BSc. (Hons) in Information Technology
IE4060 – Robotics and Intelligent Systems
Assignment 01

Intelligent Differential Drive Robot Simulation

Objective

The goal of this assignment is to design, implement, and test an **intelligent controller** for a **differential drive robot (DDR)** that can autonomously move to a target position in a 2D environment. You will simulate the robot in **Python with Pygame**, implement control algorithms (P, PD, PID), and analyze their performance.

Part 1 – Derivation and Background

1. **Derive the mathematical model** of a differential drive robot, starting from its wheel velocities and geometry.
   - Show how linear and angular velocities relate to wheel speeds.
   - Derive equations for position and orientation update.
2. **Define error terms** required for control:
   - Distance to the target.
   - Heading (orientation) error.
3. **Propose control laws** (P, PD, PID) for linear and angular velocity.
   - Clearly explain your reasoning.
   - Show the differences in expected behavior for each controller.

Part 2 – Implementation Tasks

Task 1: Basic Robot Motion (P Control)

- Implement a proportional controller in Python + Pygame.
- Make the robot move toward a single target.
- Visualize the robot, its heading direction, and the target.

Task 2: Trajectory Visualization

- Store the robot’s path during simulation.
- Display the trajectory in Pygame as a line connecting past positions.
- Show the target point and final stopping position.

Task 3: PD Control

- Extend your controller to include derivative action for distance control.
- Compare its behavior with P control (overshooting vs smoother stopping).

Task 4: PID Heading Control

- Implement PID control for the robot’s turning.
- Tune the gains to improve accuracy and reduce oscillations near the target.
- Discuss the improvements observed.

Part 3 – Report Requirements

Each student must submit a **report** (4–6 pages) containing:

1. **Introduction**
    - What is a differential drive robot?
    - Why is control necessary?
2. **Derivations**
    - Derive the kinematic equations.
    - Define and derive distance error and heading error.
    - Derive the control laws (P, PD, PID).
3. **Implementation**
    - Explain your Pygame setup.
    - Show simulation screenshots.

4. **Results and Discussion**
    - Compare results of P, PD, and PID control.
    - Discuss tuning of gains.
    - Explain any difficulties encountered.
5. **Conclusion**
    - Summarize what you learned.
    - Suggest possible improvements (e.g., obstacle avoidance).

Deliverables

- Python code (well-commented).
- Report (PDF).

Evaluation Criteria

- Correct mathematical derivations (25%)
- Working simulation in Pygame (20%)
- Implementation of P, PD, and PID (20%)
- Trajectory visualization and stopping condition (15%)
- Quality of report (analysis, screenshots, explanation) (20%)

This assignment tests your ability to connect **robot kinematics, control theory, and programming** into a practical simulation.

** End of Assignment **