

Hardware in Loop Technique to Control the Speed of DC Motor Using PID controller

Kavya Hiremath¹, C M Maheshan²

¹PG Scholar (Control and Instrumentation), Department of Electrical Engineering, University Visvesvaraya College of Engineering, Bengaluru, Karnataka, India

²Associate Professor, Department of Electrical engineering, University Visvesvaraya College of Engineering, Bengaluru, Karnataka, India

ABSTRACT

In robotics and other industrial applications, a DC motor's speed control is crucial. The rotational speed of a DC motor may be controlled using a variety of techniques. The real-time rotational speed control of a DC motor is discussed in this study. A controller is created with the aid of a Matlab program to regulate the rotational speed of a DC motor in real time and an Arduino Mega Board is used to connect the entire hardware to the laptop through a USB port. The low-cost hardware is made up of an Arduino mega board for laptop interface, an L293D IC, second motor acting as tacho-generator linked with the driving motor. The output of the second motor fed into the microcontroller. The microcontroller receives an input from the user. This input defines the desired condition of the System. The microcontroller also receives an input from the motor. The motor speed information of which is fed back to the input. This input can also be called the negative feedback. The microprocessor, based on its Simulink model, performs various calculations, and gives an output to the actuator. The actuator, based on the output, controls plant to try to maintain those conditions using PID controller. Driving a motor where the motor driver is the actuator, and the motor is the plant. The motor thus rotates at a given speed.

Keywords: Real time hardware in Loop control, PID Control, Arduino mega 2560, Serial communication, Pulse Width Modulation.

Date of Submission: 02-02-2023

Date of acceptance: 14-02-2023

I. INTRODUCTION

1.1 OVERVIEW

An essential component of robotics and other industrial applications is the DC motor. The terminal voltage regulated approach may be used to regulate the dc motor's rotational speed. The provided input voltage affects the rotational speed of the DC motor. By delivering a constant input voltage, the speed of the motor is modified by intermittently switching the supply voltage on and off, so that the motor only detects the average voltage effect[1] and not the switching operation. Using PWM technology, the speed of the DC motor may be changed.

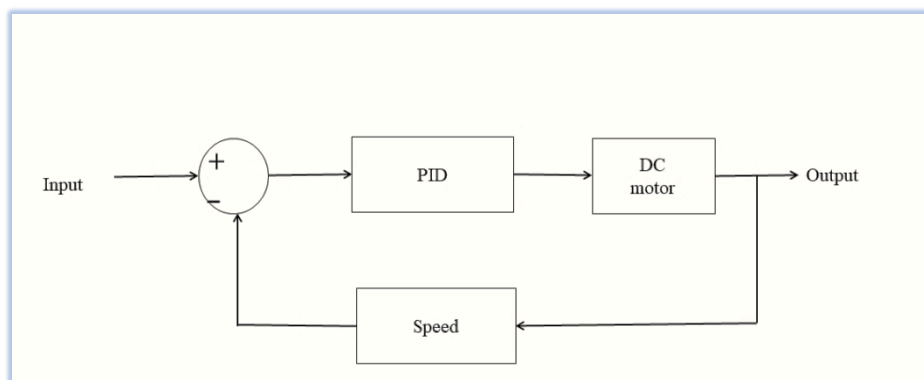


Figure 1.1: shows the hardware system's block diagram.

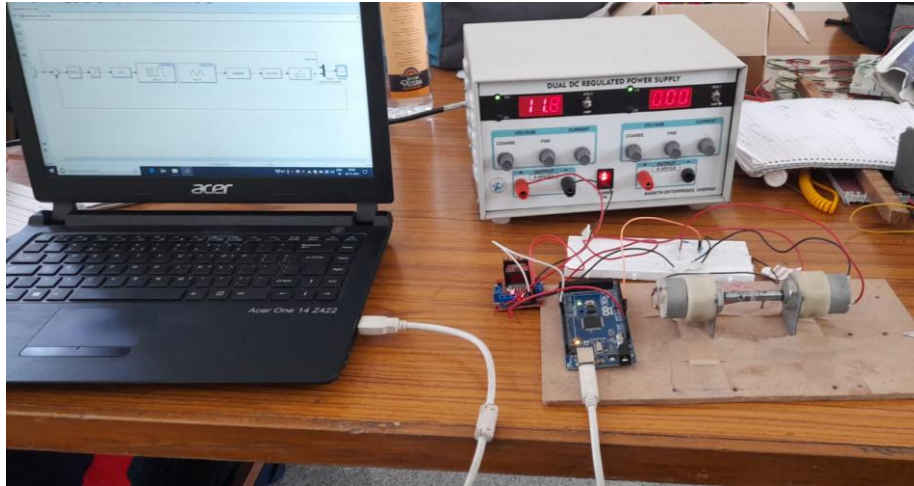


Figure 1. 2: Actual Hardware

1.2 SYSTEM DESCRIPTION

Controlling the terminal voltage of the DC motor is the simplest way to alter its rate of rotation. The maximum motor speed increases with voltage. Utilizing PWM method, the voltage is controlled. The PWM technique modulates the motor current by turning on and off the operational power of the motors. The motor's speed is determined by the ratio of "on" to "off" time. The DC motor's speed depends on the terminal voltage. When the voltage is low, the motor's speed is lower, and when it is higher than what is indicated in the motor's datasheet, the motor's speed can reach its maximum rotating speed. Using PWM approaches, the terminal voltage is managed. The System's primary elements are.

1.3 Arduino Mega Board

The Arduino Mega is a general-purpose, open-architecture microcontroller board based on the ATmega2560, and it has 54 digital input/output pins, 14 of which may be used for PWM output and 16 for analogue input, totaling 54. A 16 Mhz crystal oscillator, a D-type USB connection, a power cable, and 4 UART (Hardware Serial Ports) are also included.

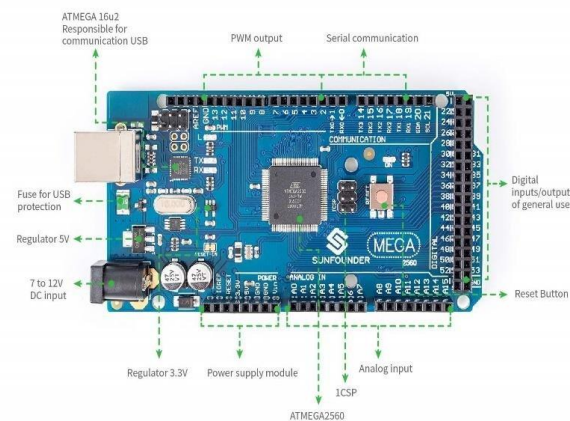


Figure 1.3: Arduino Mega 2560 Board

The Arduino board is utilised in this project to enable hardware-to-laptop interface for Matlab.

1.4 Driver IC (L293D)

The 16-pin motor driver IC L293D may be used to separately operate two different motors. This IC may be used to regulate the motor's rotational direction and speed.

This IC has two power pins, one of which (5 volts) has to be linked to a +5 volt source in order for the IC to function. Depending on the motor's specifications, a different power pin is utilised to run the motor.

1.5 DC Motor

One type of electric machine that converts electrical energy into mechanical energy is the DC motor. In this project, two 12 Volt 500 RPM DC motors are utilised, one as a generator and the other as a motor. Utilizing variable terminal supply voltage, the motor's rotations per minute (RPM) speed is managed. A 12 volt DC motor

is affordable, portable, and strong enough to be employed in a variety of applications, including toys, automotive mechanics, and tolls.

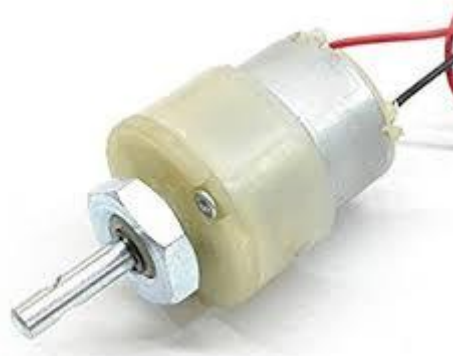


Figure 1.5: 12 Volt, 500 RPM DC motor

II. PLANTSIMULATION

According to the block design in the picture, the hardware interface is completed using the MATLAB Simulink toolbox.

This model record the open loop step response data from Arduino

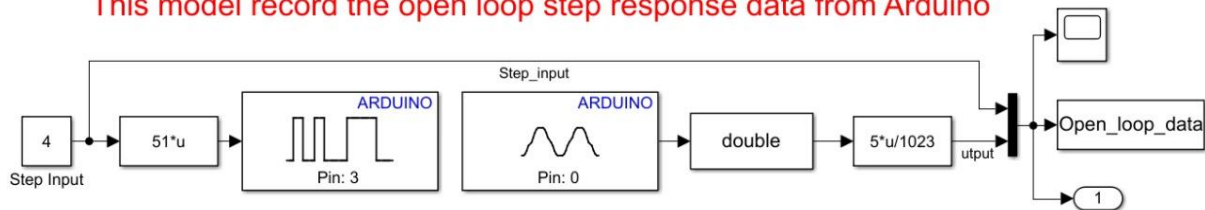


Figure 2.1: displays the open loop Simulink model of the plant.

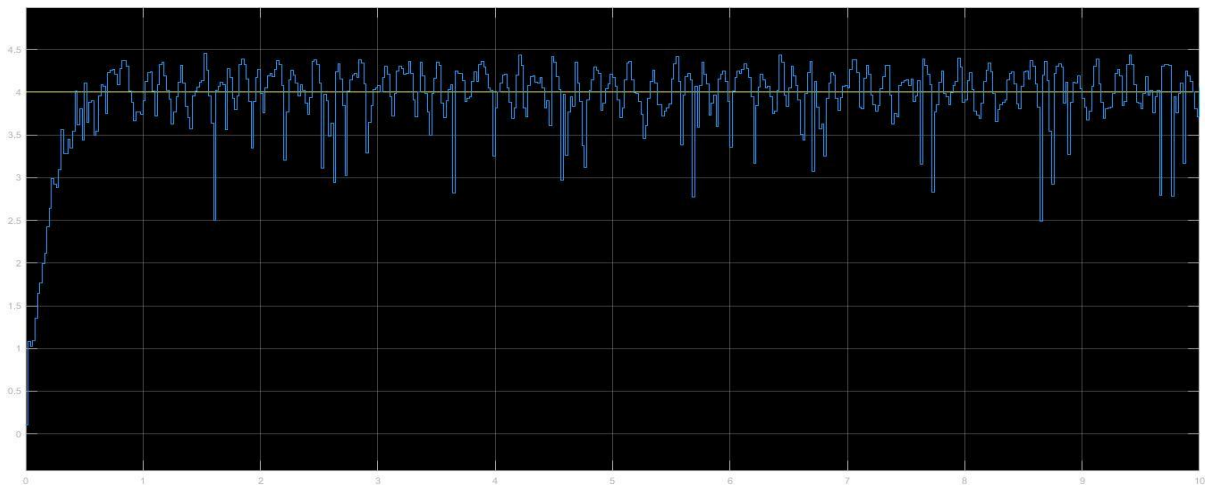


Figure 2.2 displays the system's output response..

III. Mathematical model and Transfer function of DC motor

The DC motor is a widely used actuator in control systems. It can provide both rotational and translational motion when paired with wheels, drums, and cables. The following diagram illustrates the rotor's free-body diagram and the armature's electrical equivalent circuit.

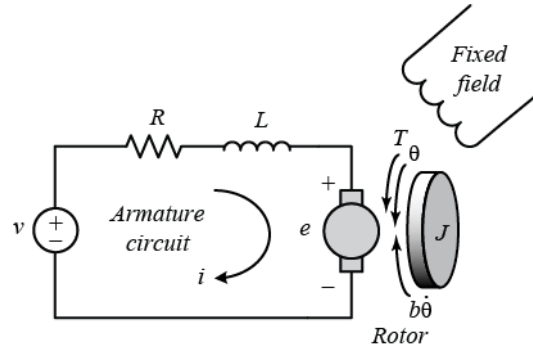


Figure 3.1.1:Free body diagram of DC motor

For this example, we will assume that the input of the system is the voltage source V applied to the motor's armature, while the output is the rotational speed of the shaft θ . The rotor and shaft are assumed to be rigid. We further assume a viscous friction model, that is, the friction torque is proportional to shaft angular velocity.

The physical parameters for our example are:

- (J) moment of inertia of the rotor 0.01 kg.m^2
- (b) motor viscous friction constant 0.1 N.m.s
- (Ke) electromotive force constant 0.01 V/rad/sec
- (Kt) motor torque constant 0.01 N.m/Amp
- (R) electric resistance 1 Ohm
- (L) electric inductance 0.5 H

In general, the torque generated by a DC motor is proportional to the armature current and the strength of the magnetic field. In this example we will assume that the magnetic field is constant and, therefore, that the motor torque is proportional to only the armature current i by a constant factor K_t as shown in the equation below. This is referred to as an armature-controlled motor.

$$T = K_t i \quad (3.1)$$

The back emf, e , is proportional to the angular velocity of the shaft by a constant factor K_e .

$$e = K_e \dot{\theta} \quad (3.2)$$

In SI units, the motor torque and back emf constants are equal, that is, $K_e = K_t$; therefore, we will use K to represent both the motor torque constant and the back emf constant.

From the figure above, we can derive the following governing equations based on Newton's 2nd law and Kirchhoff's voltage law.

$$J\ddot{\theta} + b\dot{\theta} = Ki \quad (3.3)$$

$$L \frac{di}{dt} + Ri = V - K\dot{\theta} \quad (3.4)$$

• Transfer Function

Applying the Laplace transform, the above modeling equations can be expressed in terms of the Laplace variable s .

$$s(Js + b)\Theta(s) = KI(s) \quad (3.5)$$

$$(Ls + R)I(s) = V(s) - K_s\Theta(s) \quad (3.6)$$

We arrive at the following open-loop transfer function by eliminating $I(s)$ between the two above equations, where the rotational speed is considered the output and the armature voltage is considered the input.

$$P(s) = \frac{\dot{\theta}(s)}{v(s)} = K / ((Js + b)(Ls + R) + K)^2 \quad (3.7)$$

Using Matlab's System Identification toolbox, the transfer function of the hardware is calculated.
Transfer Function

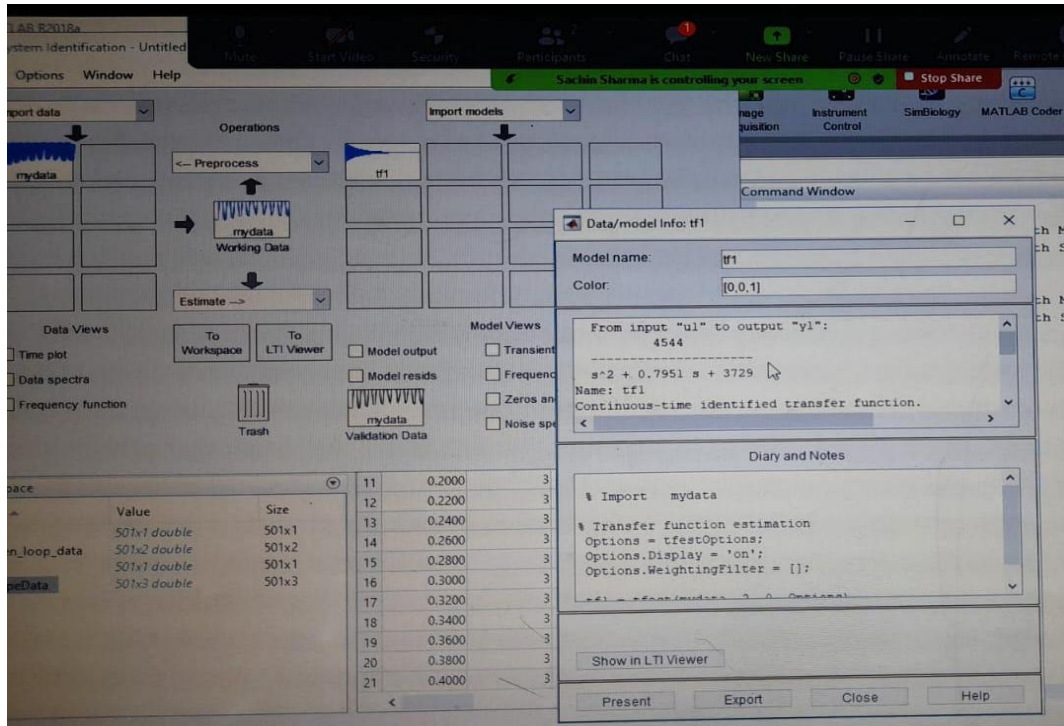


Figure 3.1.: System Identification panel view

From input “u1” to output “y1”

$$T.F = \frac{4544}{s^2 + 0.7951s + 3729}$$

3.3 PID Controller Design

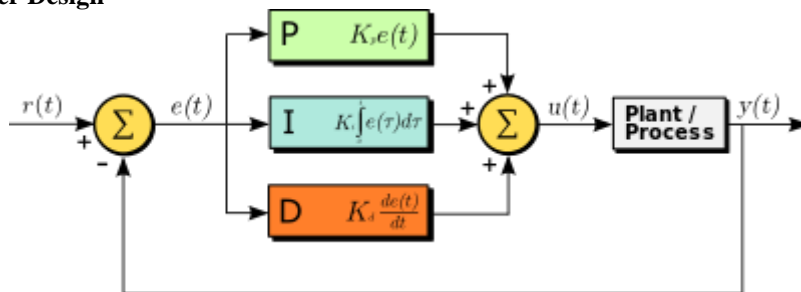


Figure 3.2 PID controller

Because of its simplicity and affordability, PID controllers are still the most popular controllers in industries. The proportional Gain Constant (K_p), Integral Gain Constant (K_i), and Derivative Gain Constant (K_d) determine the controller's output (K_d). The PID controller's mathematical equation is shown below.

$$u(t) = K_p * e(t) + K_i * \int e(t) dt + K_d * \frac{de(t)}{dt} \quad (3.3.1)$$

$$u(s) = (K_p + K_i * 1/s + K_d * s) * E(s) \quad (3.3.2)$$

Using the auto-tuning function of the PID Controller on Matlab, the tuning parameter of the PID Controller (K_p , K_i , K_d) is optimised.

IV. CLOSE LOOP SIMULINK MODAL WITH OUTPUT RESPONSE

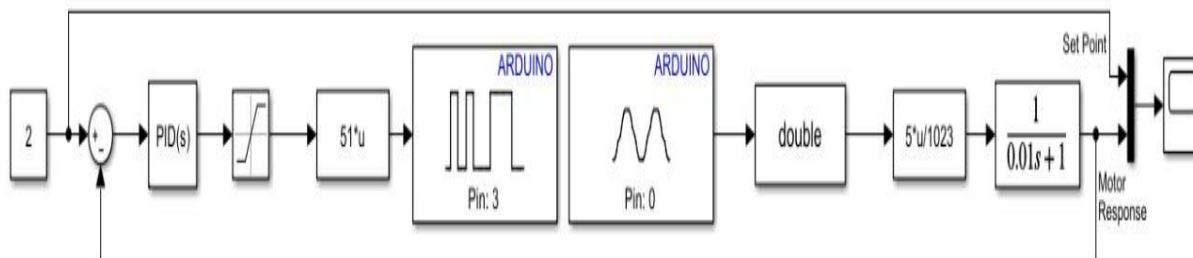


Figure 4.1: Close loop Response of the system

The TLBO algorithm is a teaching-learning process inspired algorithm and is based on the effect of influence of a teacher on the output of learners in a class. The algorithm describes two basic modes of learning:

- (i) Through teacher (known as teacher phase)
- (ii) Through interaction with the other learners (known as learner phase).

In this optimization algorithm a group of learners is considered as population and different subjects offered to the learners are considered as different design variables of the optimization problem and a learner's result is analogous to the 'fitness' value of the optimization problem. The best solution in the entire population is considered as the teacher. The design variables are actually the parameters involved in the objective function of the given optimization problem and the best solution is the best value of the objective function. Teaching-learning-based optimization (TLBO) is a population-based algorithm which simulates the teaching-learning process of the class room. This algorithm requires only the common control parameters such as the population size and the number of generations and does not require any algorithm-specific control parameters. Locate global minima for situations with significant nonlinearity. An approach for addressing both restricted and unconstrained optimization problems based on a natural selection process that resembles biological evolution is known as a genetic algorithm (GA). An individual population of solutions is repeatedly modified by the algorithm. The genetic algorithm chooses individuals at random from the existing population at each stage, using them as parents to create the offspring for the following generation. The population "evolves" toward the best option over the course of subsequent generations.

the evolutionary method to resolve issues when the objective function is discontinuous, non differentiable, stochastic, or highly nonlinear that are not well suited for ordinary optimization procedures.

```

1  % this is objective function for online optimization
2  function [J] = cost(x)
3
4  -  Kp = x(1);
5  -  Ki = x(2);
6  -  Kd = x(3);
7  -  N = x(4);
8
9  % Choose solver and set model workspace to this function
10 - opt = simset('solver','ode45','SrcWorkspace','Current');
11
12 - try
13   % yout is outputs of all out ports applied
14   [tout,xout,yout] = sim('GA_opt_model_s_domain_PID',[0 50],opt);
15
16   J = yout(end); %objective function
17
18 catch
19   J = 1000000000;
20
21 end
    
```

Figure 4.2 : Online optimization Script window

Here (figure 4.2) is a basic script in Scratch to find the values of KP, KI, and KD in a PID control algorithm using an optimization function:

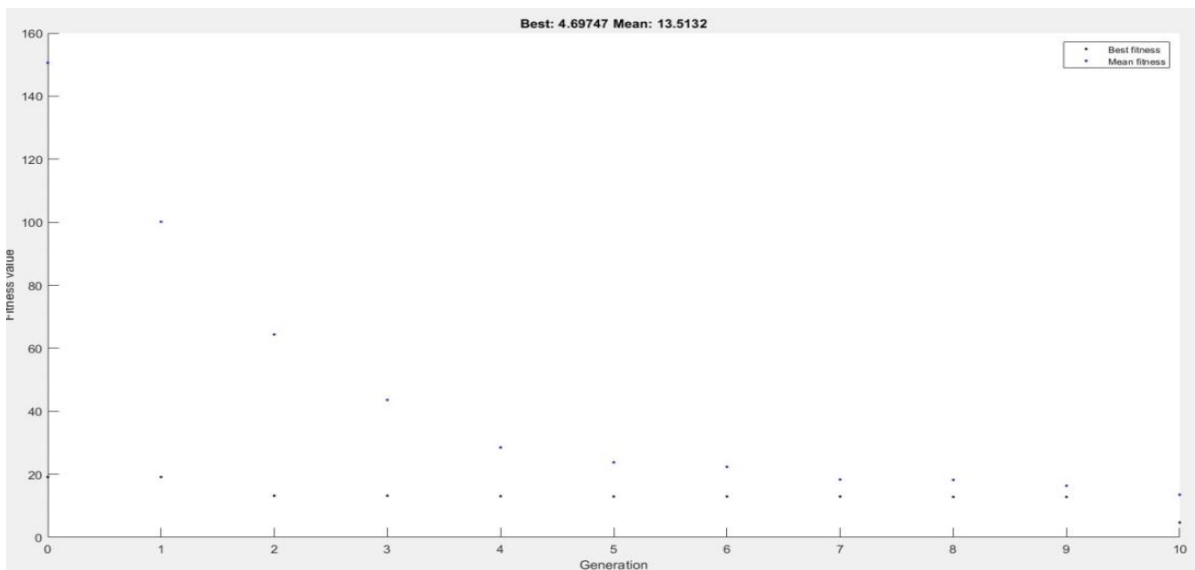


Figure 4.3 : Optimization graph for Ten Iterations

Population fitness graph represents the evolution of population fitness over generations during the optimization process of a problem. TLBO (Teaching-Learning-based Optimization) is a metaheuristic optimization algorithm that uses teaching-learning mechanism to optimize the solution of a problem. The population fitness graph generated by TLBO method shows the average fitness value of the population at each generation. The fitness value represents the quality of the solution, with higher values indicating a better solution. Over generations, the average fitness value of the population is expected to increase, which means the solutions are getting better. The population fitness graph can be used to evaluate the performance of the TLBO algorithm and to determine the stopping criteria, i.e., when to stop the optimization process. The shape of the graph can help to identify the convergence behavior of the algorithm and the presence of local optima as show in fig . 4.3.

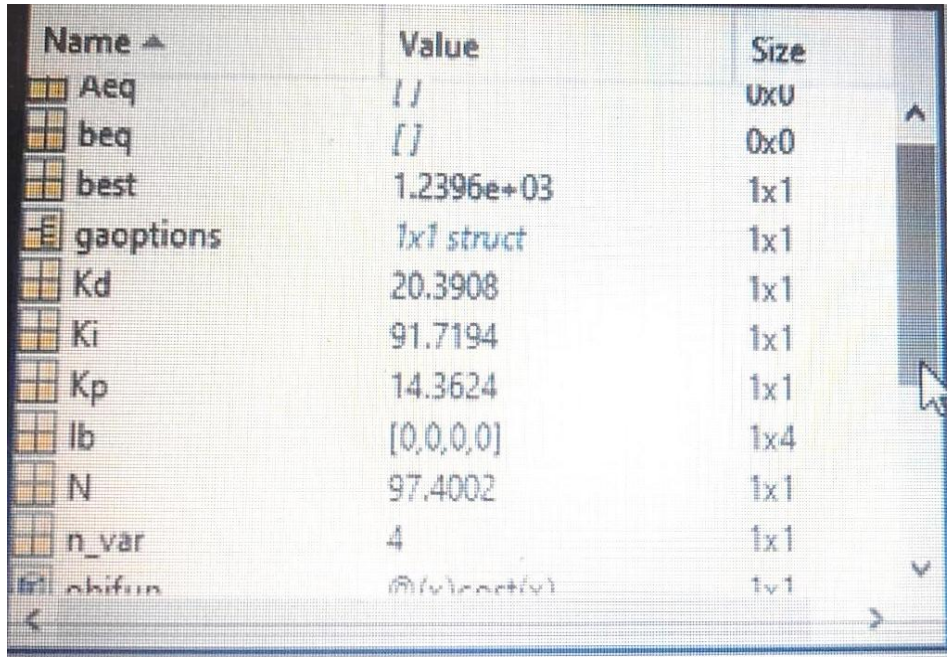


Figure 4.4 : Online Optimization values of Kp,Ki and Kd

In TLBO, the KP, KI, and KD values are treated as variables and are optimized using a teaching-learning process. A teacher agent generates new solutions for the PID controller parameters and a learner agent evaluates the performance of the system with these parameters. The learner then provides feedback to the teacher to improve the next iteration of solutions. The TLBO algorithm iteratively updates the values of KP, KI, and KD until the performance criteria (such as stability, overshoot, settling time, etc.) are met. The TLBO algorithm has been shown (figure 4.8) to be effective in finding optimal PID parameters for control systems, especially in cases where the conventional tuning methods are not applicable or ineffective.

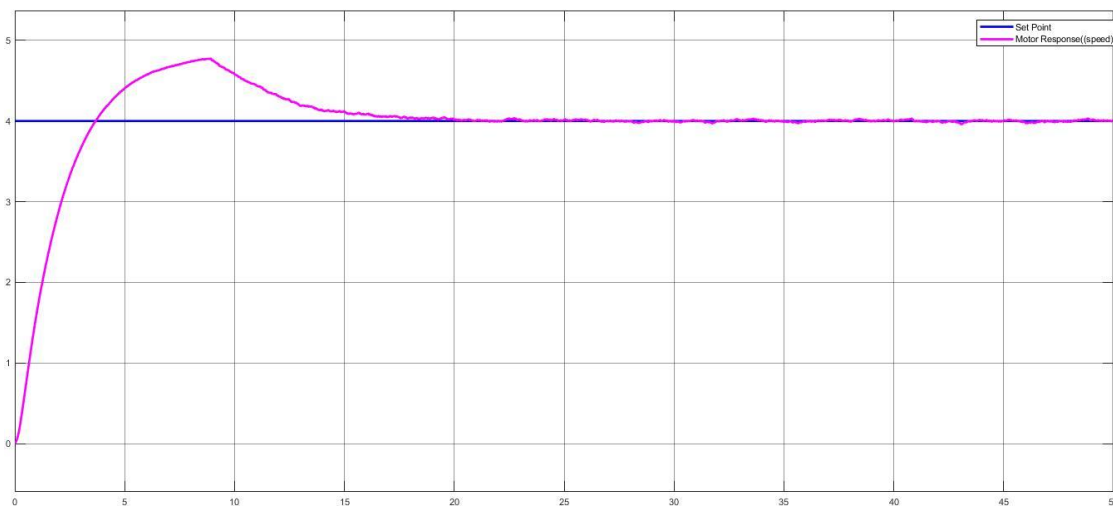


Figure4.5: Tuned output of DC Motor after applying PID with filter

DC motor rotational speed is managed using a PID controller. In figure 4.5, the system's output response is displayed. The step input is represented by the blue line, and the system output is represented by the yellow line. The system's noise causes the system to oscillate about its steady state

Table 4.1 :Output Result

Sl. no	Step input	RPM
1	2	90

2	3	175
3	4	276

The output speed measurement table for a DC motor typically displays the speed of the motor in rotations per minute (RPM) as a function of time with step input voltage.

V.RESULTS

The goal of this project was to create inexpensive hardware that could be used to test various controllers created using Matlab Simulink. This hardware is used to test the PID controller, which was created in Matlab. Figure 9 illustrates that the PID controller's performance is good.

Additionally, alternative Matlab-designed controllers may be tested using the same hardware. To verify the performance of the controller with the hardware, another optimised controller block can be used in place of the PID controller .

VI. CONCLUSION

The real-time hardware in loop control of a DC motor configuration utilising a PID controller and an Arduino Mega Board is demonstrated in this work. The major goal of this project is to establish a low-cost platform for improved comprehension, application, and real-time testing of various controllers created in businesses and laboratories for educational purposes.

REFERENCES

- [1]. Gavran, M., Fruk, M., & Vujisić, G. (2017, May). PI controller for DC motor speed realized with Arduino and Simulink. In 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 1557- 1561). IEEE.
- [2]. Thirupathi Allam, Matla Raju, S. Sundeep Kumar, "Design of PID controller for DC Motor Speed Control Using Arduino Microcontroller", (IRJET 2016) Volume: 03, Issue: 09.
- [3]. Hardik S Jain, Akshat Palak, Sandesh Agrawal, Krishnam Tibrewal, Malaya kumar Hota, "DC motor Speed Control using PID Controller, IR Sensor and PWM Hysteresis" (IJITEE), ISSN: 2278-3075, Volume8, Issue-6, April 2019.
- [4]. Shuvra Prokash Biswas, Md. Kamal Hosain, Md. Waliur Rahman (2017, IEEE) " Real – Time Arduino Based Simulator Enabled Hardware-in-the-loop Electric DC Machine Drive System
- [5]. Barber, Ramón, Mario De La Horra Köllmer, and Jonathan Crespo. "Control Practices Using Simulink with Arduino As Low Cost Hardware." In ACE, pp. 250-255. 2013.
- [6]. Nawi Berahim, Sulaini Besar, Mohd Zani, Abdul Rahim, Shamsul Aizam Zulkifli, Zairi Ismael Rizman, "PID Voltage control for DC Motor Using MATLAB Simulink and Arduino Microcontroller", ISSN: 2090- 4274, J. Appl. Environ. Biol. Sci., 5(9) 166-173, 2015.
- [7]. Hirdesh Kr.Saini, Shahid Firoz, Ashutosh Pandey (2017, April) "Arduino Based DC Motor Speed Control". IJRDO – Journal of Electrical And Electronics Engineering, Volume – 3, ISSN: 2456-6055.
- [8]. Asha, K. R., Tasleem, P. S., Kumar, A. R., Swamy, S. M., & Rekha, K. R. (2017, March). Real Time Speed Control of a DC Motor by Temperature Variation Using LabVIEW and Arduino. In 2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT) (pp. 72-75). IEEE.
- [9]. Hat, M., Ibrahim, K. K., Salam, B., Mohd, T. A. T., & Hassan, M. K. (2015). Model Based Design of pid controller for BLDC motor with implementation of embedded arduino mega controller Instrumentation Engineering, 3(9), 12162-12167.
- [10]. Shetti, P. R., & Mangave, A. G. (2014). DC Motor Speed Control with Feedback Monitor Based on C# Application. International Journal of Research in Engineering and Technology, 3(3), 398-401..
- [11]. Vikhe, P., Punjabi, N., & Kadu, C. (2014). Real time DC motor speed control using PID controller in LabVIEW. International Journal of Advanced Research in Electrical, Electronics and