RobotLab 2: Move Along a Fixed Path

ENGN 4627/6627 Robotics 2018 S2

August 13, 2018

1 Purpose

This lab will help you practice the basic kinematics and motion control techniques. In the first task, you are asked to program the turtlebot to go around a fixed known path (e.g., a square path with an arch (semi-circle) at one end). The second task asks you to predict the locations (and the entire path) of your robot using wheel encoder odometry.

2 Preparation

This lab assumes that you're familiar with ROS framework/environment and have learned how to write Python code under ROS. It is also assumed that you have learned about the kinematics theory and motion prediction in the lectures. Otherwise, you need to review the lecture content as well as tutorial materials.

In particular, you are strongly recommended to try to work out "Question-4" and "Question-5" in this lab instruction document (towards the end of the this document), even before attending the first Lab session.

3 Task 1: move along a fixed path using open-loop control

Program the robot to autonomously move along a $2m \times 2m$ square with a semicircular path at one of the edges (Figure 1).

You can pick one of the two corners of the square as the starting position, and heading (the desired moving direction). The robot must then return to the same position, at the same heading direction, after completing the entire path.

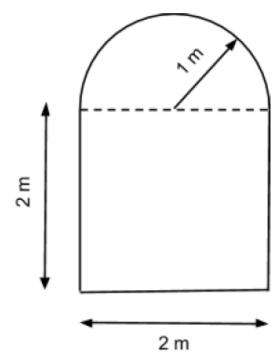

Figure 1: Path

In doing this experiment, you will need to choose the best parameters such that your robot can follow the prescribed path as accurately as possible. Note that the robot must perform its task fully autonomously, therefore no human intervention is allowed. Also, the mission must be completed within a reasonable amount of time (e.g within 1-2 minutes).

You are required to use open-loop control (i.e., no feedback). There will be a masking tape on the floor, which is however only used to visually verify that your robot is following the desired path.

Once this task is completed, you can move on to remaining tasks. Please notice that task 2 is optional, you will get 1 bonus mark if you complete task 2 before moving on to task 3 and 4.

4 Task 2 (Optional): Odometer and Gyroscope Calibration

Note, this is an optional task. Performing odometer and gyro calibration will give you more reliable readings for better motion prediction. 1 bonus mark will be awarded if you complete this task before moving on to task 3 and 4. Please make sure to notify your tutor(s) when you complete this task.

Note: This link provides a good instruction of how to perform odometer and gyro calibration. (`http://wiki.ros.org/turtlebot_calibration/Tutorials/Calibrate%20Odometry%20and%20Gyro`) (Gyro Chip: ADXRS652)

5 Task 3: Odometer Motion Prediction

Your task is to use the odometer measurements (given as estimated linear and angular velocity in the robot frame) to estimate the position of the robot. Below is the summary of the tasks to be completed.

Tasks:

- Add Python codes to use the odometer sensor readings (i.e. linear and angular velocities, and time-stamps) to estimate robot's pose at time step t, $(x, y, \theta)_t$. The initial robot pose is $(0, 0, 0)_0$, which is reset every time your node is restarted.
- Publish your estimate to a new ROS topic, using the same message type as the original odom topic (nav_msgs/Odometry)
- Record the estimated pose and timestamp of the robot in a text file, and plot the estimated robot motion and expected robot motion in your report.
- Visualise the estimated robot position (x,y) in realtime during DEMO using RVIS.

Note: You are only allowed to use the instantaneous rate of change of motion (i.e., msg.twist) to estimate the robot pose. Do not use the integrated motion (i.e., msg.pose).

6 Task 4: Odometer using gyroscope

Your Turtlebot is also equipped with a gyroscope sensor. Instead of using odometer angular velocity, please repeat the above experiments by using gyroscope reading for angular estimation (θ).

7 What to Hand In (due in Week 5)

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:

All groups should demonstrate their finished tasks during the last 1.5 hours of Week 5's lab sessions. Groups' Demo will only be assessed during the lab session they signed in.

The lab report (see next page) will due on Sunday Week 5.

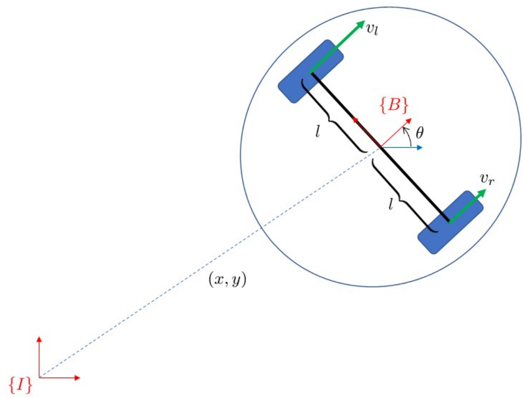

Figure 2: TurtleBot Figure

8 Turtlebot Lab 2 Report

In your Lab 2 Report, you only need to answer the following questions.

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

Question 1: What are the sources of errors for the inaccuracy in the actual motion from the intended motion along the robot path ? Please list at least two sources of errors. (1 mark)

Question 2: (ROS programming) What is the type of the message you read from the sensor?
List the members of the message in detail and explain their meanings, and also give a typical values you observed. (1 mark)

Question 3: Plot the estimated robot pose/motion using MATLAB or *Python Matplotlib* and append the graph in your report.
Comment on what you can observe from these graphs. (3 marks)

Note: Use your own judgement to decide what kind of graphs (what are the suitable axes, use of cumulative sum, etc?) to use, in order to best illustrate and compare the data collected.

Question 4: Consider a two-wheel differential drive robot such as the TurtleBot

(Fig. 2). Attach a frame of reference $\{B\}$ to the centre point between the two wheels of the robot as shown in the Figure. Suppose $(x, y, \theta)^\top$ is the robot's current position and orientation in a world frame $\{I\}$. The distance between the two wheels is $2l$. Denote v_l and v_r the left wheel's, and right wheel's linear velocities. Imagine applying constant velocities v_l and v_r for t seconds. That is allow the robot to move forward along the curved path generated by constant wheel speed. Denote the robot's pose after t seconds by $(x_t, y_t, \theta_t)^\top$.

(a) Under the assumption that the ratio of v_l/v_r is constant, which is true for constant v_l and v_r the motion of the robot is holonomic, that is it follows a fixed trajectory (like train tracks) parametrized by the time t for which it rolls. Compute a function f such that

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = f(t; l, x, y, \theta, v_l, v_r)$$

(1 marks)

(b) Suppose $l = 0.5$m. Assume the initial pose is $(x = 1.5\text{m}, y = 2.0\text{m}, \theta = 180°)^\top$.

The robot executes the following sequence of steering commands:

(b.1) ($v_l = 0.3m/s, v_r = 0.3m/s, t = 3$ seconds). Compute its position after the command finishes. (1 mark)

(b.2) Starting from where the robot finished in the previous command, apply ($v_l = 0.1m/s, v_r = -0.1m/s, t = 1$ seconds). Compute its position after the command finishes. (1 mark)

(b.3) Starting from where the robot finished in the previous command, apply ($v_l = 0.2m/s, v_r = 0m/s, t = 2$ seconds) Compute its position after the command finishes. (1 mark)

Question 5: (Discritized wheel odometry)

Consider a two-wheel differential drive robot. The diameter of each wheel is denoted D. The distance between the two wheels is denoted $B = 2l$. The time axis is discretized into intervals of length $\delta t > 0$ and indexed by $t \in \mathbb{N}$.

Suppose $(x_{t-1}, y_{t-1}, \theta_{t-1})^\top$ is the robot's pose at time $t-1$, in a world frame. Denote ω_l and ω_r the left wheel and the right wheel's angular velocities and assume they constant during each time interval. Let $(x_t, y_t, \theta_t)^\top$ denote the robot's pose at time t.

Consider a general discrete kinematics algorithm

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} \delta_t x \\ \delta_t y \\ \delta_t \theta \end{pmatrix}$$

Write the formulae for the incremental movements $(\delta_t x, \delta_t y, \delta_t \theta)^\top$ as a function of known parameters. (Hint: read the course notes for Lecture 3b) (1 mark)