ELEE 4200

Autonomous Mobile Robots

Homework/Assignment No:8

Find the Door

Author :
Name Wang Jingya
T.No.: 02157119

Instructor:
Dr. Mohan Krishnan

November 19, 2019

Grading Rubric - Information for Students

Grading rubric that will broadly apply to assessing performance. That is, assessment exercises that are associated with a predominantly qualitative rather than quantitative character. If, for a particular exercise, there is substantial deviation from the scheme outlined below, I will let you know! Note: Be aware that if your "Quality of Presentation" is poor, it could impact scores assigned for the other two categories! (DO NOT EDIT THIS PAGE)

		Level of accomplishment					
		0	**1**	**2**	**3**	**4**	**5**
Aspect of effort	**Extent of Completion of Technical Requirements**	Achieved none of the objectives or did not submit	Achieved very few of the objectives	Achieved some of the objectives	Achieved most of the objectives	Achieved almost all the objectives	Demonstrated additional initiative beyond what was required.
	Quality of Analysis & Conclusions	Inadequate	Poor	Below average	Average	Good	Insightful!
	Quality of Presentation	Inadequate	Poor	Below average	Average	Good	Exceptional!

Figure 1: Level of accomplishments

- **Student's overall Level/Score (To be entered by Professor/GTA):**.

Contents

List of Figures

List of Tables

1 Abstract

Our goal is to make the robot find the largest door among the four sides walls doors, and control the robot going out of the wall avoiding hitting on the obstacles.

2 Introduction

- The following are the requirements:
  1.Setting the initial position as following:
  position: x: 2.4683, y: -4.5436 ,z: 0 ,
  orientation: x: -0.0029, y: -0.0028, z: -0.7175, w: 0.6965 ,
  twist: linear: x: 0 , y: 0 ,z: 0 , angular: x: 0.0 , y: 0 , z: 0

  2. Find the widest door among the four sides wall doors.

  3. Going out of the enclosed area through the widest door and avoiding to hit on the obstacles.

  4. Plotting the robot running path.

  5. Plotting the map about the enclosed area

3 Methodology

3.1 Task A:Using the Hokuyo Lida

1.The whole idea:

1. Making the robot run to the center of the enclosed obstacles;
2. Control the robot rotating a whole circle. As it rotating, recording the obstacles coordinates and plotting the map. Additionally, find the widest door among the four doors.
3. According to the widest door, make sure the direction that the robot should run to. Then, make the robot go through the widest door, and using the prior avoiding obstacles code to make the robot go through the target door without hitting the wall.

2.Each section of the code:

3.2.1 Running to the center point:

Because the Hokuyo LIDAR has a longer detecting range, so that I choose the method to compare the distances between the robot and the each side of the wall. And if each distance has almost similar value, I defined it as the center point. So, control the robot to that point.

The specific solution is that:

1.I divide the range into three areas and then calculate the mean value of these three ranges. Then let each two among the three mean values do subtraction calculation. Using the errors to judge. If the three errors are almost equal, then define the situation as the robot is in the center point. Otherwise, adjust the robot moving to make it run to the center point.

Paying attention:

There is one thing that we should pay attention is that Some of the feedback value are infinite value, so that during the calculation, before adding the value to the sum, I firstly check whether the value is infinite. If it is, don't add that value. Additionally, as adding, record that how many numbers have been added. This is because that some Infinite value didn't conclude in the sum, so that if it divided the whole index number directly, the result will smaller than the real value. The specific code is shown like Figure 2

2. Using each two values to do the differences calculation, in order that I can use these values to order the robot moving. The code is like following Figure 3:

3. Using the values to control the robot to the center point.

First, judge the differences between the left side and the right side. If the left side average value is larger than the right side value, then make the robot turn to the left. Otherwise, turn to the opposite.

Secondly, If there is no huge difference between the left side mean value and the right side mean value. Then, comparing the front side mean value with the right side and the left side. For example, if the right side

```
% Calculate the mean value from 1 to 270
% During calculation, we should pay attention to the Infinite values,
% So, before adding, I judge whether the value is Infinite. If it is,
% don't add that value. Additionally, as adding, recording how many
% numbers have been added.

  for i1 = 1:270
      if distance(i1)~=Inf
          sum1 = sum1+distance(i1);
          n1 = n1+1;
      end
  end

   aver_dis1 = sum1/n1;
```

Figure 2: Calculate the mean value of the distance between the robot and the wall

```
% Make the differences calculation, in order that I can use these
% values to order the robot moving
    error = aver_dis1-aver_dis2;
    error1 = aver_dis1-aver_dis3;
    error2 = aver_dis2-aver_dis3;
```

Figure 3: Calculate the error between each mean value of the distance

mean value is larger than the front side mean value, make the robot turn to the right. Then same principles are used in the other directions. Thirdly, If the robot almost reaches the center point, stop the robot and give 'begin' the other value to make it jump out of the while loop. The specific operation and velocity value is in the m file.

3.2.2 Rotating at the center point:

At the center point, the robot need to rotate one round and have two tasks:
1. Draw the obstacles maps.
2. Find the widest door on which side of the obstacles.
(1)The specific solution for the first task is that:
Using the time to control the rotating angle. Firstly, the robot needs to rotate a whole circle, which is 360 degrees. And the angular velocity is defined by myself, so that i can use the following formula to calculate the target rotating time.

$$time = \frac{angle}{angular\_velocity}$$

The same theory used to find the target door direction. I set every four door a direction value, and the value represent the direction angle that the robot should rotate to. For example, the upper door represent the $\frac{\pi}{2}$. The left door represent the π, the right door represent the 0 and the under door represent $\frac{3\pi}{2}$. Then, I use the angle values to calculate the rotating time that the robot needed to turn to that target angle. And also, I begin to record the time at the beginning of the while loop. Inside the wile loop, I also check whether it has running the target time.

(2)The specific solution for the second task is that: 1.I set four counters and give them initial values as 0. With the help of time, when the robot has rotated the time which the direction should, the robot will know which door it is facing now, and it will use that counter to record the door's width. Because as the robot rotating, the head of the robot is pointing to the wall. So it should record the head LIDAR line's data(The index is 540) in order to record the door's width.If the LIDAR feedback value is Infinite, The counter should add one.

Paying attention:

In the front of one door, because there is a cylinder. So the detecting data might not Infinite. But if we ignore the section, the door's width might become lower. So I defined that if the feedback length equal to infinite or less than a small value, the counter will add one.

2. As it rotating a whole circle, recording the door's width: When the robot rotate less than 90 degrees, it is recording the first door's width. When the robot rotate larger than 90 degrees less than 180 degrees, it is recording the second door's width. And so on. I used the time to control the rotating degrees. Additionally, receive the current model_state position and using quaternion calculate the position angle. Then, Using the angle to draw the world map.

The specific solution:

Using the LIDAR feedback data as the length, combined with the position angle, using trigonometric formula to calculate the obstacle coordinates. The formulas are like following:

$$X = length \times cos(position\_angle)$$
$$Y = length \times sin(position\_angle)$$

3. If the robot rotating a whole circle and has already draw the area map, stop the robot and give 'begin' the other value to make it jump

out of the while loop. The specific operation and velocity value is in the m file.

3.2.3 Finding the maximum door:

After comparing, find the widest door. And make the robot run to the relating direction. For example, if the biggest door is the upper one, control the robot rotate to the π.

The specific solution for the task is that:

Receive the pose data from Odometry subscriber. Using the function quat2eul() to convert the receiving quaternion to the three dimensions angle.

Paying attention:

If the converting result is negative, it should firstly change the negative one to the positive one in the range of 0 to 2π. The specific code is like below:

```
if rotation(1) < 0
    rotation(1) = rotation(1) + 2*pi;
else
    rotation(1) = rotation(1);
end
```

Figure 4: Change the angle to the range of 0 to 2π

3.2.4 Running the robot through the maximum door:

The final step is to control the robot to avoid the obstacles through the target door. The avoiding algorithm I used is same as the Homework 6. The specific process is in the Homework 6 report. Additionally, I used the front line feedback data to control the robot to stop. If the detecting data is Infinite, change the data to 0. When the average of the front lines feedback are 0, I defined the situation as pass through the door. Then stop the while loop. Finally, plot the robot path. Through the whole running process, I use the following code to record the robot coordinates.

```
% This section is used to receive the position information from the
% model_state, in order to plot the trajectory figures.
      model=receive(model_subs);

      position_X(n) = model.Pose(22,1).Position.X;
      position_Y(n) = model.Pose(22,1).Position.Y;
      n = n+1;
```

Figure 5: Record the robot coordinates

3.2 Task B:Using the Kinect Lidar

1.The whole idea:

The whole idea is same as the Task A; Firstly, make the robot run to the center of the enclosed area; Then, rotating a whole circle and drawing the area map. Additionally, finding the widest door. Finally, running through the widest and avoiding the obstacles.

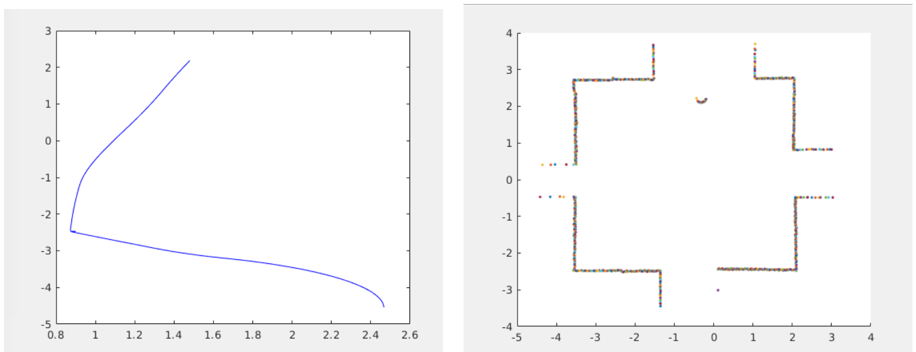
Paying attention:

The different part compared with Task A is that the robot using different method running to the center of the enclosed area. The robot first rotate to the destinating direction, then control the robot to run along the direction line. When the robot almost nearing the goal point, stopping the robot.

4 Results

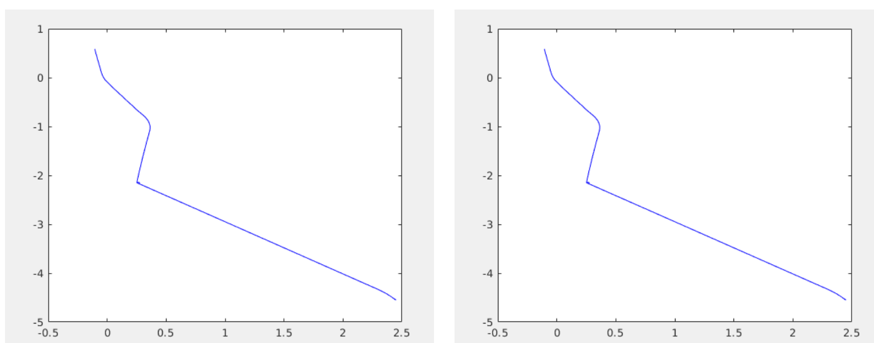
1. The result videos are in the zip file.

2. The following are the result figures.



(a) The robot path for Task A (b) The area map for Task A

Figure 6: Task A



(a) The robot path for Task B (b) The area map for Task B

Figure 7: Task B

5 Discussions and Conclusion

There are two problems in this homework. **The first one:** As shown in the figure 8:

In the Task B, because the Kinect Lidar's range is small, so that if I use the same algorithm as Task A, the robot can not reach the center point. Because the range of the Kinect Lidar is just 10 meters, so it can not detect the surrounding walls.

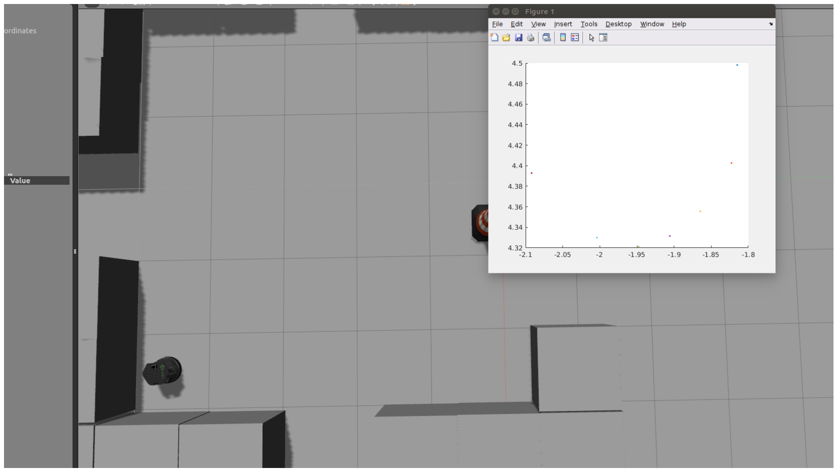


Figure 8: The first problem

The solution:
So, for Task B, I changed to the current algorithm.

The second one: As shown in the figure 9:

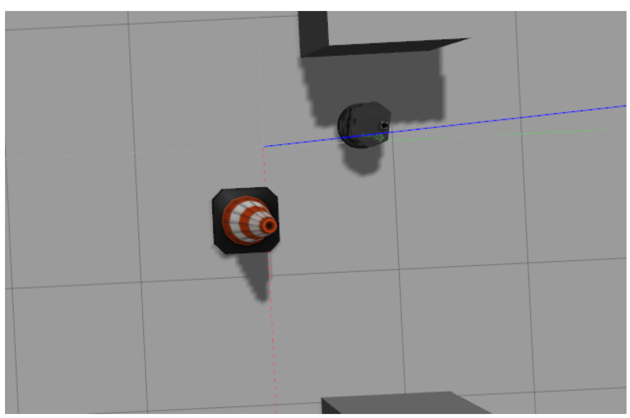


Figure 9: The second problem

In the Task B, because the Kinect Lidar's range is small, so that when the robot pass through the target door, it cannot fully ass through that door. I think the reason is because of my algorithm. I used the front detecting lines to detect. If the average of front lines feedback is

Infinite, stop the robot and plot the robot path. Because the detecting range is 10 meters, so the robot stop before it fully pass through the door.
The solution:
I didn't solve the problem.

References

A Appendix

The following are the content in my zip file

1. MATLAB code

2. Video :
   Task A: About using the Hokuyo Lidar
   Task B: About using the Kinect Lidar

3. Figures :
   Task A: The robot path figure and the enclosed area map figure
   Task B: The robot path figure and the enclosed area map figure
   Problems: Two problems figures