**Line Follower Robot Using PID Control**

**Team members:**

خالد عصام محمد مهران................................................................. 18010597

مازن طارق محمد فتح الله تاج الدين.................................................... 18011317

محمد خالد محمد فراج ................................................................... 18011454

محمد صديق محمد صديق................................................................ 18011496

محمد مصطفى علي عبد اللطيف........................................................ 18011618

محمد يوسف عبد الله محمد.............................................................. 18011650

مؤمن احمد عبد السلام محمد............................................................. 18011898

محمود أحمد محمد عبد الحميد........................................................... 18011660

محمود محمد ياسين عبد الجواد.......................................................... 18011715

Index:

a. PID Control Main concepts.

b. Detailed explanation of the implemented Algorithms.

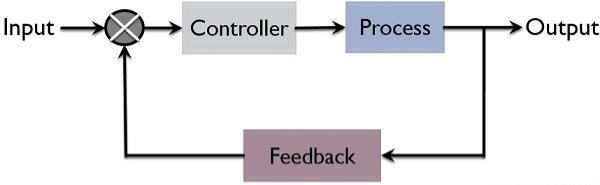
c. How did the group tune the PID parameters.

d. Proposed Improvements on each of the selected components.

1. PID Control Main concepts.

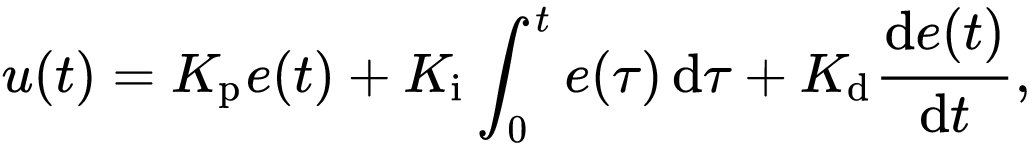
**Closed Loop Control System**

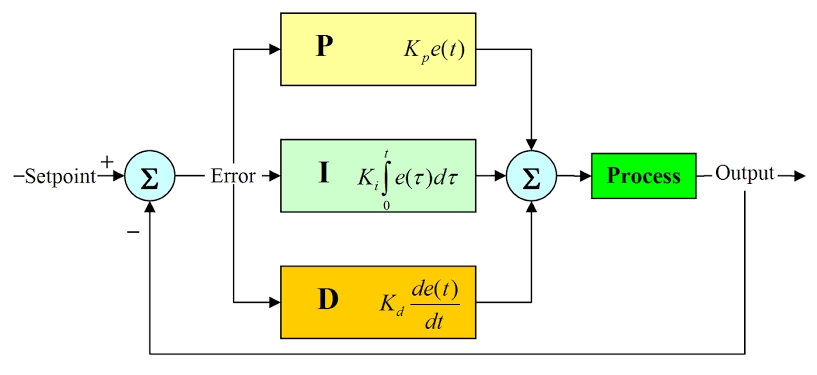
A closed loop system is a system in which the input signal depends on the output of the system. The goal of the system is to try to make its output reach a certain point called the set point. The output is fed back to the controller using a sensor and the error is calculated by subtracting the feedback signal from the set point. The input to the system then changes according to that error.



**PID Control**

PID control is a special type of closed loop control systems. PID stands for Proportional, Integral and Derivative. It is used to calculate the input signal to the system in a way that insures reaching the set point in the quickest way possible with the best performance of the system and with the least amount of oscillations and overshoots. The input signal is calculated using the following formula.

Where:

u(t) is the control signal.

is the error.

is the proportional gain.

is the integral gain.

is the derivative gain.

**Proportional Response**

The proportional component depends only on the error value. It is responsible for the speed of the system response, which means that by increasing the proportional gain the system tends to reach the set point quicker, but increasing it too much causes oscillations and may cause instability in the system.

**Integral Response**

Integral component sums the error of the system over time, which means that a small error value can make the component grow larger over time. Its aim is to eliminate system’s steady state error. Steady state error is the final difference between the process variable (system output) and set point.

**Derivative Response**

The derivative component depends on the rate of change of the error. Increasing the derivative gain causes the system to react more strongly to changes in the error which increases the speed of the overall control system response, however, most practical systems use small derivative gain because the derivative response is highly sensitive to noise and can lead to system instability.

**Tuning**

Tuning is the process of determining the values of the gains (). There are different methods such the trial-and-error method and the Ziegler-Nichols method. In trial-and-error method you first set the and values to zero, then increase until you reach the desired speed of system response, then increase slowly to eliminate steady state error with a little overshoot and finally increase value until the system reaches the set point quickly. In Ziegler-Nichols method you first set and values to zero, then increase until system starts to oscillate and record the values of critical gain () and period of oscillation (), the system parameters are calculated according to the type of control as shown:

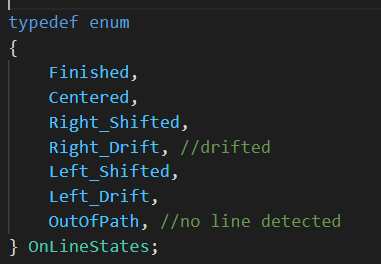
|  |  |  |  |
| --- | --- | --- | --- |
| Control |  |  |  |
| P | 0.5 | 0 | 0 |
| PI | 0.45 | /1.2 | 0 |
| PID | 0.6 | 0.5 | /8 |

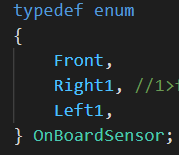
**b. Detailed explanation of the implemented Algorithms.**

Define states for the robot’s positions on the path.

If robot deviation angle was increased the error will increase which makes PID response for this deviation to cancel it by making robot turn in opposite direction of deviation.

Robot expected states on the path:

 Then naming sensors



Then update sensor readings and update state depending on readings

Calculate erorr related to state and pass it to PID controller.

