

5、巡逻玩法

5、巡逻玩法

5.1、使用方法

5.1.1、启动

5.1.2、参数测试

5.1.3、巡逻功能

5.2、源码解析

功能包路径：~/rplidar_ws/src/transbot_bringup

5.1、使用方法

注意：遥控手柄的【R2】具备所有玩法的【暂停/开启】的功能。

环境不同，参数会有不同；该功能需要耐心调试，才能跑出好的效果。

5.1.1、启动

一键启动 (jetson nano)

```
roslaunch transbot_bringup transbot_patrol.launch lidar_type:=a1
```

lidar_type参数：使用激光雷达的型号：[a1,a2,a3,s1,s2]。

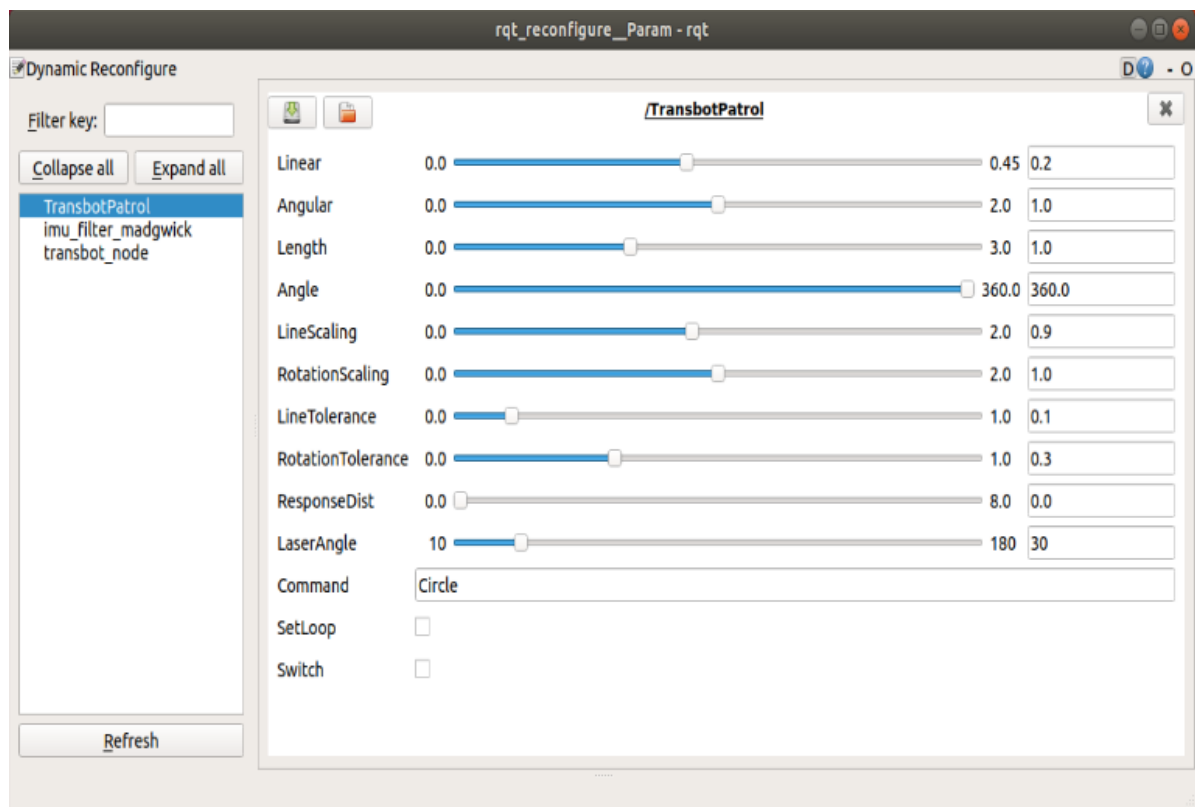
```
[ INFO] [1631677752.901855503]: Calibrating gyros; do not move the IMU
[INFO] [1631677752.944921]: camDevice: "USBCam"
[INFO] [1631677753.724653]: Bring up rqt_reconfigure to control the Transbot.
[ INFO] [1631677757.933047540]: Gyro calibration complete! (bias = [-0.007, -0.049, 0.015])
gyro_bias_x_: -0.00674949 gyro_bias_y_: -0.0485921 gyro_bias_z_: 0.01481
[ INFO] [1631677757.988560552]: First IMU message received.
```

启动成功，打印日志

```
# 巡逻功能启动成功的日志
Bring up rqt_reconfigure to control the Transbot.
# 底层驱动启动成功的日志
First IMU message received.
# 激光雷达启动成功的日志
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
```

此时，当上述三部分启动成功后，在虚拟机端启动动态参数调试工具。

```
roslaunch rqt_reconfigure rqt_reconfigure
```



参数解析：

参数	范围	解析
【Linear】	【0.0, 0.45】	小车的运行线速度
【Angular】	【0.0, 2.0】	小车的运行角速度
【Length】	【0.0, 3.0】	小车的直线运行距离
【Angle】	【0.0, 360.0】	小车的旋转角度
【LineScaling】	【0.0, 2.0】	直线距离缩放比例，默认0.9
【RotationScaling】	【0.0, 2.0】	旋转角度缩放比例，默认1.0
【LineTolerance】	【0.0, 1.0】	允许的直线距离误差
【RotationTolerance】	【0.0, 1.0】	允许的旋转角度误差
【ResponseDist】	【0.4, 8.0】	在响应距离内有障碍物，小车停止运动；移走障碍物，小车继续完成巡逻任务。
【LaserAngle】	【10, 180】	激光雷达检测角度（左右一侧角度）
【Command】	默认 【Square】	巡逻方式：【LengthTest, AngleTest, Triangle, Square, Parallelogram, Circle】
【SetLoop】	【False, True】	是否循环巡逻，默认False
【Switch】	【False, True】	巡逻功能【开始/暂停】

参数调试时，主要调试【RotationScaling】，可根据实际情况调试为[1.0, 1.1, 1.2, 1.3, 1.4]等，观察效果，圆形一般是0.9。

5.1.2、参数测试

使用巡逻功能时，可跳过【1）】，【2）】，直接运行其他指令，如果精度特别差，再重新调试，调试完毕后记得将参数修改到【PatrolParam.cfg】文件对应参数，修改后，默认参数自动成为修改值。（要在工作空间下重新编译，更新环境变量才可生效）

```
cd ~/rplidar_ws/      # 进入工作空间
catkin_make           # 编译
source devel/setup.bash # 更新环境变量
```

1）、【LengthTest】：直行测试命令，调节【LineScaling】和【LineTolerance】参数，使得小车真实运行距离接近数值【Length】。

【LineScaling】参数越小，直行距离越大。【LineTolerance】参数越小，前后震动越大。多次调试，去最佳数据即可，误差总是存在的。

2)、【AngleTest】：旋转测试命令，调节【RotationScaling】和【RotationTolerance】参数，使得小车真实旋转距离接近数值【Angle】。

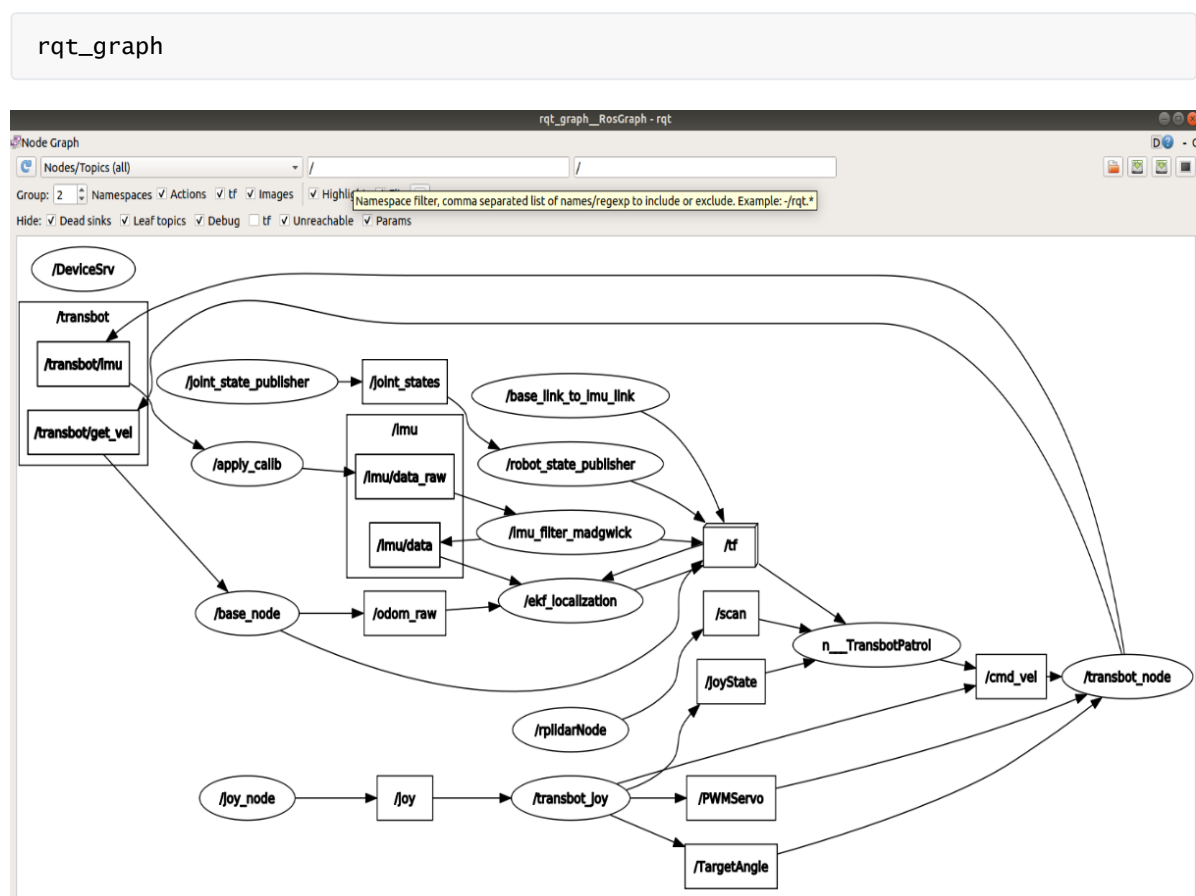
【RotationScaling】参数越小，旋转角度越大。【RotationTolerance】参数越小，左右震动越大。多次调试，去最佳数据即可，误差总是存在的。

经过【1)】，【2)】调试后，【LineScaling】和【LineTolerance】，【RotationScaling】和【RotationTolerance】，一般不需要再调整。

5.1.3、巡逻功能

- 启动成功后，在【Command】输入巡逻需要执行的轨迹【Triangle, Square, Parallelogram, Circle】。
- 参数【Length】可根据需求调整，例如，默认值是1.0，指令是Square，小车的轨迹便是边长是1.0的正方形。
- 调节参数【Linear】需注意，速度越大，惯性越大，精度就越小。
- 参数【LaserAngle】：例如：角度为30°，此时系统只分析小车左右各30°（正前方为0°）
- 参数【Switch】：设置完毕后，点击【Switch】后面的方框，开始巡逻。默认执行一次，执行完毕后，方框的对号，自动消失。
- 如果需要循环巡逻，点击【SetLoop】后面的方框，连续巡逻，误差会累计越来越大。

节点查看



5.2、源码解析

launch文件

- transbot_patrol.launch

```
<launch>
  <!-- 启动激光雷达-->
  <arg name="lidar_type" default="a1" doc="lidar_type type [a1,a2,a3,s1,s2]"/>
  <include file="$(find rplidar_ros)/launch/rplidar.launch">
    <arg name="lidar_type" value="$(arg lidar_type)"/>
  </include>
  <!-- 启动底层驱动 -->
  <include file="$(find transbot_bringup)/launch/bringup.launch"/>
  <!-- 启动巡逻节点 -->
  <node pkg="transbot_bringup" type="transbot_patrol.py" name="TransbotPatrol"
    required="true" output="screen"/>
</launch>
```

py源码: ~/rplidar_ws/src/transbot_bringup/scripts/transbot_patrol.py

注意: 巡逻指令为【Circle】时, 参数【RotationScaling】。可以在源码中写死, 只能从源码中修改; 也可以注释掉, 一般是0.9。

```
while not rospy.is_shutdown():
    if self.Switch==True:
        # 三角形巡逻
        if self.Command == "Triangle": self.Triangle(index, 135)
        # 直线测试指令
        elif self.Command == "LengthTest":
            position = self.get_position()
            advancing = self.advancing(position.x, position.y,
self.Length)

            if advancing == True: self.Command = "finish"
        # 旋转测试指令
        elif self.Command == "AngleTest":
            spin = self.Spin(self.Angle)
            if spin == True: self.Command = "finish"
        # 正方形巡逻
        elif self.Command == "Square": self.Square(index, 90)
        # 平行四边形巡逻
        elif self.Command == "Parallelogram": self.Parallelogram(index,
120)

        # 圆形巡逻
        elif self.Command == "Circle":
            # 参数【RotationScaling】固定
            self.RotationScaling = 0.9
            spin = self.Spin(360)
            if spin == True: self.Command = "finish"
        # 执行完毕, 停止运动
        if self.Command == "finish":
            self.pub_cmdVel.publish(Twist())
            # 如果不是循环执行, 就结束指令, 否则, 指令继续。
            if self.SetLoop == False:
                params = {'Switch': False}
                self.dyn_client.update_configuration(params)
            else: self.Command = self.command_src
```

