Omni-Wheel Robot - Motion Tracking & Visualization

This notebook will help you **see** how the robot moves! After each movement, we will **plot the robot's path** to visualize the motion.

Learning Objectives

- Write and test movement scripts.
- Track the robot's position in 2D space.
- Visualize how speed, duration, and direction affect movement.

In [ ]:
```
import time
import random
import sys
import os

# Add parent directory to the Python path
sys.path.insert(0, os.path.abspath('..'))
import rclpy
import matplotlib.pyplot as plt
import numpy as np
from controllers.omni_robot_controller import OmniWheelControlNode  # Import

# Initialize ROS2 node
rclpy.init()
node = OmniWheelControlNode()

# Track the robot's position
position = [0, 0]  # Start at origin
path = [tuple(position)]  # Store movement history

def update_position(direction, speed, duration):
    """Update the estimated robot position."""
    global position, path
    radian_direction = np.radians(direction)
    distance = speed * duration
    position[0] += distance * np.cos(radian_direction)
    position[1] += distance * np.sin(radian_direction)
    path.append(tuple(position))  # Store new position

def plot_path():
    """Plot the robot's movement path."""
    x_vals, y_vals = zip(*path)
    plt.figure(figsize=(6,6))
```

```
    plt.plot(x_vals, y_vals, marker='o', linestyle='-', color='b', label='Ro
    plt.scatter(x_vals[-1], y_vals[-1], color='red', label='Current Positior
    plt.xlabel("X Position (m)")
    plt.ylabel("Y Position (m)")
    plt.legend()
    plt.title("Robot Movement Path")
    plt.grid(True)
    plt.show()
```

Challenge 1: Triangle Pattern

Goal: Move the robot in a triangle shape and visualize the path.

In [ ]:
```
# Move in a triangle pattern and plot movement
for _ in range(3):
    node.move_in_direction(0, 0.5, 2)
    update_position(0, 0.5, 2)
    node.rotate_right(120, 1)
    plot_path()  # Visualize after each step
```

Challenge 2: Zig-Zag Movement

Goal: Move in a zig-zag pattern and track the motion.

In [ ]:
```
# Move in a zig-zag pattern and plot movement
for _ in range(4):
    node.move_in_direction(45, 0.5, 2)
    update_position(45, 0.5, 2)
    plot_path()
    node.move_in_direction(135, 0.5, 2)
    update_position(135, 0.5, 2)
    plot_path()
```

Challenge 3: Spiral Path

Goal: Move in a spiral motion and visualize the expanding pattern.

In [ ]:
```
# Move in a spiral pattern and plot movement
for i in range(1, 6):
    node.move_in_direction(0, 0.5, i)
    update_position(0, 0.5, i)
    node.rotate_right(30, 1)
    plot_path()
```

Challenge 4: Obstacle Avoidance Simulation

Goal: Stop after each movement and visualize the movement path.

In [ ]:
```
# Move with stops and plot movement
node.move_in_direction(0, 0.5, 2)
update_position(0, 0.5, 2)
plot_path()
node.stop_all_motors()
node.move_in_direction(90, 0.5, 2)
update_position(90, 0.5, 2)
plot_path()
```

Shutting Down the Node

Once you're done, **shutdown the node** properly.

In [ ]:
```
node.destroy_node()
rclpy.shutdown()
```