Lab #2: Motion Control of a Differential Drive Robot

In this experiment, you will implement the positioning controller derived in the lecture. Assume that the odometry estimations provide the position and orientation of the robot accurately enough; therefore, you will utilize them in your implemented controller.

Your controller should **successfully** take the robot from a starting state (position and orientation) to a goal state, while taking into account the actuation limits of the robot. Please follow the instructions below for implementation.

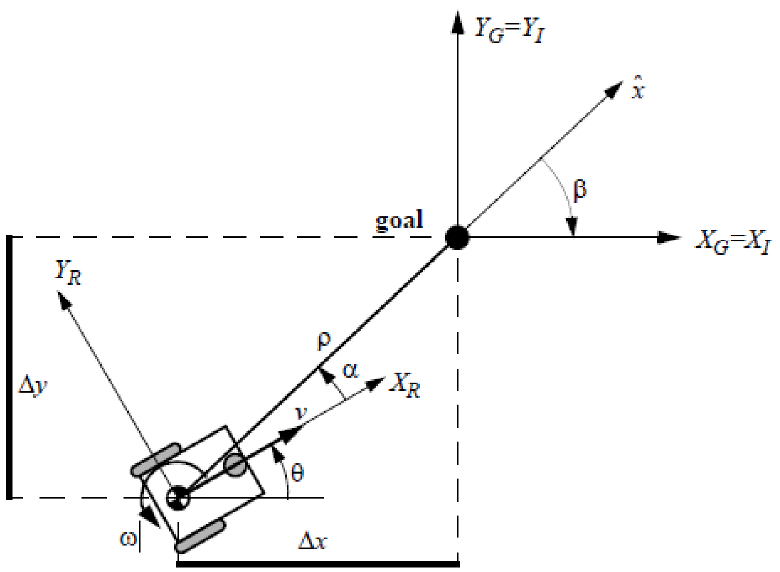

Figure 1: Robot Kinematics after Polar Transformation

Recall the polar coordinate transformation of the robot kinematics shown in Figure 1 and given by the following equations:

$$\rho = \sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{1}$$

$$\alpha = -\theta + \operatorname{atan2}(\Delta y, \Delta x) \tag{2}$$

$$\beta = -\theta - \alpha \tag{3}$$

where $\Delta x = x_G - x$ and $\Delta y = y_G - y$. Now consider the control law derived in class as:

$$v = k_\rho \rho \tag{4}$$

$$\omega = k_\alpha \alpha + k_\beta \beta \tag{5}$$

Odometry Estimation Recall

Feedback signals (x, y, θ) will be obtained by odometry estimation using the following equations:

$$x(k) = x(k-1) + D_c \cos\left(\theta(k-1) + \frac{\Delta\theta}{2}\right), \tag{6}$$

$$y(k) = y(k-1) + D_c \sin\left(\theta(k-1) + \frac{\Delta\theta}{2}\right), \tag{7}$$

$$\theta(k) = \theta(k-1) + \Delta\theta, \tag{8}$$

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 $x(k)$, $y(k)$, and $\theta(k)$ represent the position and orientation of the robot at the current time step, while $x(k-1)$, $y(k-1)$, and $\theta(k-1)$ represent the position and orientation of the robot at the previous time step. Let D_r and D_l be the distances travelled by the right and left wheels:

$$D_r = R\,\Delta E_r, \qquad D_l = R\,\Delta E_l, \tag{9}$$

$$D_c = \frac{D_r + D_l}{2}, \qquad \Delta\theta = \frac{D_r - D_l}{L}. \tag{10}$$

Here, R is the wheel radius, L is the wheel separation (track), and ΔE_r, ΔE_l are encoder difference values in radians for a specified time period for right and left motors, respectively.

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:

- You are expected to write **motionController.m** script file. The overall program is given in **page 4**.
- Submit your report **via SUCourse** until the report submission deadline.

Post-Lab Report Deadline: 29 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:

1. How did you adjust the control gains k_ρ, k_α, and k_β? How do they affect the robot's motion?
2. Choose $\alpha \in (-\pi/2, \pi/2]$ and test the robot's parking performance for various initial positions and orientations. Do ρ and θ converge to zero? Discuss the resulting motion trajectories.
3. Choose $\alpha \in (-\pi, -\pi/2] \cup (\pi/2, \pi]$ and test your robot's parking performance for various initial positions and orientations. Do ρ and θ converge to zero? Discuss the resulting motion trajectories.

Hint

For easier implementation, follow the pseudo code provided in the appendix.

Appendix: Psuedo Code

```
clear; close all; clc; clear node;
%% 1 Environment Setup
%% 2 Publishers & Subscribers
%% 3 Initial Encoder Reading
%% 4 Parameters
x = -0.3; % Your Starting x [m]
y = -0.3; % Your Starting y [m]
theta = 0; % Your Starting theta [rad]
x_goal = 0.0;
y_goal = 0.0;
R = 0.033; % Wheel Radius [m]
L = 0.287; % Axle Distance [m]
%% 5 Variables Initialization
%% 6 Main Loop
velMsg = ros2message(velPub);
exeTime = 20; period = 0.01; t0 = tic;
while toc(t0) < exeTime
    %% 6.1 Read Encoders
    %% 6.2 Odometric Estimation
    %% 6.3 Values Recording (x,y,theta)
    %% 6.4 Motion Controller
    pause(period);
end
%% 7 Plotting Results
figure; plot(x_values,y_values);
xlabel("X Position (m)"); ylabel("Y Position (m)");
xlim([-0.5 0.5]); ylim([-0.5 0.5]); grid on;
figure; subplot(1,4,1); plot(x_values);
xlabel('Time'); ylabel('X [m]'); grid on;
subplot(1,4,2); plot(y_values);
xlabel('Time'); ylabel('Y [m]'); grid on;
subplot(1,4,3); plot(rho_values);
xlabel('Time'); ylabel('Rho [m]'); grid on;
subplot(1,4,4); plot(theta_values);
xlabel('Time'); ylabel('Theta [rad]'); grid on;
```