MEEM 4707: Autonomous system

Fall, 2022

Lab – 3 & 4

By

Chaitanya Jaolekar and Grace Ackerman

Name	Logic Build	Coding	Report Writing	Total
Grace Ackerman	50%	50%	50%	150%
Chaitanya Jaolekar	50%	50%	50%	150%

Problem 1

1. Set up a timing-based controller to draw a square of dimension 0.5 m, counterclockwise, once.
2. Set up a timing-based controller to draw a square of dimension 1 m, clockwise, once.

Capture your trajectory and the estimated trajectory by the robot (with encoders) in Gazebo and include them in your report. Note that you may not be able to follow the squares strictly, and you don't need to try to have exact square routes. We will work on the calibration using this data during the next lab.

COUNTER-CLOCKWISE:

The graph of estimated trajectory and recorded trajectory during the lab for counter-clockwise motion are being shown below. The video and the code used for this assignment are being attached separately.

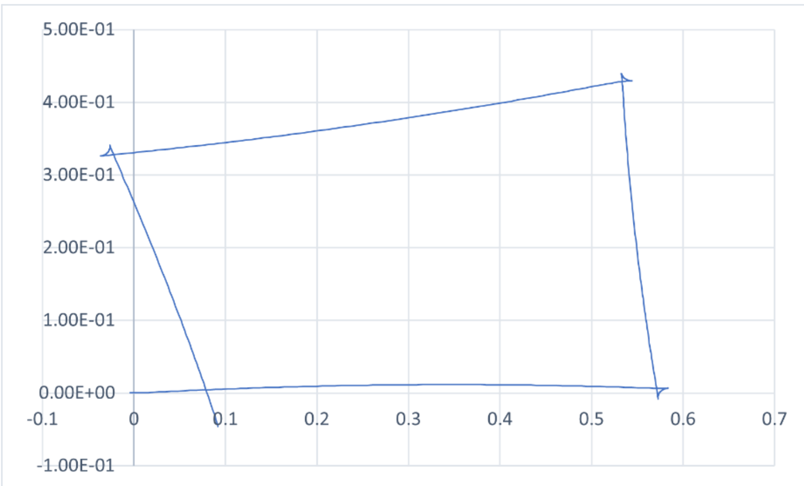

Figure 1 Estimated Trajectory (0.5m CCW)

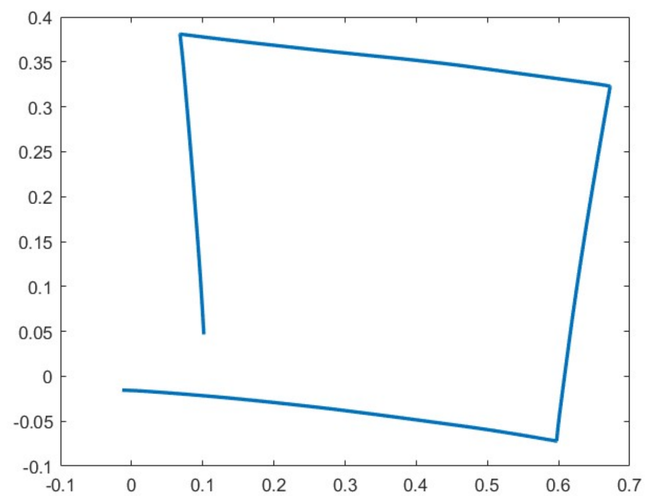

Figure 2 Actual Trajectory of the Robot (0.5m CCW)

CLOCKWISE:

The graph of estimated trajectory for clockwise motion is being shown below. There were some issues during recording the actual trajectory of the robot for this motion. After consulting with the TA, we decided to submit the recorded video only since the issue couldn't be resolved. The video and the code used for this assignment are being attached separately.

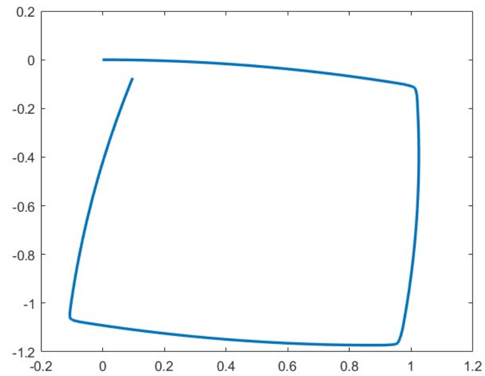

Figure 3 Estimated Trajectory (1m CW)

Problem 2.

Use the tuned parameters in Gazebo to let the robot follow a 0.5-meter square counterclockwise once. Capture your desired trajectory and the estimated trajectory by the robot (recorded with the encoders) in Gazebo and include them in your report. Compare the tuned trajectory with the trajectory in Problem 1 and discuss the difference.

ANSWER: The equations used to calculate the parameters a1,a2,a3 and a4 are as follows

$$V_{actual} = a_1 \times V_{command} + 0.25 \times a_2 \times b \times \omega_{command} + a_3$$
$$\omega_{actual} = \frac{a_2}{b} \times V_{command} + a_1 \times \omega_{command} + a_4$$

The code used to compute the values of the tunning parameters from above equations is shown below:

```
syms a1 a2 a3 a4

eqn1 = 0.2*a1+a3==0.200185;
eqn2= (1/0.287)*a2*0.2+a4==-0.09474;
eqn3= 0.25*0.287*0.9*a2+a3==0.000633;
eqn4= 0.9*a1+a4==0.89575;
[X,Y]=equationsToMatrix([eqn1,eqn2,eqn3,eqn4],[a1,a2,a3,a4])
A=double(linsolve(X,Y))
```

The values of the parameters after running the above code are:

a1	0.9538
a2	-0.0896
a3	0.0056
a4	0.0242

The above values were used to evaluate the command velocities to get the required desired velocities. The command velocities are as follows:

Command Velocities	Straight Path	Turning
V_cmd	0.2041	0.0003
omega_cmd	0.0414	0.9183

Following curve shows the trajectory recorded in the gazebo environment after parameter tunning:

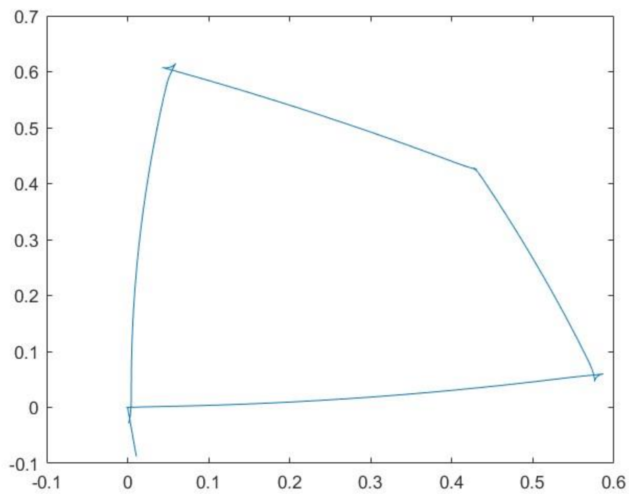

Figure 4 Trajectory after parameter tunning

Discussion from Lab3&4

- ## Objective
  - The objective of this lab was to learn and complete timing-based control and then complete motion calibration and calculation of tunning parameters for TurtleBot 3.
  - To develop an understanding of the code and its implementation on the real robot and in the virtual gazebo environment.

- ## Approach to achieve the Objective
  - The approach to successfully achieve the above-mentioned objectives was to complete the prelab assignment and understand it thoroughly.
  - Run your code in the gazebo environment first before running it on the actual robot to avoid any accidents.
  - Work with your teammate and exchange ideas to achieve the desired goals.

- ## Challenges faced and countermeasures taken
  - We faced challenges while implementing our clockwise motion code on the real robot.
  - After several iterations (varying the time parameters for each iteration), we got a fair square-shaped trajectory for CW motion. However, there was some error while recording the path of the robot.
  - After consulting with the TA, we decided to only submit the video of the CW motion as the issue couldn't be resolved.
  - We faced additional challenges while recording the tuned trajectory as we were not able to follow the desired square path.
  - Our robot's path before tunning the parameter was more accurate than the path followed after parameter tunning.
  - We tried changing our command velocities and adjusted our timing-based control but still we were not successful in tracing the desired square.

- ## The difference in strategy: Pre-lab vs. Lab strategy
  - The major difference was in calculating the parameters for tunning.
  - In pre lab I used random values from the CSV file whereas during lab we followed the procedure explained in the video to calculate the parameters.
  - Changing the time parameter in the code to get a close enough trajectory.

- ## Observations and Learnings
  - The trajectory plotted from the gazebo environment is slightly different from the one recorded after running the code on the actual robot.
  - After parameter tuning, we were not successful in following the desired square path. We tried many iterations by changing velocities and redid all the calculations but to no avail.