Lab #1: Odometry Estimation

In this experiment, you will estimate the motion trajectory of a **TurtleBot3 Waffle Pi** wheeled robot using its wheels encoders, then you will compare the estimated trajectory with the desired trajectory.

The **TurtleBot3** uses two independently actuated side wheels and two passive ball wheels for support (Figure 1). This configuration behaves equivalently to a two-wheeled differential drive system, enabling precise modeling and control using standard differential drive kinematics.

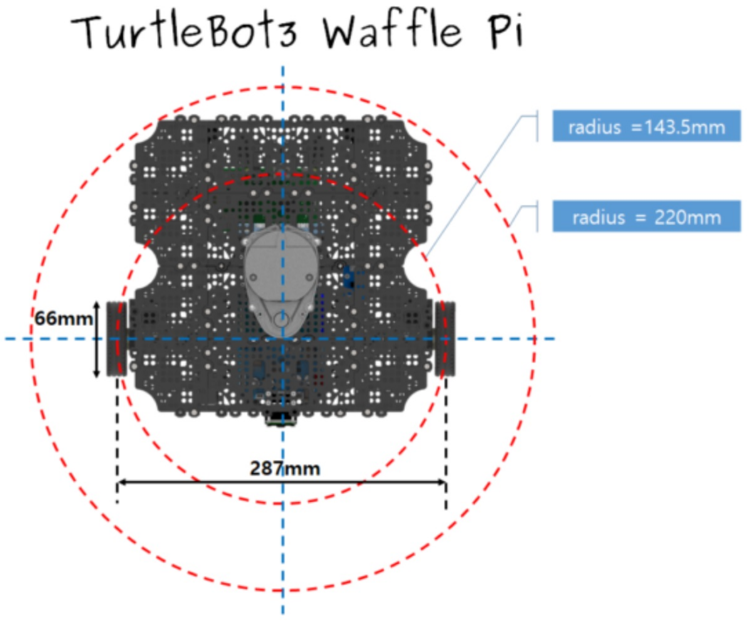

Figure 1: TurtleBot3's Specifications

The robot's position on the global x–y plane and the angle θ between the heading and the global x–axis are used to describe the motion as in Figure 2.

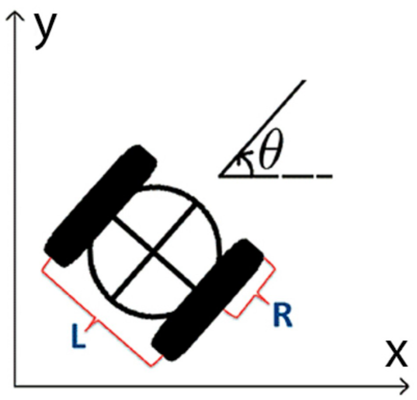

Figure 2: Differential Drive Robot Model

By using the incremental motion of the right and left wheels, the robot motion can be modelled in a discrete manner as:

$$x(k+1) = x(k) + D_c \cos\left(\theta(k) + \frac{\Delta\theta}{2}\right), \tag{1}$$

$$y(k+1) = y(k) + D_c \sin\left(\theta(k) + \frac{\Delta\theta}{2}\right), \tag{2}$$

$$\theta(k+1) = \theta(k) + \Delta\theta. \tag{3}$$

where D_c is the distance travelled by the axle center over one sample, and $\Delta\theta$ is the change in heading over one sample. The variables $x(k+1)$, $y(k+1)$, and $\theta(k+1)$ represent the position and orientation of the robot at the next time step.
Let D_r and D_l be the distances travelled by the right and left wheels:

$$D_r = 2\pi R\,\frac{\Delta E_r}{360}, \qquad D_l = 2\pi R\,\frac{\Delta E_l}{360}, \tag{4}$$

$$D_c = \frac{D_r + D_l}{2}, \qquad \Delta\theta = \frac{D_r - D_l}{L}. \tag{5}$$

Here, R is the wheel radius, L is the wheel separation (track), and ΔE_r, ΔE_l are encoder difference values in **degrees** for a specified time period for right and left motors, respectively. **Initially**, you may assume $x(0) = 0$, $y(0) = 0$, and $\theta(0) = 0$.

In this lab, you are expected to implement a differential-drive robot model and collect **position** and **orientation** data during the robot's motion.

Environment Set-up

Refer to the Lab #0 document to connect to the **TurtleBot3** and prepare the MATLAB environment in order to start the algorithm implementation.

Things to do:

- Submit your report **via SUCourse** until the report submission deadline.

Post-Lab Report Deadline: 22 October 2025, 23:55 via **SUCourse**

- Your report must include:
  - **Introduction**
  - **Procedure**
  - **Results**
  - **Conclusion**
  - **Discussion**
  - **Appendix**
  - Provide your **MATLAB** codes in Appendix section appropriately.

Answer the following questions in the Discussion section of your Post-lab report:

- **Linear Motion**
  - Apply linear motion several times for forward and backward directions. What does your calculated odometry look like? What was your desired trajectory? What is your error? Generate your desired trajectory and plot both the desired trajectory and the calculated odometry on the same figure.
  - Decrease and increase the forward speed v. How do odometry estimates and error change?
- **Circular Motion**
  - Apply circular motion in left or right direction. What does your calculated odometry look like? What was your desired trajectory? What is your error? Generate your desired trajectory and plot both the desired trajectory and the calculated odometry on the same figure.
  - Decrease and increase the sampling time for ΔE_r, ΔE_l. How do odometry estimations and error change?
- **Combined Motion**
  - Make a forward motion for 3 seconds ($v = 0.1, \omega = 0.0$), then a circular motion for 3 seconds ($v = 0.05, \omega = 0.1$), and finally another circular motion for 3 seconds ($v = -0.1, \omega = -0.05$). Plot both the desired trajectory and the calculated odometry on the same figure.

Hint

For easier implementation, follow the pseudo code provided in the appendix for Linear and Circular motion, and create a separate script for Combined motion.

Appendix

- Pseudo code for your MATLAB script:

```
clear; close all; clc;
clear node;

% Environment Setup

% Publishers & Subscribers

% Initial Encoder Reading

% Parameters Initialization

% Start Movement

exeTime = 3;
period = 0.01;
t0 = tic;
while toc(t0) < exeTime
    % Read Encoders

    % Odometric Estimation

    % Values Recording (x,y,theta)

        pause(period);
end

% Stop Movement

% Plot the results (Insure proper plot limits)
```