Control Systems Lab - Experiment No. 1: DC Motor Position Control

Group No.: 11
Members:
Sanjay Meena(22b3978)
Devtanu Barman(22b3904)
Prince Chouhan(22b3970)

August 26, 2024

1 Aim

Design and implement a PID position controller using Arduino Mega.

2 Objectives

- Rotate the DC motor by an angle of 180 degrees from any given point.
- Ensure the task meets the design specifications: 0.5-second rise time, 1-second settling time, and 10% overshoot.

3 Prerequisites

- Arduino programming
- PID control logic

4 Materials and Equipment Required

- DC Motor setup
- Arduino Mega

- A-B cable
- Power supply
- L293D IC
- Jumper wires
- Single-stranded wires
- Breadboard
- Screwdriver
- Wire stripper

5 Block Diagram

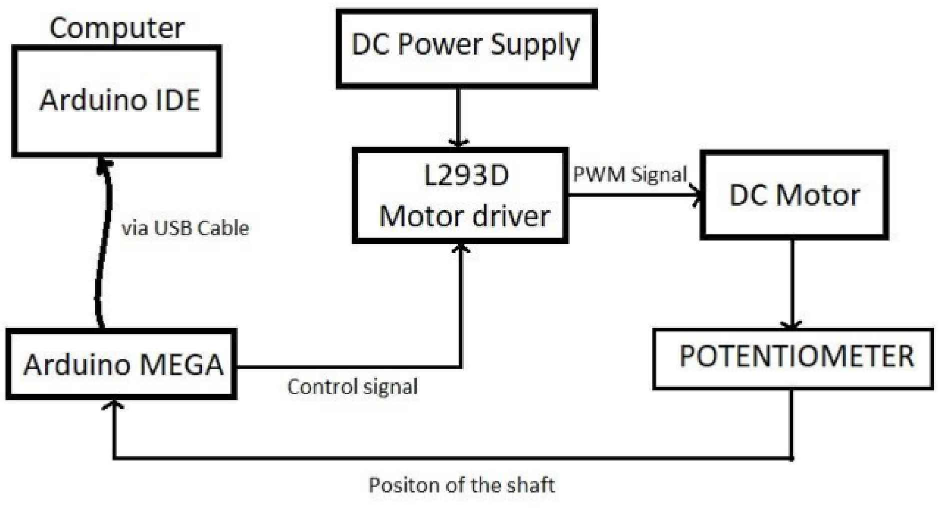
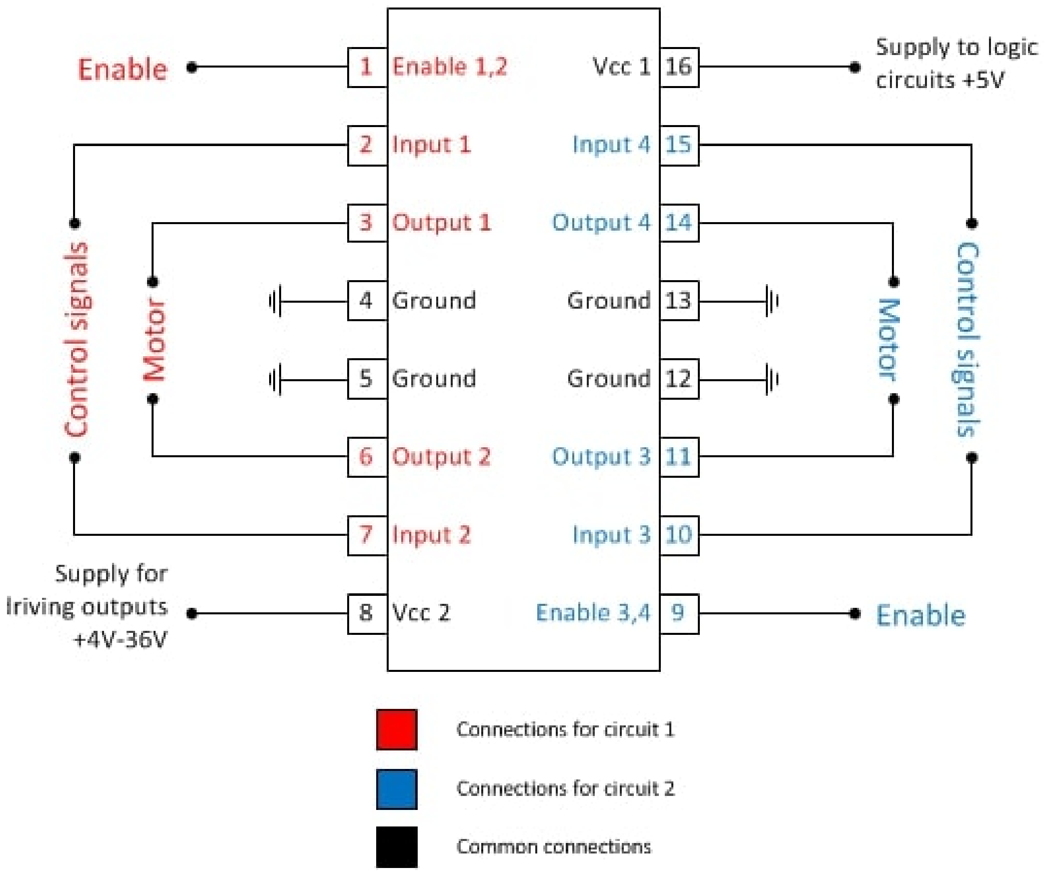


Fig: Block diagram for DC motor position control

6 Pin Diagram of Motor Driver (L293D)



7 Arduino Code

```
const int motorPin1 = 10;

const int motorPin2 = 9 ;

const int inputPin = A0;

const int ENABLE = 8;

int t = 0;

int prevTime = 0;
int dt = 0;

float Kp = 20.0;

float Kd = 15.0;

float Ki = 0.0005;

int midPoint = 535.74;

int maxOutput = 1023; // to prevent overshoot

int theta = 0;

int final = 0;

int prevError = 0;

int E = 0; // sum of errors

int e = 0;

void setup() {

  Serial.begin(9600);

  pinMode(inputPin, INPUT);

  pinMode(motorPin1, OUTPUT);

  pinMode(motorPin2, OUTPUT);

  // Initial sensor state

  theta = analogRead(inputPin);

  // Determining destination

  if (theta > midPoint) {

    final = theta - midPoint;
  } else {

    final = theta + midPoint;

  }

  prevError = final - theta;

  E = prevError;  // Initialize E with the initial error

}

void loop() {

  // Calculate time difference

  t = millis();

  dt = t - prevTime;

  theta = analogRead(inputPin);

  e = final - theta;

  E += e * dt;  // Multiply error by time difference for integral term

  int de = e - prevError;

  prevError = e;

  // Implement PID

  //int u = (int)abs(Kp * e + Kd * (de / dt) + Ki * E);

  int u = (int)abs(Kp*e + Kd*de + Ki*E); // (If time dosent work)

  // u = Kp*e + Kd*de/dt + integral edt

  u = min(u, maxOutput);
  if (e > 0) { // move clockwise

    analogWrite(motorPin1, u);

    analogWrite(motorPin2, 0);

  } else { // move anticlockwise

    analogWrite(motorPin2, u);

    analogWrite(motorPin1, 0);

  }

  prevTime = t;
Serial.print(millis()); // Time in milliseconds
  Serial.print("         ");
  Serial.println(theta);
  }
```

8 Results

The corresponding response graph of the DC motor under PID control is shown in Figure 1.

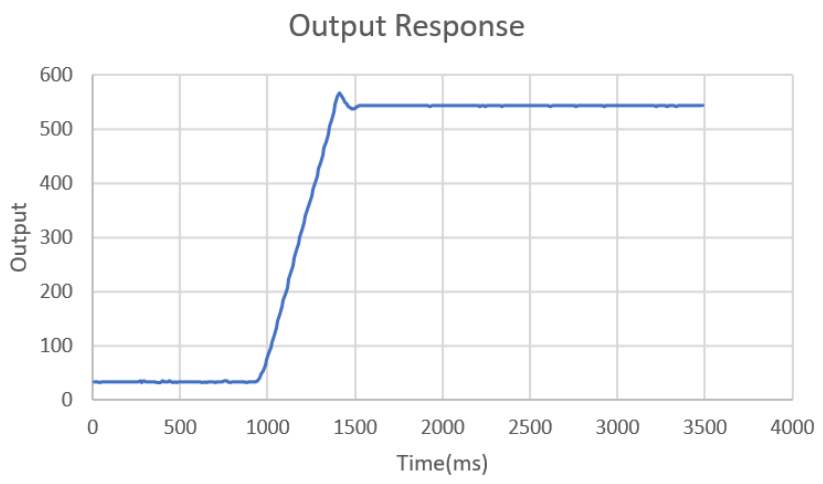


Figure 1: Output Response

9 Tabulation

Rise Time	0.3690s
% Overshoot	4.24%
2% Settling Time	1.3800s

Table 1: Calculated Quantities from Output Response

10 Calculations

From output response graph, we can see that

Steady State Value = 543

Maximum Output Value = 566

So, % Overshoot = [(566-543)/543] x 100% = 4.24%

So, Rise time = (Time to reach 90% of 543 - Time to reach 10% of 543) = (1344 - 975) ms = 369 ms = 0.369s

So, 2% Settling time = Time to reach 98% of 543 = 1.38s

11 Challenges faced in the Experiment

The main challenge was to find the right values for the proportional (Kp), integral (Ki), and derivative (Kd) gains as per our requirement. The arduino coding also involved some difficulty for us as we were not that familiar with the arduino programming.

12 Conclusion

However we overcame the challenges and the PID controller successfully rotated the DC motor to the desired position with a rise time of 0.369 seconds, a settling time of 1.38 seconds and an overshoot 4.24%, meeting the design specifications.

Thank you!