EE324: Experiment No.1
DC MOTOR POSITION CONTROL

Batch No:3
Mudit Goyal 200070045
Mudavath Vishnuvardhan-200070044
Narne Avinash Chowdary-200070047

October 7, 2022

1 Overview of the experiment

1.1 Aim of the experiment

Design and implement a PID controller using Arduino Mega to rotate the motor by a fixed angle.

1.2 Method

Use L293D IC and aurdino to rotate the dc motor by an angle of 180 degrees from any given point while also adhering to the design specifications of 0.5 second rise time, 1 second's 2 % settling time and 10% overshoot.

2 Control Algorithm

2.1 PID Control Application

Output of PID controller is calculated in time domain as follows:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt}$$

The variable (e) represents the tracking error, the difference between the desired angle(r) and the current angle(y). This error signal (e) is fed to the PID controller, and the controller computes both the derivative and the integral of this error signal with respect to time. The control signal (u) is fed to the pwm signal and is equal to the proportional gain (K_p) times the magnitude of the error plus the integral gain (K_i) times the integral of the error plus the derivative gain (K_d) times the derivative of the error.
$\mathrm{K}_p = proportional gain$
$K_i = Integral gain$
$K_d = differential gain$

2.2 Characteristics of P,I,D

- P-controller as term suggests follow the error and provides stability with good speed of response.This causes for a given level of error the closed-loop system to react more quickly, but can also increase the overshoot. Secondary effect of K_p is of changing the steady-state error.

- I-controller is mainly used to reduce the steady state error. By decreasing the integral gain Ki, the speed of the response is increased or in other words it can make the system more sluggish (and oscillatory) since when the error signal changes sign, it take a while for the integrator to unwind.

- D-controller increases the speed of the response because it anticipates the future behavior of the error. The more rapidly D-controller responds to the changes in the process variable, if the derivative term is large (K_d).

Response	Rise Time	Over Shoot	Settling Time	Steady state error
K_P	Decrease	Increase	-	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_D	-	Decrease	Decrease	-

Table 1: PID

3 Approach

In this experiment, we programmed the arduino to send PWM control signal to which was used to control the motor speed, depending on the input recieved. To start with, we set the respective pins on the arduino to input and output. To get the PWM signal we took the input as the pot reading after passing it through a 10-bit ADC and used the PID controller to calculate the power signals to be sent which should lie between 0-255 as it will be passed through a 8-bit DAC. Then through trial and error we changed the K_P, K_I, K_D values to match the design specifications. Though there were certain challenges that we faced in this approach and how we solved those complications have been discussed in the next section.

4 Challenges faced and their Solutions

- First challenge we encountered was to figure out how to control the speed of motor and then after much debate and taking a little help from one of the TA's we finally understood that we have to send PWM signal to control the voltage levels of the motor which in turn will control the speed of motor. We then wrote a code which would use PID to determine the speed and accordingly the direction in which the motor has to rotate.

- Second challenge we faced was that the motor did not behave the way we wanted it for some range of angles. This was actually due to the non-linear region of motor. This linear region exists because of the circular nature and the air gap that exists inside the pot. In this region we will get random values and we won't get the desired behaviour of the motor so this is the region which we had to avoid. So, we first figured out that particular range by rotating the motor by hand and marking down the values. Then we wrote our code in certain way that it will never cross that region by creating required offsets and setting the rotation of motor accordingly.

5 Experiment results

This section contains the data obtained from the experiment and using that data a graph has been plotted and various results have been derived from the observation and data.

5.1 Graph

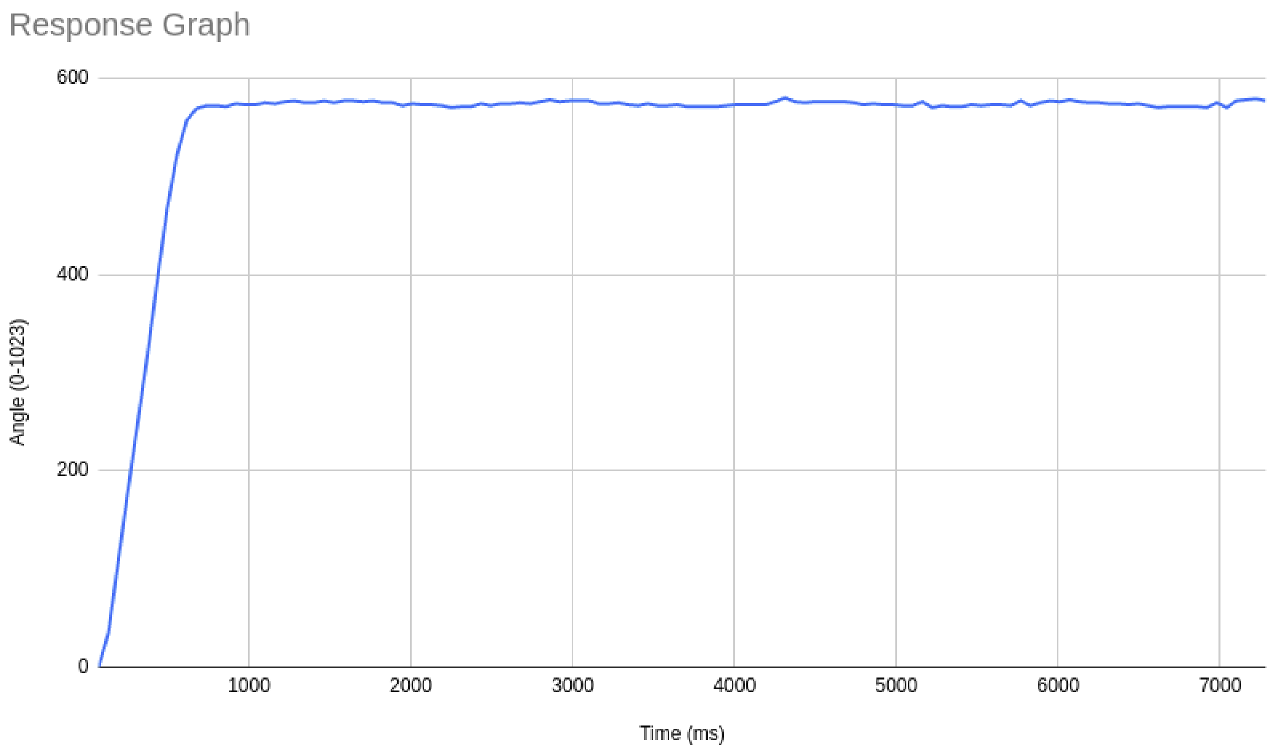

Figure 1: Time Response

5.2 Data

Angle (0-1023)	Time (ms)	Angle (0-1023)	Time (ms)	Angle (0-1023)	Time (ms)
0	70	577	1645	574	3221
35	131	576	1705	575	3281
107	191	577	1767	573	3342
179	251	575	1827	572	3403
249	312	575	1888	574	3464
319	373	572	1948	572	3524
395	434	574	2009	572	3585
467	494	573	2070	573	3645
522	555	573	2130	571	3706
557	615	572	2191	571	3767
569	675	570	2251	571	3827
572	737	571	2312	571	3888
572	797	571	2372	572	3948
571	858	574	2434	573	4009
574	918	572	2494	573	4070
573	978	574	2554	573	4130
573	1039	574	2615	573	4191
575	1099	575	2675	576	4251
574	1161	574	2737	580	4312
576	1221	576	2797	576	4373
577	1282	578	2857	575	4433
575	1342	576	2918	576	4494
575	1402	577	2978	576	4554
577	1464	577	3040	576	4615
575	1524	577	3100	576	4676
577	1585	574	3161	575	4737

On the next page the values of K_P, K_I, K_D and various design parameters have been tabulated.

5.3 Observations and Inference

From the above values,

K_P	K_I	K_D
1.4	0.0001	0.4

Table 2: Results

2% Settling time	Rise time	% Overshoot
$0.675s$	$0.405s$	0.517%

Table 3: Observations