

# Wall following Turtlebot3 with PID controller:

ZAIN-UL-ABIDEEN

450682

RIME-23

## System Design:

The system consists of the following components:

- **Hardware:** TurtleBot3 robot, LIDAR sensor
- **Software:** Robot Operating System (ROS), Python libraries
- **Control Algorithm:** PID controller
- ( $K_p=1.0$ ,  $K_d=0.5$ ,  $K_i=0$ )

## Methodology:

1. Setting up dependencies and libraries for ROS Packages
2. The LIDAR data is processed to extract the distance to the wall.
3. A PID controller is employed to calculate the steering velocity based on the error between the desired distance and the actual distance to the wall.
4. The proportional term ( $K_p$ ) adjusts the velocity proportionally to the error.
5. The integral term ( $K_i$ ) reduces steady-state error by accumulating past errors. (Set to 0.0 in this project)
6. The derivative term ( $K_d$ ) anticipates future errors based on the rate of change of the error.
7. The calculated steering velocity is combined with a forward velocity to achieve both wall following and movement

## Simulation:

- The robot was tested in Gazebo environment with a wall boundary. The PID gains were tuned to achieve smooth and stable wall following while maintaining a desired distance. The obstacle avoidance logic was also evaluated with various obstacles placed in the path.
- Simulation Results in the link below.

<https://www.youtube.com/watch?v=877i0Nle0VU>