

Mobile Robot Navigation and Control, Lab3

Author: Ying Yiwen Weiyu Chen
Number: 12210159 12210460

Abstract

In this lab, we learn how to operate the laser distance sensor that comes with TurtleBot3.

1 Introduction

We use Turtlebot3 and HLS-LFCD2 LiDAR sensor to control the robot, get the lidar message. First set up the environment, to enable us to control the robot on our computer. Take one computer as master, another computer and turtlebot as host. Then we can control the robot by ros package. Also, we can write ros package ourselves to deal with the scan message.

2 Setting up TurtleBot 3

Using the command "roslaunch turtlebot3_bringup turtlebot3_robot.launch" on turtlebot, "roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch" on my computer, I can control the robot.

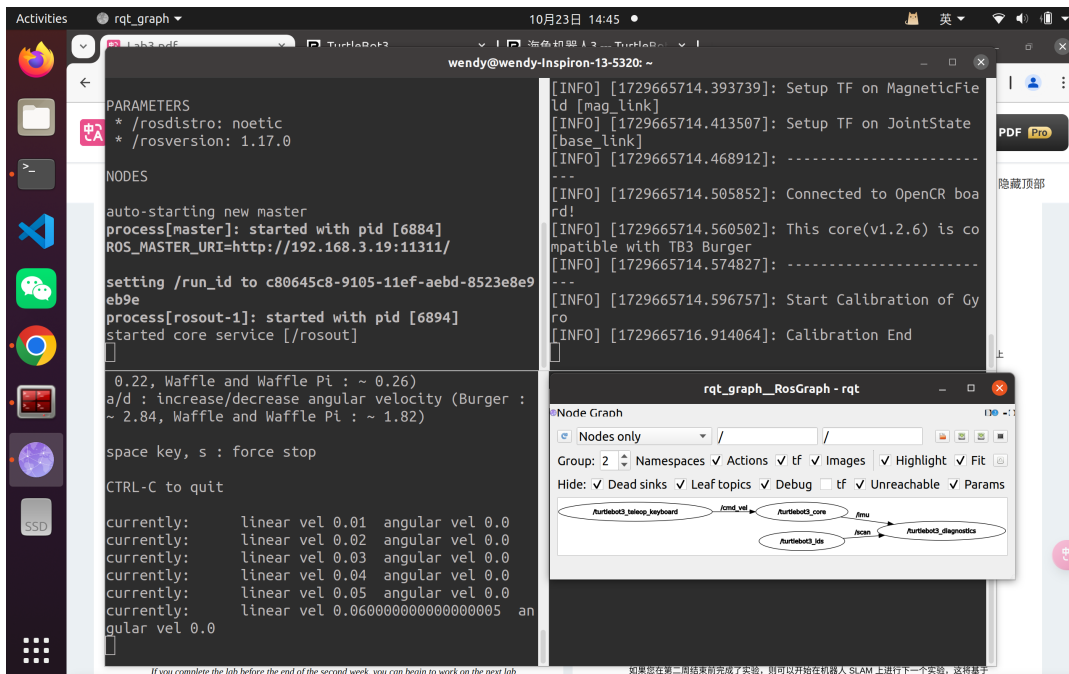
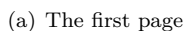


Fig. 1: rqt graph

The message `/scan` tells the information in the lidar. Using "rostopic echo `/scan`":



(b) The second page

Fig. 2: The information in scan

Using the rviz, we can observe the pointcloud more clearly.

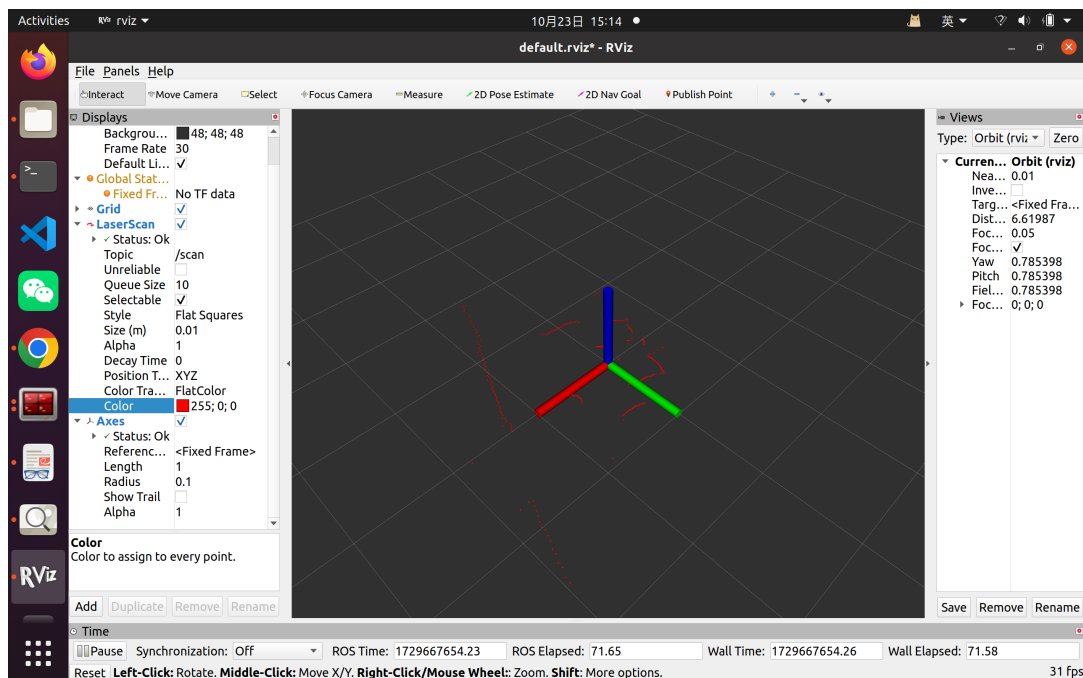


Fig. 3: rviz result

4 Verify the Performance of LiDAR

We put the robot beside a wall, when there is nothing else in other direction, the minimum distance (except 0) is the normal distance towards the wall. As the wall is a plane, we can easily get the true distance. According to the manual, the distance is from the center of LiDAR to the obstacle. Like this:



Fig. 4: Experiment Scenario

```
1 def callback(scan):
2     global dis
3     dis_min = scan.range_max
4     for i in range(0, 360):
5         distance = scan.ranges[i]
6         if distance < dis_min and distance != 0:
7             dis_min = distance
8
9     # the minimum distance represent the measured result normal to the wall
10    dis.append(dis_min)
```

```

1 def process_data(_):
2     global dis
3     if len(dis) > 0:
4         dis_array = np.array(dis)
5         rospy.loginfo("accuracy: %f m", abs(dis_array.mean() - dis_true))
6         rospy.loginfo("precision: %f m", (dis_array.max() - dis_array.min()) / 2)
7         dis.clear()

```

We can also read min and max in the scan, as follow:

```

1 # range_min and range_max
2 scan = rospy.wait_for_message('/scan', LaserScan)
3 rospy.loginfo("range min: %f m, max: %f m", scan.range_min, scan.range_max)

```

When the true distance is 0.464m.

The result is:

range minimum: 0.120000m, range maximum: 3.500000m

accuracy: 0.002667m, precision: 0.0005m

```

wendy@wendy-Inspiron-13-5320: ~/catkin_ws
wendy@wendy-Inspiron-13-5320:~/catkin_ws$ roslaunch lab3 lab3.py
[INFO] [1730202786.020146]: range min: 0.120000 m, max: 3.500000 m
[INFO] [1730202789.028242]: accuracy: 0.002562 m
[INFO] [1730202789.032342]: precision: 0.000500 m
[INFO] [1730202792.024914]: accuracy: 0.002667 m
[INFO] [1730202792.027863]: precision: 0.000500 m
[INFO] [1730202795.024821]: accuracy: 0.002800 m
[INFO] [1730202795.028472]: precision: 0.000500 m
[INFO] [1730202798.027364]: accuracy: 0.002867 m
[INFO] [1730202798.029977]: precision: 0.000500 m
[INFO] [1730202801.024815]: accuracy: 0.002786 m
[INFO] [1730202801.027927]: precision: 0.000500 m
[INFO] [1730202804.028021]: accuracy: 0.002533 m
[INFO] [1730202804.031744]: precision: 0.000500 m
[INFO] [1730202807.027890]: accuracy: 0.002750 m
[INFO] [1730202807.031978]: precision: 0.000500 m
[INFO] [1730202810.025051]: accuracy: 0.002714 m
[INFO] [1730202810.029079]: precision: 0.000500 m
[INFO] [1730202813.024902]: accuracy: 0.002533 m
[INFO] [1730202813.027994]: precision: 0.000500 m
[INFO] [1730202816.027908]: accuracy: 0.002600 m
[INFO] [1730202816.032300]: precision: 0.000500 m
[INFO] [1730202819.027851]: accuracy: 0.002667 m
[INFO] [1730202819.031629]: precision: 0.000500 m
[INFO] [1730202822.027580]: accuracy: 0.002667 m
[INFO] [1730202822.031076]: precision: 0.000500 m
[INFO] [1730202825.027942]: accuracy: 0.002733 m
[INFO] [1730202825.031963]: precision: 0.000500 m
[INFO] [1730202828.025041]: accuracy: 0.002733 m
[INFO] [1730202828.028882]: precision: 0.000500 m
[INFO] [1730202831.027620]: accuracy: 0.002667 m
[INFO] [1730202831.030532]: precision: 0.000500 m
[INFO] [1730202834.025857]: accuracy: 0.002800 m
[INFO] [1730202834.028924]: precision: 0.000500 m
[INFO] [1730202837.025070]: accuracy: 0.002600 m
[INFO] [1730202837.029277]: precision: 0.000500 m

```

(a) My result

Light source	Semiconductor Laser Diode($\lambda=785\text{nm}$)
LASER safety	IEC60825-1 Class 1
Current consumption	400mA or less (Rush current 1A)
Detection distance	120mm ~ 3,500mm
Interface	3.3V USART (230,400 bps) 42bytes per 6 degrees, Full Duplex option
Ambient Light Resistance	10,000 lux or less
Sampling Rate	1.8kHz
Dimensions	69.5(W) X 95.5(D) X 39.5(H)mm
Mass	Under 125g

Measurement Performance Specifications

Items	Specifications
Distance Range	120 ~ 3,500mm
Distance Accuracy (120mm ~ 499mm)	$\pm 15\text{mm}$
Distance Accuracy(500mm ~ 3,500mm)	$\pm 5.0\%$
Distance Precision(120mm ~ 499mm)	$\pm 10\text{mm}$
Distance Precision(500mm ~ 3,500mm)	$\pm 3.5\%$
Scan Rate	300 \pm 10 rpm
Angular Range	360°

(b) Manual

Fig. 5: The information in scan

Our experimental results are in accordance with the instructions.

The range is the same as the manual, 120mm-3500mm, and the tolerances are within the limits specified in the manual(120mm-499mm, $\pm 15\text{mm}$).

5 Conclusion

We successfully demonstrated that the HLS-LFCD2 LiDAR sensor operates effectively for distance measurement in a mobile robot navigation context. The findings are consistent with the manual specifications, confirming the sensor's suitability for navigation tasks.