PID Controller Mechanism

A PID (Proportional-Integral-Derivative) controller adjusts the robot's steering to minimize the error. The controller uses three components:

1. **Proportional (P) Term**: This term produces an output value that is proportional to the current error. The proportional gain, K_p, determines the reaction to the current error. A high K_p can cause the robot to react strongly to deviations.

$$P_{out} = K_p \times Error.$$

2. **Integral (I) Term**: This term accounts for past errors by accumulating the error over time. The integral gain, K_i , helps eliminate any residual steady-state error. However, too high K_i can lead to instability.

$$I_{out} = K_i \times \sum Error \times \Delta t.$$

3. **Derivative (D) Term**: This term predicts future error by calculating the rate of change of the error. The derivative gain, K_d, provides damping to the system, helping to reduce overshoot and oscillations.

$$D_{out} = K_d \times \frac{d(Error)}{dt}.$$

The total PID output is the sum of these three terms:

$$PID_{out} = P_{out} + I_{out} + D_{out}$$

Effects of K_p, K_i, and K_d on the Robot

- K_p **(Proportional Gain)**: Higher K_p results in a stronger correction based on the current error. If K_p is too high, the robot may oscillate around the line. If too low, the robot may respond sluggishly.
- K_i **(Integral Gain)**: Higher K_i helps eliminate steady-state error but can cause overshooting and instability if too high. It's useful for correcting systematic bias.
- K_d **(Derivative Gain)**: Higher K_d provides more damping, reducing overshoot and improving stability. If KdK_dKd is too high, it can cause excessive damping and slow response.

Implementation

In our robot, the PID controller takes the calculated error as input and adjusts the motor speeds accordingly to correct the path. The motor speed adjustments help steer the robot back towards the line, ensuring accurate and stable line following.

This PID control mechanism allows the robot to smoothly follow the line, correcting its path based on real-time sensor inputs and the calculated error. Adjusting K_p, K_i, and K_d Values are crucial for achieving optimal performance.