Laboratory Session 2: PID Control using Arduino

The objective of this session is to have a look at an experimental example of a PID controller, using Arduino.

We want to control the lighting level of a room using the PID Library which can be used with Arduino.

1. Open-loop system.

Download the code: "LDRtest.ino". Before starting applying a PID control, the main objective is to know the behavior of the plant. Draw the block diagram that we will analyze. Identify which element is the INPUT and which one is the OUTPUT.

To develop an open-loop system, we have to develop a circuit using an LED. Look for a resistor of 100 Ω, and connect them to the LED and to ground. Whether you do not have a 100 Ω resistor, try to achieve this value (± 10%) by connecting different resistors in parallel.

- Draw the schematic of the circuit (ACTUATOR PART).

Now, think about the objective of the LDR in this system. Which is its function?

Draw the block diagram considering also the LDR.

Develop the following circuit in your breadboard: Connect the LDR to 5V, to your Arduino (think whether it is going to be an INPUT or an OUTPUT) and place a 10 kΩ to the LDR and to ground.

Open the file "LDRtest.ino. You also have to develop the code where you can write a value from 0 to 255 to regulate the output (LED lighting level). Please answer these questions:

- Notice that the ADC value from the LDR system can achieve values between 0 and 1023. However, we want to scale it between 0 and 255. Write an analog read function considering this. How does the system behave? Is it possible to achieve all the values between 0 and 255? Which is the maximum value that we can achieve?

- Does the LED behavior change if you put a box on it?

Now, to achieve a better response, we will implement a low pass filter in order to attenuate the signal noises and then be able to achieve a better response. The filter is based on an RC. R = 10 Ω and C = 100 µF. We need to connect this filter before sending this value to the Arduino input pin.

- Draw the schematic of the circuit. Does the system have a better response?

2. Closed-loop system. One LED

As we have seen in the videos, a closed-loop system is better to achieve the desired behavior of the system. We will apply this to our plant, implementing a PID Controller. In the previous exercise, we have implemented an open loop, since the LDR is acting as a sensor to tell us whether the system is performing well or not. Despite this, we are not changing the behavior of the plant to achieve the setpoint. We will do that closing the loop by implementing a PID control.

- Closed Loop & PID Controller code.
    1. Download and install the PID Library.
    2. Please download and open the file: ldr_closed_loop_PID_incomplete.ino
    3. Fill in the gaps
    4. Compile and run the program. Open the serial monitor and write a SetPoint Value and observe how the system behaves.

- Try to change the values for Kp, Ki, and Kd. Does the system behave better than in case no. 1? Try different values for Kp, Ki and Kd and also different setpoints.

5. Using the serial monitor, try to apply Z-N to find the values for Kp, Ki, and Kd. Are you applying a P, a PI or a PID controller?

- Which is the maximum value that we can achieve now?

3. A closed-loop system using more than one LED.

It might happen that using only one LED is not possible to achieve high lighting values. For this reason, it may be useful to connect more than one LED.

- How many LEDs have you connected to? Can we achieve higher values then?