International Journal of Electrical Engineering & Emerging Technology, Vol. 04, No. 02, DEC 2021, pp 21-24



# DC Motor Speed Control Through Arduino and L298N Motor Driver Using PID Controller

Pirah Peerzada[a], Wasi Hyder Larik[a] , Aiman Abbas Mahar[b]
[a] Department of Electrical Engineering, [b] Department of electronics Engineering.
Mehran University of Engineering and Technology, Jamshoro, 76062, Pakistan
Corresponding author e-mail: (pirhpeerzada@gmail.com , wasilarik@gmail.com)
[**Received on:** 16/10/2021 **Accepted on**: 21/10/2021 **Published on:** 04/12/2021]

***Abstract*** **Electric drives that are high in efficiency are required for emerging technologies such as electric cars. In this situation, a direct current (DC) motor is more efficient, has appealing speed-torque characteristics, and is easier to regulate. This research models a DC motor and analyses its speed regulation under various loading circumstances. PID controllers are used to regulate the speed of the machine.**

***Key Words*—Arduino UNO, L298N motor driver, PID controller.**

## I. INTRODUCTION

Among various actuators, DC motors are the mostly common in use. It is so because of their higher efficiency, greater speed torque characteristics and availability in the market. Moreover, there is a great variety in DC motors due to which they are largely used according to their applications [1]. In DC motor, to control its speed is a great concern. There are numerous techniques through which speed of the DC motor can be controlled [2]. This paper aims to develop an algorithm in which we can achieve desired speed of a DC motor on simulation, created in Simulink-MATLAB. The algorithm is furthermore implemented on hardware system to check the actual response of DC motor. Moreover, simulated response is compared with actual response of hardware system in order to observe the accuracy and precision. The algorithm which would be used here is based on proportional integral derivative (PID) theory of control engineering.

## II. SYSTEM BLOCK DESCRIPTION

The block diagram that is shown in Fig.1 represents the processing of our developed system. This block diagram is divided into three major components viz Arduino UNO, PID algorithm and DC motor with encoder.

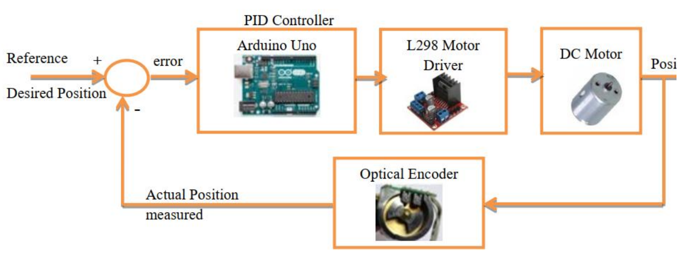


Figure 1. Block Diagram for DC Motor Speed Control

Microcontroller, Arduino UNO, is supplying voltage input to plant (motor) after getting it processed. While the PID gains that are preset in Arduino UNO are responsible to compensate and decide how much voltage is to be supplied further to our plant [3-4]. The DC motor, the third major component of this system, is receiving voltage from Arduino UNO via motor driver L298N. the encoder that relates to DC motor is providing feedback of output response to the microcontroller to aid Arduino in adjusting the input voltage to DC motor.

## III. HARDWARE COMPONENTS

### 1. Arduino UNO

Arduino UNO that is shown in fig. 2. is a microcontroller that is most popular among all the microcontrollers produced by Arduino. It has 14 digital pins (of which 6 pins can be used as PWM pins), 6 analog pins, 6 power pins. Moreover, it has a USB jack to upload programs in it, an external jack for DC power supply and a reset button.

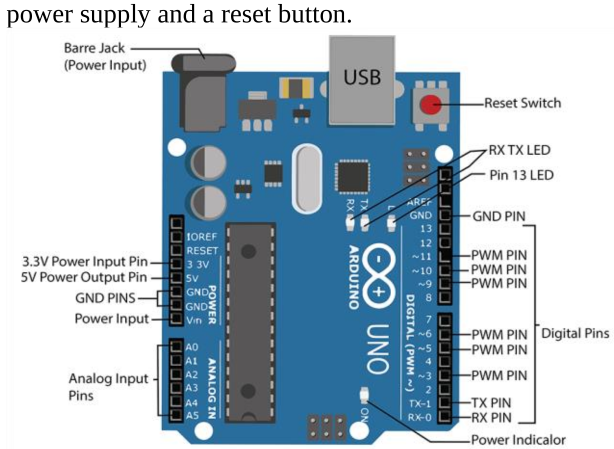


Figure 2. Arduino UNO

### 2. L298N Motor Driver

This L298N Motor Driver Module that is shown in fig. 3 is a high-power motor driver module for driving DC and Stepper Motors.

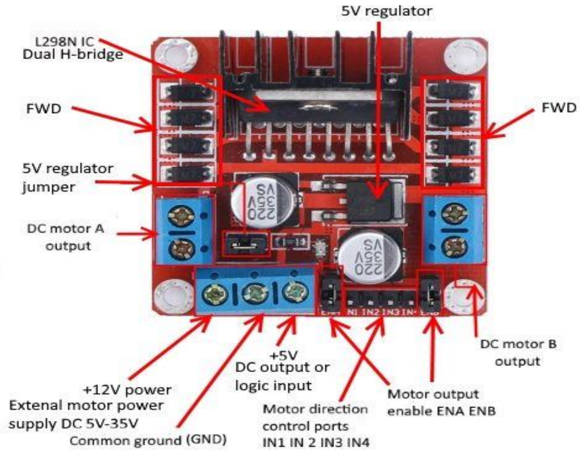

Figure 3. Motor Driver L298N

This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control.

### 3. Toshiba DGM-0090-2A DC Motor

Toshiba DGM-0090-2A, 23V/455 RPM, DC motor is a powerful geared motor that is used here in this system. This motor is coupled with an encoder which is actually a sensor and creates a feedback path for developed PID system. Encoder will count the pulses due to rotation of the shaft of this motor.

Figure 4. Toshiba DGM-0090-2A DC Motor

In result, actual RPM response of motor will be sent to the controller to further adjust the output response.

4. Encoder

The magnetic encoder, that is attached behind the DC motor here, will count the pulses that are made through the rotation of DC motor. These pulses are translated into the speed of DC motor, i.e., RPMs. The relation between encoder counts and motor's RPMs is given as:

$$Enc\ Diff = (Enc)new - (Enc)old$$

$$Motor\ Speed\ in\ RPM = \frac{Enc\ Diff * 60 * 100}{Pulse\ count\ per\ rotation}$$

### 5. Dell 23V/6A Power Supply

To energize our system, a dell 23V/6A power supply, that is shown in fig. 5. is connected as the main power source.
*-

## IV Analysis of PID Controller

Proportional-integral-derivative (PID) controllers are commonly used in DC motor speed control. To make it easier to identify the correct values of the PID controller's parameters for the DC motor under control at any set point, a suitable tuning process was required [5-7]. Controller tuning techniques give controller parameters in the form of formulae or algorithms. They ensure that the obtained control system is stable and achieves the desired results. Because of their excellent performance, ease of control, and high efficiency, direct current motor drives are used in a wide range of speed and position control systems [8-9]. The PID controller is in the forward path, and its output determines the voltage provided to the motor's armature. The controller computes an error value by dividing a measured process variable by the set-point value. It then seeks to decrease the error by increasing or decreasing the control inputs to the process, bringing the process variable closer to the set point [10].

## V SIMULATION

The system is first modelled in the MATLAB Simulink environment with the actual parameters that are used in the hardware system here. Fig. 5. Shows the simulation of the developed system.

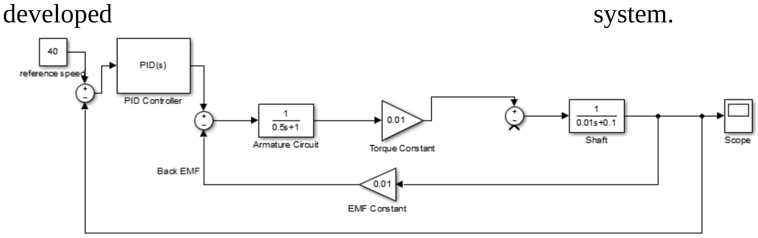

Figure 5. Simulation of the DC motor with controller

The output of this system, that is shown on the screen of the oscilloscope connected at the end of the simulated mode, can be seen in fig. 6.

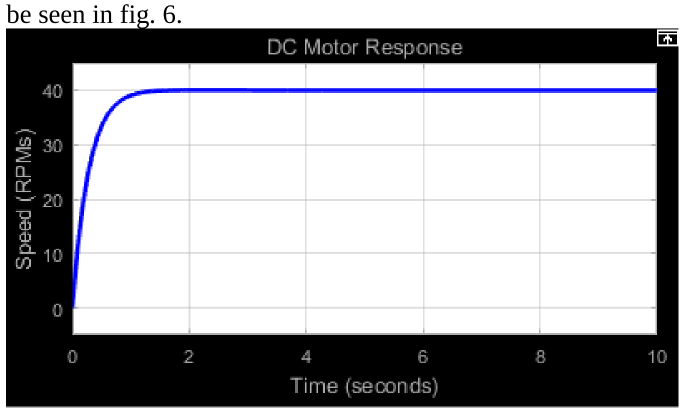

Figure 6. Output of the simulated DC motor with controller

Here, step input is provided to the system and the speed of the DC motor reaches to 40RPMs (setpoint) in a span of 1 second.

This shows that the controller tends our motor to reach its set value in a very short time.

## VI METHODLOGY

In this system, voltage is input while the speed of the DC motor is our output. Voltage supply is connected with the motor driver. The voltage that is being supplied to the DC motor is controlled through PWM. Now here, output speed of the DC motor is our actual value of output. When it is equal to the setpoint which we have already set in our controller, the controller will not vary the input. However, when the actual RPMs, sensed by the encoder attached with the DC motor, change from the setpoint value, an error signal generates. The controller, through the PID algorithm, tunes the error signal with proportional, integral, and derivative gains. Furthermore, that error signal, that is basically a difference of actual and set values, triggers the controller to vary the input voltage to the DC motor through the motor driver L298N. Consequently, the output is varied in accordance with the voltage on the input side of the DC motor. This process continues as long as the actual speed of the motor varies from the setpoint speed of the motor. The hardware system of the developed system of the speed control of DC motor through Arduino UNO is shown in the fig. 6.

Figure 7. Setup for DC Motor Speed Control

## VII RESULTS

The results are first obtained at the serial monitor of the Arduino IDE. Furthermore, the results are plotted in MATLAB in order to obtain the curves that are discussed ahead. Fig. 7 shows the output of the DC motor when a sine input, at setpoint value of 100 RPM is given to it. From figure, it can be clearly seen that the actual response is precisely following the simulated response of the DC motor.

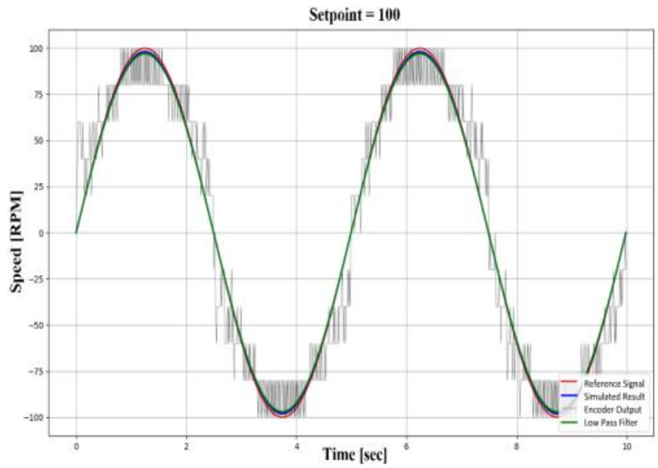


Figure 8. Motor response for sine input (setpoint=100 RPM)

In fig. 8., output can be seen when a sine input at setpoint value of 360 RPM is applied.

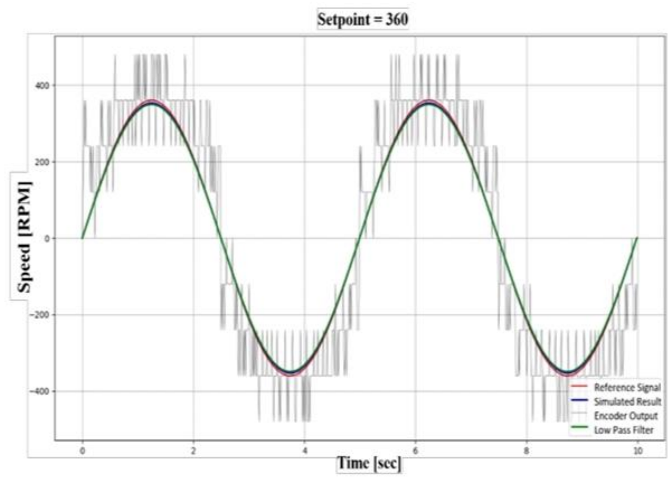


Figure 9. Motor response for sine input (setpoint=360 RPM)

In fig. 9., step input is given to the system and its response is observed at a setpoint of 100 RPMs.

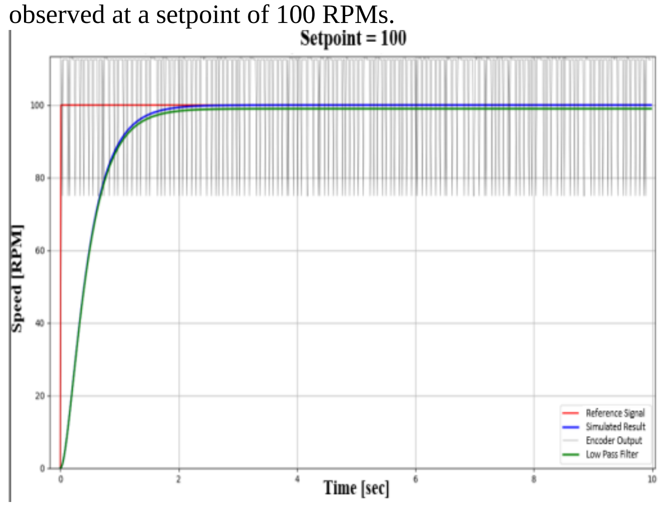


Figure 10. Motor response for step input (setpoint=100 RPM)

While fig. 10., shows the output of the developed system when step input is given at a setpoint of 360 RPMs.

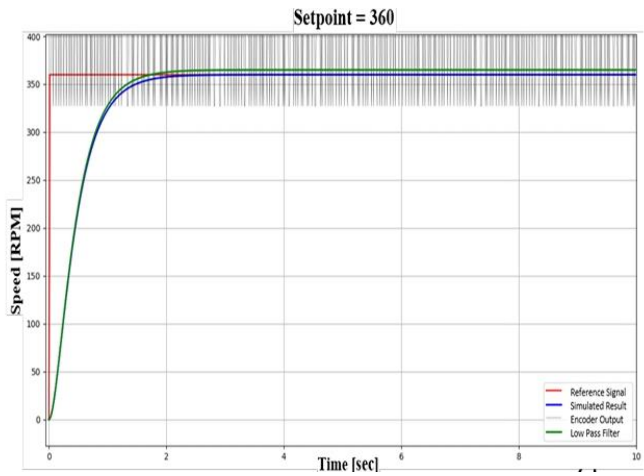

Figure 11. Motor response for step input (setpoint=360 RPM)

## CONCLUSION

All the discussion concludes that speed of the DC motor has been controlled successfully. PID algorithm is used here to achieve the desired speed of the DC motor. Initially, the system is simulated in MATLAB-Simulink environment. Furthermore, it is implemented on hardware system. Both the simulated and actual results have been compared and it has been observed that the actual response of our hardware system is precisely following the simulated response of our system. This is a big achievement of this study.

## REFERENCES

[1] Lopes, J. A. P., Soares, F. J., & Almeida, P. M. R. (2010). Integration of electric vehicles in the electric power system. Proceedings of the IEEE, 99(1), 168-183.

[2] Munoz-Hernandez, G. A., Mino-Aguilar, G., Guerrero-Castellanos, J. F., & Peralta-Sanchez, E. (2020). Fractional order PI-based control applied to the traction system of an electric vehicle (EV). Applied Sciences, 10(1), 364.

[3] Badamasi, Y. A. (2014, September). The working principle of an Arduino. In 2014 11th international conference on electronics, computer and computation (ICECCO) (pp. 1-4). IEEE.

[4] Severance, C. (2014). Massimo banzi: Building arduino. Computer, 47(1), 11-12.

[5] 28 Bansal, H. O., Sharma, R., & Shreeraman, P. R. (2012). PID controller tuning techniques: a review. Journal of control engineering and technology, 2(4), 168-176.

[6] Sabir, M. M., & Khan, J. A. (2014). Optimal design of PID controller for the speed control of DC motor by using metaheuristic techniques. Advances in artificial neural systems, 2014.

[7] Aborisade, D. O. (2014). DC motor with load coupled by gears speed control using modified Ziegler-Nichols based PID tunings. Control Theory and Informatics, 4(5).

[8] Bouyedda, H., Ladaci, S., & Schulte, H. (2019, October). Cruise control design for an Electric Vehicle using a fractional order PI λ D μ with Genetic Algorithms optimization. In 2019 8th International Conference on Systems and Control (ICSC) (pp. 158-163). IEEE.

[9] Arulmozhiyal, R., & Kandiban, R. (2012, January). Design of fuzzy PID controller for brushless DC motor. In 2012 International Conference on Computer Communication and Informatics (pp. 1-7). IEEE.

[10] Toscano, R. (2005). A simple robust PI/PID controller design via numerical optimization approach. Journal of process control, 15(1), 81-88.