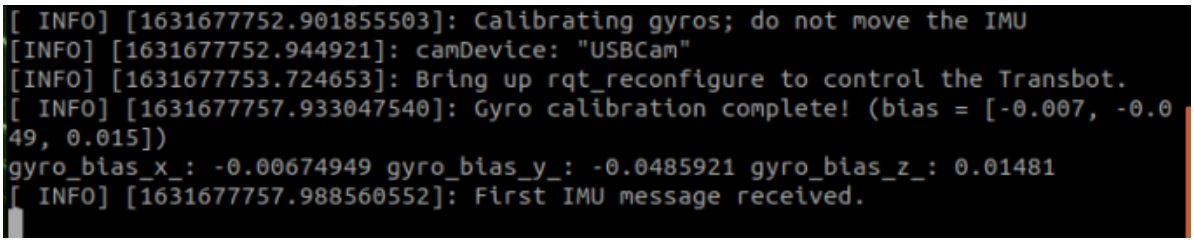
5.1、Instructions

5.1.1、Start up

Start up（Raspberry Pi as a example）

PS: The startup steps here are not the same as the one-key startup in the video, it needs to be started in steps.

```
roslaunch   transbot_bringup     bringup.launch #Turn on the car
roslaunch   rplidar_ros   rplidar.launch        #Turn on the lidar
rosrun   transbot_bringup    transbot_patrol.py #Turn on patrol
```

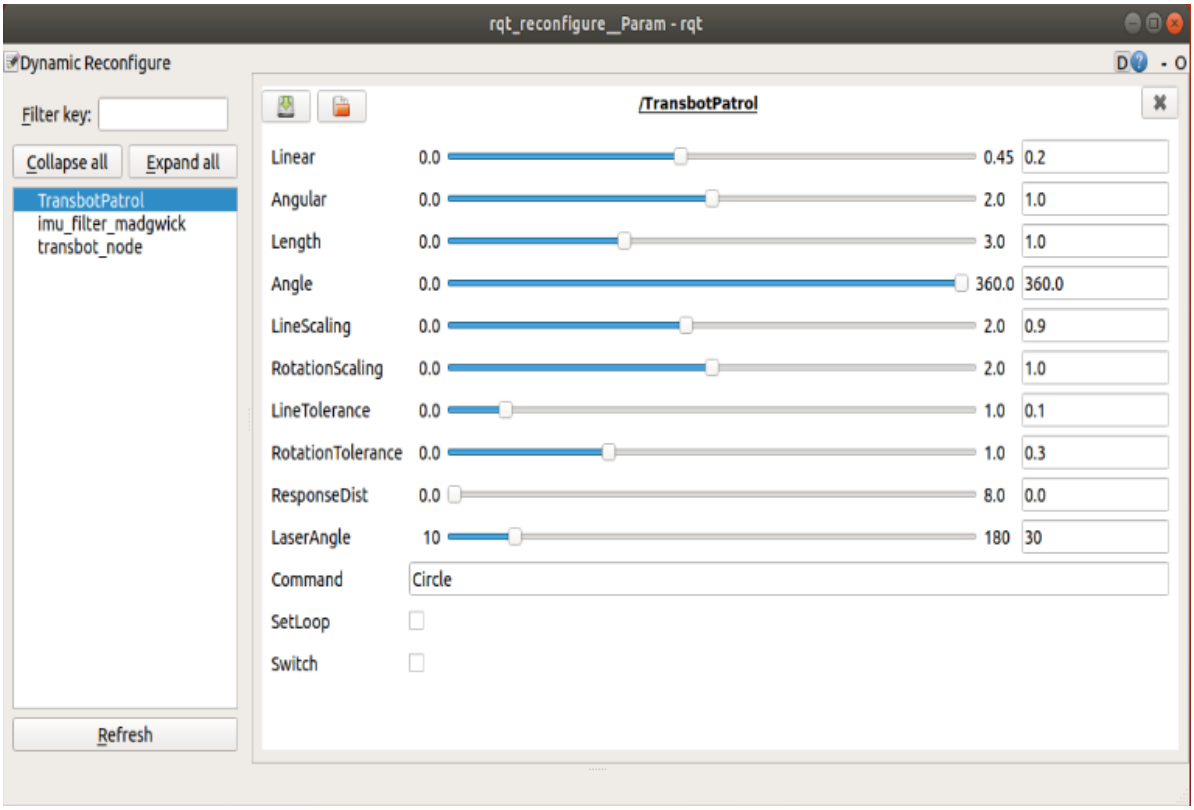


Start successfully, print log

```
# Log of successful patrol function activation
Bring up rqt_reconfigure to control the Transbot.
# Log of the 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 point, when the above three parts are successfully started, we can start the dynamic parameter debugging tool on the virtual machine side.

```
rosrun rqt_reconfigure rqt_reconfigure
```



Parameter analysis:

Parameter	Range	Analysis
【Linear】	【0.0, 0.45】	Linear speed of the car
【Angular】	【0.0, 2.0】	Angular speed of the car
【Length】	【0.0, 3.0】	The straight running distance of the car
【Angle】	【0.0, 360.0】	The rotation angle of the CAR
【LineScaling】	【0.0, 2.0】	Straight-line distance scaling ratio, default 0.9
【RotationScaling】	【0.0, 2.0】	Rotation angle zoom ratio, default 1.0
【LineTolerance】	【0.0, 1.0】	Allowable linear distance error
【RotationTolerance】	【0.0, 1.0】	Allowable rotation angle error
【ResponseDist】	【0.4, 8.0】	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】	Lidar detection angle (angle of left and right side)
【Command】	Default 【Square】	Patrol method:【LengthTest, AngleTest,Triangle, Square, Parallelogram, Circle】
【SetLoop】	【False, True】	Whether to patrol in a loop, the default is False
【Switch】	【False, True】	Patrol function [start/pause]

5.1.2、Parameter test

When using the patrol function, you can skip steps [1)] and [2)] and run other commands directly. If the accuracy is particularly poor, re-debug it.

After debugging, you need to modify the parameters to the corresponding parameters in the [PatrolParam.cfg] file.

After modification, the default parameters will automatically become the modified values. (You need to recompile in the workspace and update the environment variables to take effect)

```
cd ~/transbot_ws/        # Enter work space
catkin_make              # Compile
source devel/setup.bash  # Update environment
```

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

【LineScaling】 The smaller the parameter, the greater the straight distance. 【LineTolerance】 The smaller the parameter, the greater the front and back vibration. Debug multiple times, choose the best data.

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

【RotationScaling】 The smaller the parameter, the larger the rotation angle. 【RotationTolerance】The smaller the parameter, the greater the left and right vibration. Debug multiple times, choose the best data.

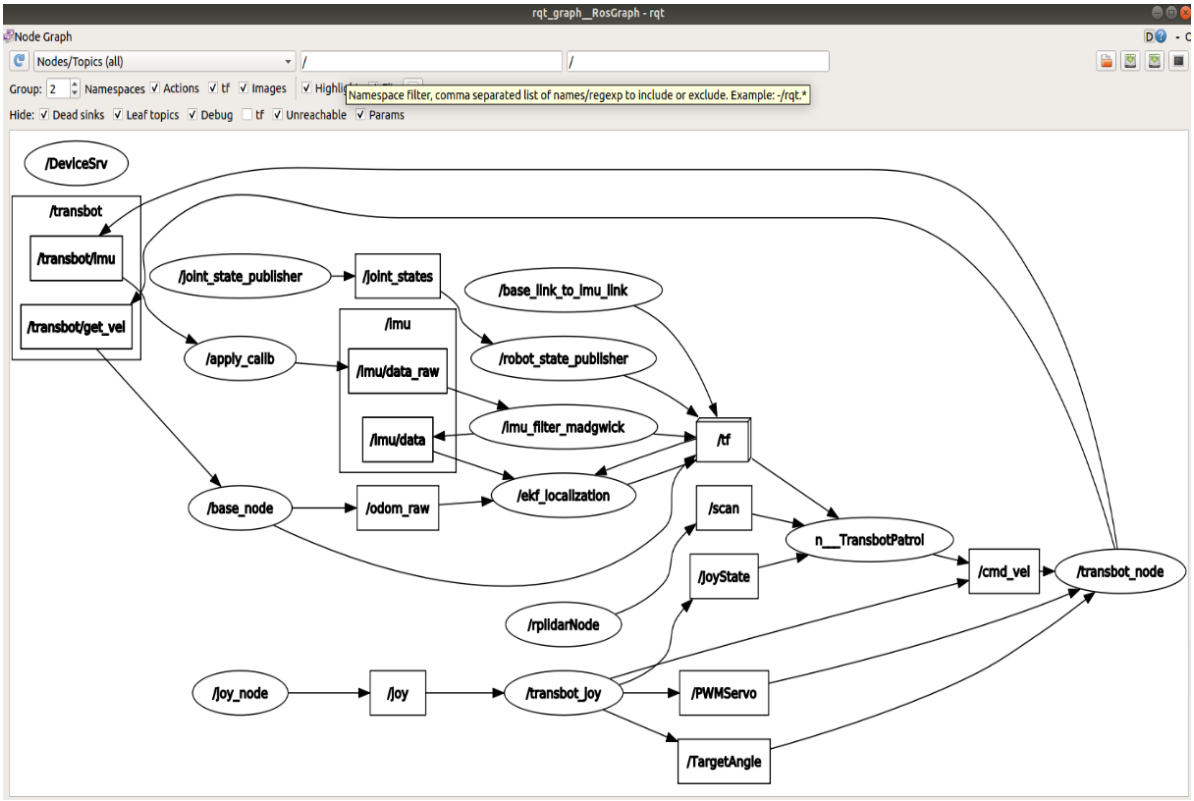
After step-1 and step-3 debugging, 【LineScaling】 and 【LineTolerance】, 【RotationScaling] and 【RotationTolerance】 generally do not need to be adjusted.

5.1.3、Patrol

- After the startup is successful, enter the trajectory [Triangle, Square, Parallelogram, Circle] to be executed in the [Command].
- The parameter [Length] can be adjusted according to requirements. For example, the default value is 1.0, the command is Square, and the track of the car is a square with a side length of 1.0.
- About the parameter [Linear], the greater the speed, the greater the inertia, and the smaller the accuracy.
- About the parameter [LaserAngle]: For example: the angle is 30°, at this time the system only analyzes the car with 30° left and right (0° in front)
- About the [Switch]: After setting, click the box behind [Switch] to start patrol. It is executed once by default. After the execution is completed, the check mark in the box automatically disappears.
- If you need to patrol in a loop, click the box behind [SetLoop] to patrol continuously, and the error will accumulate more and more.

View node

```
rqt_graph
```



5.2、Code analysis

launch file

- transbot_patrol.launch

```
<launch>
    <!-- Start the underlying driver-->
    <include file="$(find transbot_bringup)/launch/bringup.launch"/>
    <!-- Start Lidar -->
    <include file="$(find rplidar_ros)/launch/rplidar.launch"/>
    <!-- Start patrol node -->
    <node pkg="transbot_bringup" type="transbot_patrol.py" name="TransbotPatrol"
required="true" output="screen"/>
</launch>
```

py code path：~/transbot_ws/src/transbot_bringup/scripts/transbot_patrol.py

Note: When the patrol command is [Circle], the parameter [RotationScaling] has been fixed and can only be modified from the source code.

```
        while not rospy.is_shutdown():
            if self.Switch==True:
                # Triangle patrol
                if self.Command == "Triangle": self.Triangle(index, 135)
                # Line test command
                elif self.Command == "LengthTest":
                    position = self.get_position()
                    advancing = self.advancing(position.x, position.y,
self.Length)
                    if advancing == True: self.Command = "finish"
                # Rotation test command
                elif self.Command == "AngleTest":
                    spin = self.Spin(self.Angle)
```

```
                    if spin == True: self.Command = "finish"
                # Square patrol
                elif self.Command == "Square": self.Square(index, 90)
                # Parallelogram patrol
                elif self.Command == "Parallelogram": self.Parallelogram(index,
120)
                # Circular patrol
                elif self.Command == "Circle":
                    # The parameter [RotationScaling is fixed
                    self.RotationScaling = 0.9
                    spin = self.Spin(360)
                    if spin == True: self.Command = "finish"
                # After execution, stop the movement
                if self.Command == "finish":
                    self.pub_cmdVel.publish(Twist())
                    # If it is not a loop, the instruction is terminated;
otherwise, the instruction continues.
                    if self.SetLoop == False:
                        params = {'Switch': False}
                        self.dyn_client.update_configuration(params)
                    else: self.Command = self.command_src
```