NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY
PAKISTAN

MOBILE ROBOTICS

WALL FOLOWING ROBOT USING PID IN ROS2

SUBMITTED TO :
DR YASAR AYAZ

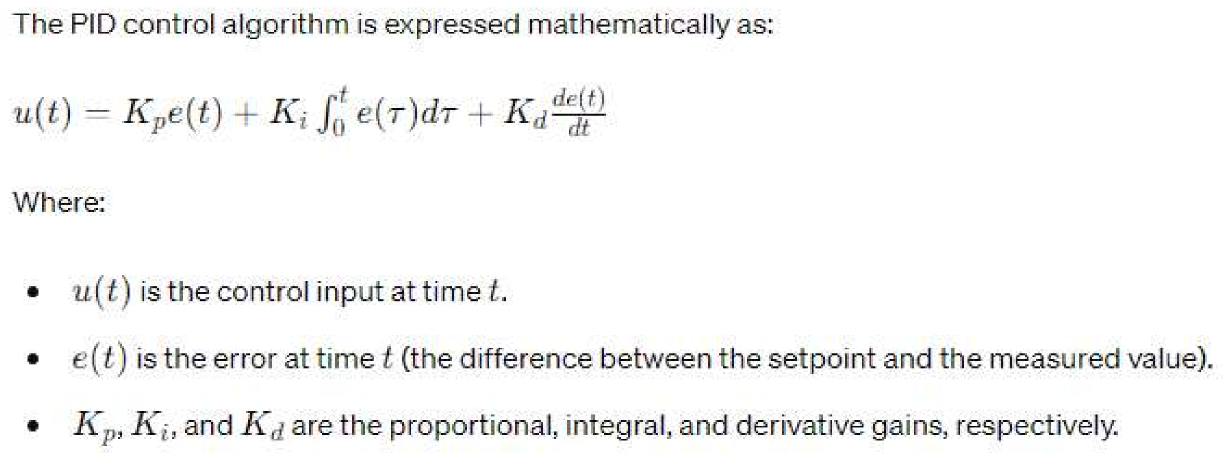
SUBMITTED BY
MESUM ALI IRFANI
450291
RIME-23

Introduction:

Wall following is a fundamental navigation technique employed by robots to traverse environments by closely following walls while maintaining a specific distance. This method is widely used in various robotic applications, including exploration, mapping, and surveillance. In this report, we detail the simulation of a wall-following robot using TurtleBot3 in ROS2 (Robot Operating System 2), implementing a PID (Proportional-Integral-Derivative) control algorithm for precise navigation.

Wall following robots utilize sensors to detect nearby obstacles and maintain a constant distance from walls. The PID control algorithm is a widely used method for controlling systems to achieve desired behavior. It calculates an error value as the difference between a desired setpoint and a measured process variable, then applies proportional, integral, and derivative terms to adjust the control input and minimize the error.

The PID control algorithm is expressed mathematically as:



The PID control algorithm is expressed mathematically as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

Where:

- $u(t)$ is the control input at time t.
- $e(t)$ is the error at time t (the difference between the setpoint and the measured value).
- K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively.

1. **ROS2 Overview:**

ROS2 is the second version of the Robot Operating System, providing a framework for building robot software. It offers improved real-time capabilities, broader language support, and enhanced security compared to its predecessor, ROS1.

2. **TurtleBot3**:

TurtleBot3, developed by ROBOTIS, is a popular, low-cost robot platform available in different models such as Burger, Waffle, and Waffle Pi. It's highly customizable, supporting various sensors and actuators, making it suitable for diverse robotic applications.

3. **Simulation Setup:**

To simulate the wall-following robot in ROS2, the following steps were taken:

- **Install ROS2 Humble:** Ensure ROS2 Humble or a later version is installed on the system, following official installation guidelines.
- **Install TurtleBot3 Package:** Utilize the ROS2 package manager to install the TurtleBot3 ROS2 package, which includes simulation models and control algorithms.
- **Launch Gazebo Simulation:** Launch the Gazebo simulation environment with the TurtleBot3 model to create a realistic 3D environment for testing.
- **Implement PID Wall Following Algorithm:** Develop a PID-based wall-following algorithm using ROS2. This algorithm should utilize sensor data to calculate control inputs that maintain a constant distance from walls.
- **Integrate Algorithm with Simulation:** Integrate the PID wall-following algorithm with the TurtleBot3 simulation, subscribing to sensor topics, processing data, and publishing control commands to navigate the robot along the wall.

4. Results:

Upon completing the setup and integration, the wall-following robot simulation can be executed. The robot should navigate the environment while effectively following walls and avoiding obstacles. ROS2 visualization tools like RViz can be used to monitor and debug the robot's behavior.

5. Conclusion:

The simulation of a wall-following robot using TurtleBot3 in ROS2, employing a PID control algorithm, provides a robust platform for developing and testing robotic navigation systems. By harnessing the capabilities of ROS2 and the flexibility of TurtleBot3, developers can create efficient and reliable robotic solutions for various applications.

6. Future Work:

- Experiment with different PID parameters to optimize performance and robustness.
- Incorporate additional sensors, such as cameras or ultrasonic sensors, to enhance environmental perception.
- Expand the simulation to encompass more complex environments and scenarios to further evaluate the robot's capabilities.

Overall, the TurtleBot3 simulation in ROS2 Humble offers a versatile and accessible platform for exploring the exciting realm of robotics and autonomous navigation.

References:

- ROS2 Documentation: https://docs.ros.org/en/humble/index.html
- TurtleBot3 GitHub Repository: https://github.com/ROBOTIS-GIT/turtlebot3
- Gazebo Simulation: http://gazebosim.org/
- RViz Visualization Tool: https://docs.ros.org/en/humble/Tutorials/Rviz.html