RobotLab 3: Follow a line with Closed-Loop Feedback

ENGN 4627/6627 Robotics 2018 S2

August 27, 2018

1 Purpose

This lab allows you to program your turtlebot to go around a fixed, known path.

There are two task in Lab 3, implement a closed-loop motion control using computer vision sensor, in our case the Kinect RGBD sensor, and get familiar with a cylinder detector package.

The full mark for Lab 3 is 15 points, which include up to 4 points for DEMO, 10 points for Lab Report, and 1 points for the quality of your source code.

2 Preparation

1. You should complete Lab 2, and master how to create, compile a ROS project to drive the turtlebot around.
2. You are expected to know some basic computer vision techniques. You need to finish reading lecture week6, as well as Chapter4 of the textbook.
3. Go through the tutorials in the following webpage (OpenCV in ROS).
   (a) `http://wiki.ros.org/vision_opencv`
   (b) `http://wiki.ros.org/cv_bridge/Tutorials/ConvertingBetweenROSImagesAndOpenCVImagesPython`
4. Go through the scripts in *pybot.zip* file (on Wattle). Learn from these to write your own codes later.

3 Task 1:

Within a work space and starting at a marked location (a red perpendicular line), the Turtlebot is required to navigate an arcdoor path and stop on the ending point that is marked.

Your robot must announce the start of its journey by beeping three times.

When your robot completes this task, and stops at the finish line, it must play the song three blind mice" to declare its "mission completed".

There will be a tape on the floor to guide the robot. Your robot must use its onboard Kinect sensor to perform this task. No modification, no change of position of the kinect sensor is allowed. The entire task must be completed within reasonable time duration, say around 3–5 minutes.

The performance of your final DEMO will be based on quality of tracking the path as determined by the marker.

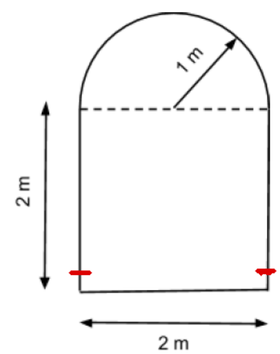


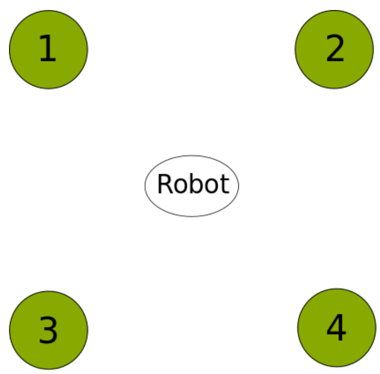
Figure 1: Path

Tutor will mark your final DEMO based on the path tracking performance.

4 Task 2: Reach the cylinder number 4

In this task four cylinders are arranged to surround the turtlebot, as is shown in Figure 4, the position of each cylinder is aleatory. To do this task a cylinder detector package is provided to you which has been explained in the Tutorial 3 (`https://github.com/crodriguezo/cylinderDetector`).

You have to program the robot to autonomously turn towards the cylinder number 4 as soon it is detected and get closer to it (do not hit the cylinder). After the robot is facing the cylinder it has to call the number/label of the cylinder and the distance to it.



<u>Note</u>: Because the closest distance that the Kinect sensor/camera can detect/see is 0.6 metres, there is a 0.6 metres or so "blind zone" in front of the Turtlebot. Within this about 0.6m, your turtlebot may not be able see anything.

<u>Marking</u>:

- The robot correctly identifies the line on the ground [0.5]
- The robot follows the line (robot base still overlaps the white tape) with closed-loop computer vision feedback (0.5 marks will be deducted each time or every second the robot got too far) [1.0]
- The robot correctly identifies the red perpendicular (finish) line on the ground [0.5]
- The robot stops on the finish line correctly (robot base overlaps the red line completely) [0.5]
- The robot plays a song/sound to signal that it reaches the finish line. [0.5]
- The robot correctly turns, then stop when facing the cylinder 4 [0.5]
- The robot calls the number/label of the cylinder and distance correctly. (Hint: *spd-say*) [0.5]

5 What to Hand In (due in Week 7)

Pack your source code and your Lab report (3 page A4) in a single Zip file. Name your zip file in the following format:
" **⟨your_last_name⟩_⟨your_uni_id⟩_Lab_⟨ X ⟩.zip** ".

Upload your single zip file to Wattle, at the corresponding Lab upload link. Since this lab is done by your group, the source code you hand in will be very similar if not identical.

However, your Lab Report must be your own **individual work**, which should not be similar to your group mates'. Your overall Lab marks will be based on the performance of your DEMO, the clarity of your source code, and the quality of your Lab Report.

The Lab Report will be based around answering some technical questions. These reports do not have to be formal, but are compulsory to test your understanding of the work done in this lab.

The report will be marked based on your answer, which should demonstrate your understanding of the subject asked. There will be 4-6 questions, and each question contributes 0.5-2 marks, the demonstration 4 marks, while the clarity of source code contributes to the remaining 1 marks.

In summary: your demonstration + the clarity of your source code (5 marks), and correctness and quality of your 3 page lab report (10 marks). In total, these make up 15 marks as the full mark for each lab.

DUE DATE:

Lab 3 demonstration will be due in the last 1.5 hours of your Week 7's Lab session.

The lab report (see next page) will due on Sunday of the same week.

6 Turtlebot Lab 3 Report

In your Lab 3 Report, you only need to answer the following questions.

(Note: Lab Report must be your individual work.)

Question 1: Describe the strategy used by your vision/control algorithm to detect the finish point of the acrdoor path (i.e., the red tape mark). (1 mark)

Question 2: Given that the mounted Kinect sensor can only see the tape at a minimum distance of about 0.5m. How does the designed control algorithm correct from the drifted position of the tape as seen by the camera with respect to the current position of the robot? (2 marks)

Question 3: Describe how does the implementation of the algorithm pass messages from the vision sensor to the control loop. (2 marks)

Hint: If multiple threads were used, how do you synchronize access to shared variables? If instead you coded two separate ROS nodes, what is the Publisher-Subscriber graph and the format of the messages? If some other approach was used, give detailed explanation on how it works.

Question 4: Classify the following robot sensors into "active sensor" or "passive sensor", "exteroceptive sensor" or "proprioceptive sensor": (2 marks)

1. Bumper
2. CCD Camera
3. CMOS Camera
4. SICK Laser scanner
5. Gyroscope
6. Wheel encoder
7. A Compass
8. A GPS receiver

Question 5: More and more modern robots are equipped with video cameras as the primary range sensors.

1. Please name three advantages (benefits) of using camera (and robot vision) compared with other range sensors (such as sonar, lidar, or other types of TOF cameras). (1.5 mark)

2. Microsoft Kinect is a powerful range sensor based on the structured light principle. It is however not without limitations. Can you list three drawbacks of using Kinect Sensor, in a typical mobile robot application scenario? (1.5 mark)