

DC MOTOR PID CONTROLLER WITH PWM FEEDBACK

Jotje Rantung, Hengky Luntungan

Department of Mechanical Engineering, Faculty of Engineering Sam Ratulangi University

ABSTRACT

This paper proposes a DC Motor PID controller with PWM feedback. Angular position control of DC motor using pulse width modulation (PWM) feedback mechanism and Arduino microcontroller model of DC Motor are presented in this paper. To do this task, the followings are done. Firstly, a dynamic mathematical model of a DC motor is derived. Secondly, designing a control system to be applied to the DC motor. Thirdly, implementing the hardware and software. The programming code for PWM signal execution is written by using c code and matrix laboratory for Arduino microcontroller. The instruction code is implemented by the Arduino microcontroller to execute the control signal. Finally, the proposed DC Motor PID controller with PWM feedback is proven by experimental. The experimental results show the effectiveness of a DC Motor PID controller with PWM feedback. The output can track the 260 rpm setpoint properly. The experimental results are not smooth due to the noise in the system.

Keyword: DC motor, PID controller, PWM, microcontroller.

1. INTRODUCTION

Direct Current (DC) Motor is an actuator that is frequently used to control not only in the industrial technology but also in households. The DC motor has a fast response, but it still has a steady state error. Therefore, an appropriate controller is required for the plant. Proportional Integral Differential (PID) controller is a control action that has a fast response. Therefore, this control action is suitable for controlling the speed of a DC motor.

In this paper, a PID controller design for a DC motor angular position control is proposed. DC motor angular position systems are usually controlled by proportional integral-derivative (PID) control algorithms with PID coefficients tuned for optimizing operation. The objective of a PID controller in a position control system is to maintain a position set point at a given value and be able to accept new setpoint values dynamically. Modern position control environments require controllers that are able to cope with parameter variations and system uncertainties.

To implement a PID controller the proportional gain K_P , the integral gain K_I and the derivative gain K_D must be determined carefully. Controlling the DC motor without using the PID controller will give some oscillation in the signal and because the system is nonlinear, controlling by function is the best way to control the nonlinear systems and PID controller is the best choice to achieve this task.

The more motor speed the less torque the less motor speed the more torque. Controlling speed is done by using pulse width modulation (PWM). All of this is written by using c code and matrix laboratory (Matlab) for Arduino microcontroller. When using digital control, a pulse-width modulated (PWM) signal is used to generate an average voltage. The motor winding acts as a low pass filter, so a PWM waveform of sufficient frequency will

generate a stable current in the motor winding. Even though speed is proportional to PWM duty cycle, in systems where precise speed control is required, some sort of feedback mechanism must be included.

2. DC Motor Dynamic Mathematical Model

The speed and torque of the motor depend on the strength of the magnetic field generated by the energized windings of the motor, which depend on the current through them. The DC motor speed is given by the relation as follows:

$$N = K \frac{(V - I_a R_a)}{Z\phi} \quad (1)$$

where,

$$K = \frac{60A}{P}$$

R_a = armature circuit resistance

I_a = armature current

V = motor input voltage

ϕ = machine's total flux

Z = motor's impedance

Therefore, adjusting the rotor voltage (and current) change motor speed. In anchor control separately excited DC motor, voltage is applied. The motor anchors are adjusted to control the motor speed without changing the voltage applied to the field [1].

The electric equivalent circuit of the armature and the free-body diagram of the armature controlled separately excited DC motor is shown in Fig. 1.

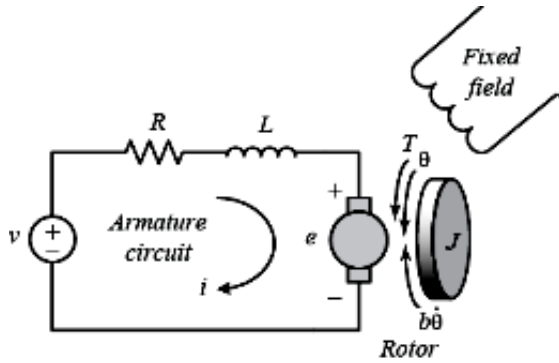


Fig. 1. The electric equivalent circuit of the armature and the free-body diagram of the DC motor

Voltage source (V) applied to the motor's armature is considered as the input of the system; while the rotational speed of the shaft ($d\theta/dt$) is the output. The rotor and shaft are assumed to be rigid. It is further assumed that the system follows viscous friction model, that is, the friction torque is proportional to shaft angular velocity [2].

In general, the torque generated by a DC motor is proportional to the armature current and the strength of the magnetic field. It is assumed that the magnetic field is constant and, therefore, that the motor torque is proportional to only the armature current (I) by a constant factor motor torque constant, K_t as follows:

$$T = K_t \cdot I \quad (2)$$

This is referred to as an armature-controlled motor. The back emf, e , is proportional to the angular velocity of the shaft by a constant factor back-emf constant, K_e . The back emf, e , is obtained as follows:

$$e = K_e \cdot \dot{\theta} \quad (3)$$

In SI units, the motor torque and back emf constants are equal, that is,

$$K_t = K_e \quad (4)$$

Therefore, K is used to represent both the motor torque constant and the back emf constant.

From Fig. 1, the following governing equations based on Newton's 2nd law and Kirchhoff's voltage law are derived.

$$J\ddot{\theta} + b\dot{\theta} = Ki \quad (5)$$

$$L\frac{\partial i}{\partial t} + Ri = V - K\dot{\theta} \quad (6)$$

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

$$(J + b)\theta(s) = KI(s) \quad (7)$$

$$(Ls + R)I(s) = V(s) - sK\theta(s) \quad (8)$$

The open-loop transfer function is obtained by eliminating $I(s)$ between the Eqs. (7) and (8). The rotational speed is considered as output and the armature voltage is considered as input. The open-loop transfer function is obtained as follows:

$$\frac{w(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \left[\frac{\text{rad / sec}}{V} \right] \quad (9)$$

In case of all known parameters of DC Motor, the open loop step response is shown in Fig. 2 as follows:

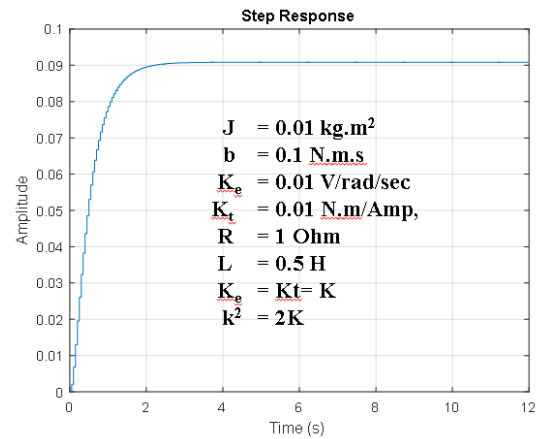


Fig. 2. DC Motor, the open loop step response in known parameters

3. PID CONTROLLER

3.1 PID Tuning Algorithms

A proportional–integral–derivative controller (PID controller) is a generic control loop feedback mechanism widely used in industrial control systems.

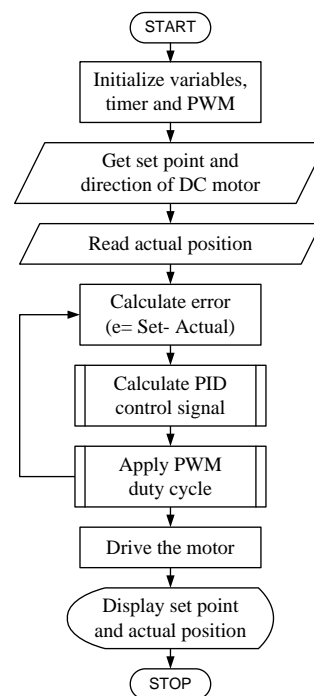


Fig. 3. Flowchart for System Implementation

A PID controller calculates an error value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs [3]. Fig. 3 mentions implementation program flow chart for this setup.

Fig. 4 shows the PID controller design for this control system. PID equation is calculated to get the control signal for DC motor. From Eq. (10), $e(t)$ is position error value difference between set angle and output measured angle(actual angle), $u(t)$ is PWM signal for DC motor and $y(t)$ is the actual angle. K_p , K_i and K_d were respectively with the values of proportion, integral, and the differential coefficient.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (10)$$

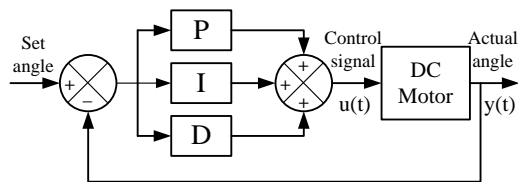


Fig. 4. Closed Loop Position Control of the DC Motor using PID Controller

The programming code for PWM signal execution is calculated by the following procedure. The instruction code is implemented by the Arduino microcontroller to execute the control signal.

```

error    = output - input;
Integral = xT * error + prv_ui;
Diff     = (error - prv_error) / xT
PWM      = kp * error + ki * Integral + kd * Diff;
Usignal  = abs(PWM);
    
```

From above code, the proportional gain (K_p) examines the magnitude of the error and it responds proportionally. Although it has a large error, the magnitude of the motor position will receive a large response. The integral gain (K_i) efforts to reduce the steady state error. And the derivative gain (K_d) attempts to look at the rate of change of the error signal. Derivative control will decrease the overshoot to become a greater system response curve of the motor position to a rapid rate of change. Finally, the controller is calculated the PWM signal depends on the signal variation of DC motor by using PID gain values to improve the system performance.

3.2 Pulse Width Modulation

Pulse width modulation (PWM) is a technique which allows adjusting the average value of the voltage that's going to the electronic device by turning on and off the power at a fast rate. The average voltage depends on the duty cycle, or the amount of time the signal is ON versus the amount of time the signal is

OFF in a single period of time. It is depending on the size of the motor, the user can simply connect an Arduino PWM output to the base of transistor or the gate of a MOSFET and then control the speed or position of the motor by controlling the PWM output [4]. PWM and duty cycle relation diagram is shown in Fig. 5.

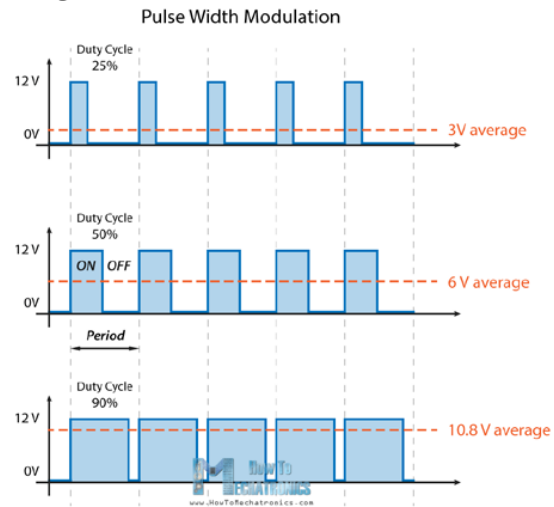


Fig. 5. PWM and Duty Cycle Diagram

In this paper, the value of PWM signal is from 0 to 255 or that's 0 to 100% duty cycle of the PWM signal. The duty cycle diagram is as shown in Fig.6. The PWM value is executed by PID controller to reduce the error difference between the desired angle and the actual angle.

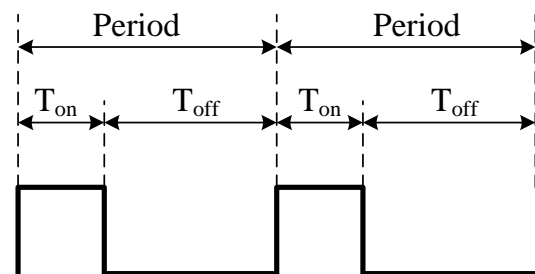


Fig. 6. Duty Cycle Diagram

The duty cycle of PWM signal is calculated by the equation as follows:

$$\left. \begin{aligned} \text{Period} &= 1 / \text{Frequency} \\ \text{Period} &= T_{on} + T_{off} \\ \text{Duty Cycle} &= T_{on} / (T_{on} + T_{off}) * 100(\%) \end{aligned} \right\} \quad (11)$$

4. HARDWARE CONFIGURATION

4.1 Arduino Uno Microcontroller

Arduino Uno is a microcontroller board based on the ATmega328P shown in Fig. 7. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and

a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started [5]. In this paper, the Arduino microcontroller is very well-suited to drive the PWM signal for DC motor for the improvement of the output response for the DC motor position control system.

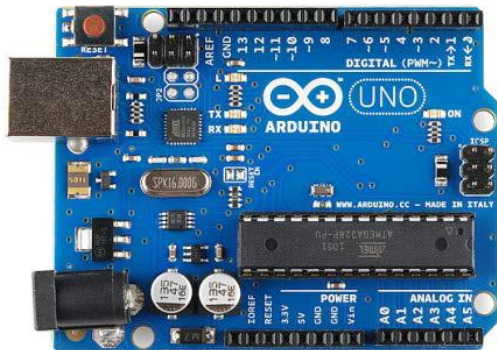


Fig. 7. Arduino Uno Microcontroller

4.2 L298N Dual H-Bridge Controller

The L298N H-bridge IC shows in Fig. 3 that can allow to control the speed and direction of two DC motors. This module can be used with motors that have a voltage of between 5 and 35VDC with a peak current up to 2A. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output. Pin assignments for L298N dual H-Bridge Module is shown in table 1. The digital pin assign from HIGH to LOW or LOW to HIGH is used IN1 and IN2 on the L298N board to control the direction. And the controller output PWM signal is send to ENA or ENB to control the position. The forward and reverse speed or position controlling for the motor has done by using PWM signal [6]. Then using the analog Write() function and send the PWM signal to the Enable pin of the L298N board, which actually drives the motor.

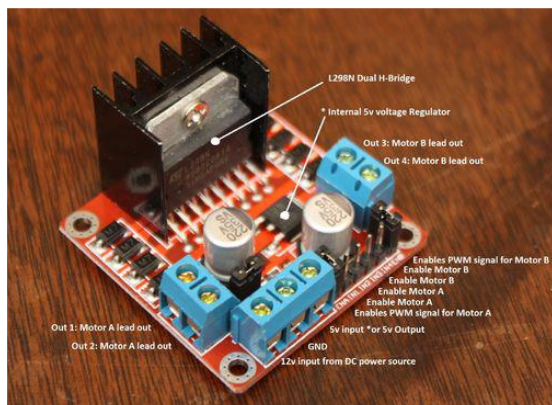


Fig. 8. L298N dual H-Bridge Controller Module

Table.1 Pins assignment for L298

Out 1: Motor A lead out

Out 2: Motor A lead out
Out 3: Motor B lead out
Out 4: Motor B lead out
GND: Ground
5V: 5V input
ENA: Enables PWM signal for Motor A
IN1: Enable Motor A
IN2: Enable Motor A
IN3: Enable Motor B
IN4: Enable Motor B
ENB: Enables PWM signal for Motor B

4.3 DC Geared Motor with Encoder

The 12VDC gear-motor with encoder shown in Fig. 9 is a powerful motor to drive the position control system. It comes with the photoelectric encoder output and planetary gear ratio reduced by 80:1 gear. It can provide 120 rpm with the rated voltage of 12VDC [7]. To read the count values from the encoder, the user would check the condition of channel A and B rotation applying by experiment. For the rotor shaft count per revolution values, it is very important to multiply the gear ratio by count values. The specification of the motor is shown in Table 2

Fig. 10 shows about the encoder circuit for Namiki 12VDC motor. It is needed to connect two 10k Ohms to channel A and channel B supplied by 5V. One of the encoder pins is connected with 120 Ohms resistor and another pin ground.

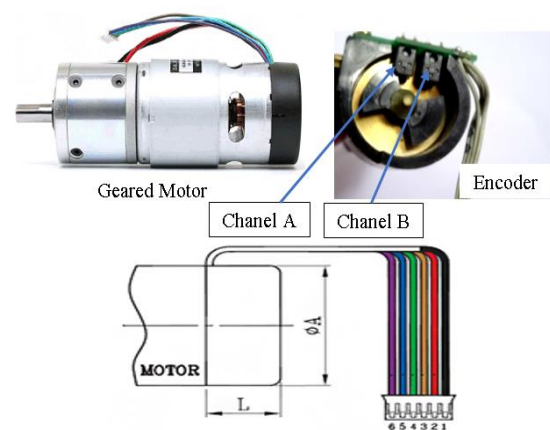


Fig. 9. DC Geared Motor with Encoder

Table. 2 Namiki motor features

Brand	Namiki Coreless Motor
Model	22CL-3501PG
Operating Voltage	12 VDC
Rated Power	15 W
Stall torque	16.5 kg.cm

Continuous torque	5 kg.cm
Diameter	22mm
Shaft length	19mm
Length	65mm(excluding encoder, gear box and output shaft)
Stall current	1.8A
Reduction ratio	80:1
Output speed	120 rpm
Weight	140g

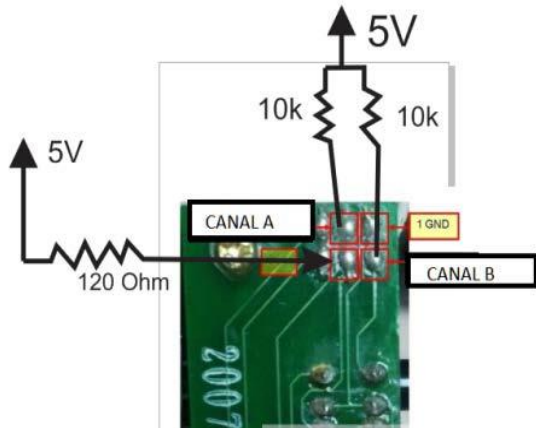


Fig. 10. Encoder Circuit for DC Geared Motor

4.4 Reading the Encoder Pulse

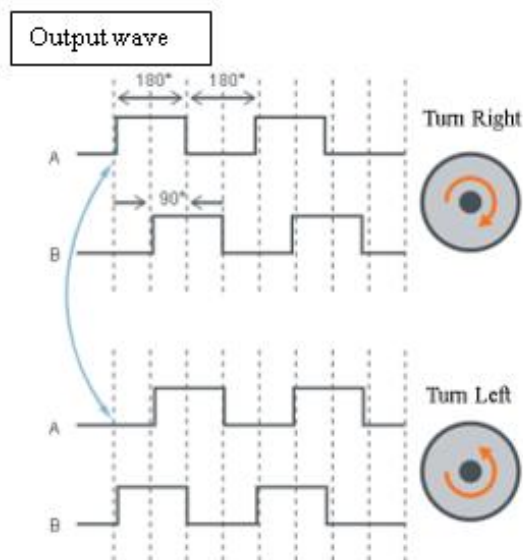
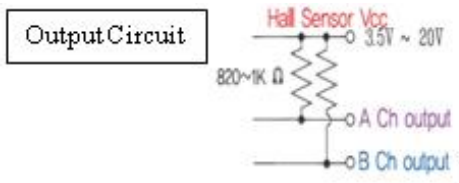


Fig. 11. Encoder Channel Description

In this work, the encoder comes with the photoelectric encoder output (channel A and channel B). It is used to sense the rotation of a magnetic disk on a rear protrusion of the motor shaft forward or backward direction according to the pulse count per revolution. From Fig. 11, it is the clockwise direction from channel A leading to channel B because the A interrupt seeing attached to B whenever it changes state LOW to HIGH or HIGH to LOW condition. The another direction is counterclockwise caused by channel B leading to channel A because the B interrupt seeing attached to A whenever it changes state LOW to HIGH or HIGH to LOW condition.

4.5 Calculating the Angular Position

The encoder from the gear-motor provides a resolution of 8 counts per revolution and exact gear ratio is 80:1 metal gearbox. Therefore, it can get the counts per revolution (CPR) to the angular position as the following calculation:

$$\text{CPR} = 8 \times 8 \text{ CPR}$$

$$\text{Count/degree} = 640 \text{ counts/rev} \\ \times 1 \text{ rev/360 deg}$$

$$\text{Let } X = 640 \text{ counts/360 deg}$$

$$\text{Therefore } 1/X = 360 \text{ deg/640 counts}$$

$$\text{For attachInterrupt X2}$$

$$\text{Angular position} = (1/X) \times 2 \times \text{counts} \\ = (720/640) \text{ deg}$$

The instruction code was implemented in the program by using the above equation to get the angular position of the DC motor.

4.6 Circuits Design of System

The circuit schematics for overall system description is shown in Fig. 12. In this circuit, three main hardware components are implemented for this research. Arduino Uno microcontroller is to control the position of DC motor by controlling the input voltage to the motor. PID tuning algorithms is implemented in microcontroller to execute the PWM signal for DC motor drive. A 12V Namiki DC gear-motor is a powerful motor to drive the position control system. It comes with the photoelectric encoder output, planetary gear reducer and 80:1 gear ratio, which provide 120 rpm with 12VDC rated voltage. L298 dual H-Bridge motor driver which is allows controlling the direction and position of DC motor.

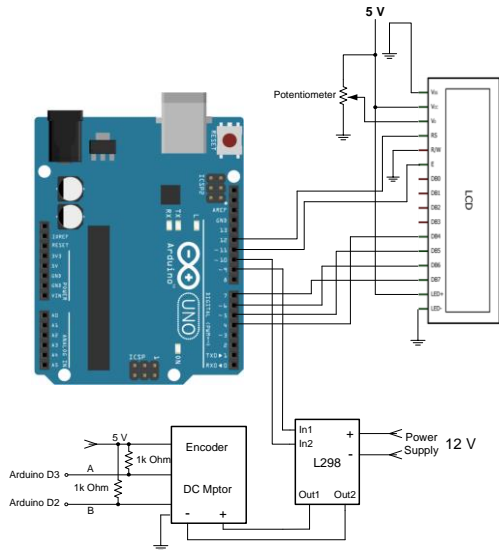


Fig. 12. Circuit Schematics for Overall System

5. EXPERIMENTAL RESULTS

Hardware implementation arrangement is shown in Fig. 13 as follows:

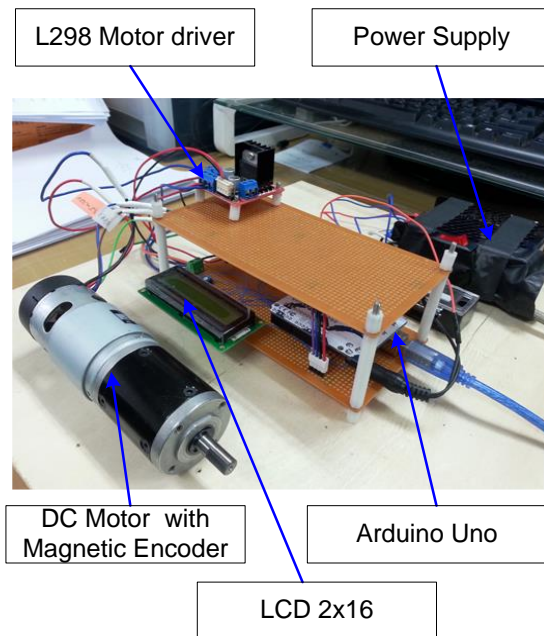


Fig. 13. System Hardware implementation

All of the measurement values can be seen in serial monitor or LCD display. Getting from the serial values, the output result curves are plotted by using MATLAB to be seen clearly.

Experiment results for the same setpoint 260 rev/min of the P, PI, and PID controller mode are shown on Figs. 14~16. Fig.14 shows the step response if only the P (Proportional) controller is applied in the DC Motor. A proportional controller allows tighter control of the process variable because its output can take any value between fully on and fully off, depending on the magnitude of the error signal. With proportional band, the controller output

is proportional to the error or a change in measurement and offset (deviation from set-point) is present. This steady state error is the difference between the attained value of the controller and the required value. In this DC Motor, the gain for P is 3 and the set point is 360. As we can see, there is undershoot at 240 and it over the target value which is 260.

Fig.15 shows the step response when the Proportional (P) and Integral (I) gain is applied to the DC Motor. To eliminate the offset error, the controller needs to change its output until the process variable error is zero.

Integral action gives the controller a large gain at low frequencies that results in eliminating offset and beating down load disturbances. It also can reduce the final error in a system. Summing even a small error over time produces a drive signal large enough to move the system toward a smaller error. In this system, the gain for P is 3 and I is 1. In this graph can be seen that the steady state error still occurs, but it is now at the set point input which is 260. However, the overshoot became greater than before which is at 300. The rise time is also faster than just using the proportional controller.

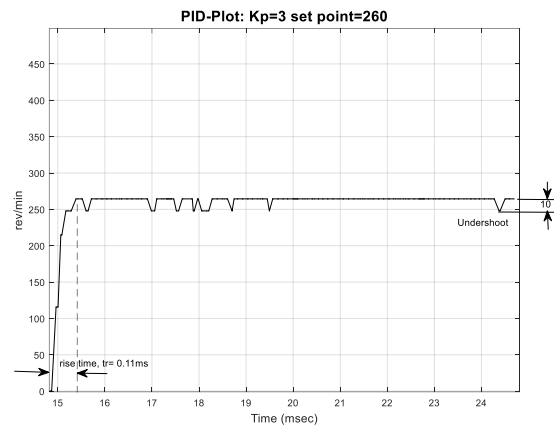


Fig. 14. P-Controller Mode

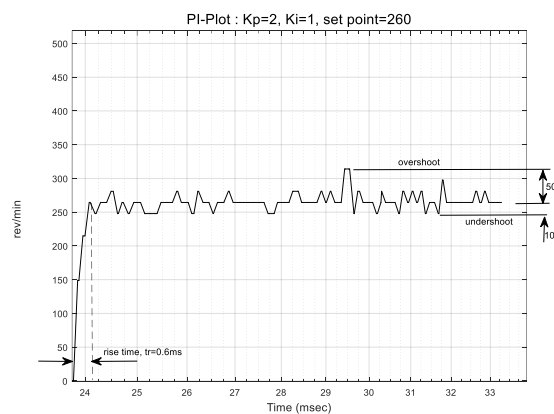


Fig. 15. PI-Controller Mode

Fig. 16 shows the step response if the entire element in PID controller is applied in the system.

As the proportional gain is increased, the controller responds faster. If the proportional gain is too high, the controller may become unstable and oscillate. The integral gain acts as a stabilizer. It provides power even if the error is zero. The Derivative controller counteracts the K_p and K_i terms when the output changes quickly. This helps reduce overshoot and ringing and it has no effect on final error. In this DC Motor, the gain for Proportional is 5, Integral is 3 and the gain for Derivative is 2. The step response for this system became smooth and the response is better than if we use P, PI, and PD. There is no overshoot; the rise time is faster and less error. The settling time is exactly at the set point which is 260.

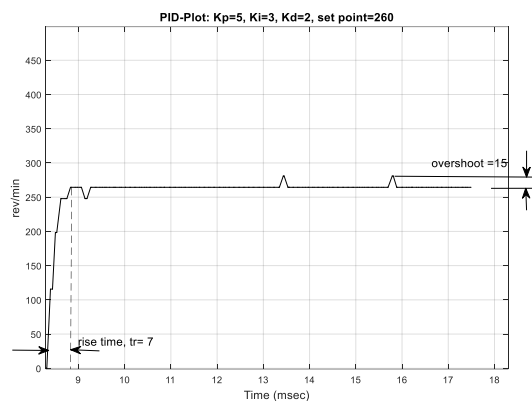


Fig. 16. PID-Controller Mode

6. CONCLUSIONS

This paper presented a DC Motor PID controller with PWM feedback. The output can track the 260 rpm setpoint properly. The experimental result show that every different control action has different result. P controller mode with $K_p=3$ has undershoot 10 rpm. PI controller mode with $K_p=2$, $K_i=1$ has overshoot 50 rpm and undershoot 10 rpm. PID controller mode with $K_p=5$, $K_i=3$, and $K_d=2$ has overshoot=15 rpm. The experimental results are not smooth due to the noise in the system.

REFERENCES

1. Sailan K., Kuhnert I., DC Motor Angular Position Control using PID Controller for the purpose of controlling the Hydraulic Pump. International Conference on Control, Engineering and Information Technology (CEIT'13), (2013), pp. 2-6.
2. Theraja B., Theraja A., A Textbook of Electrical Technology (Volume II- AC and DC machines). S. Chand, (2011), pp. 995-1020.
3. Detchrat, et. al., "IMC-Based PID Controllers Design for Two-Mass System", IMECS, (2012) Vol. II, Hong Kong.
4. <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>

5. <https://store.arduino.cc/usa/arduino-uno-rev3>
6. [Online]. Available: <http://www.instructables.com/id/Control-DC-and-stepper-motors-with-L298N-Dual-Motor/>
7. Detchrat, et. al., "IMC-Based PID Controllers Design for Two-Mass System", IMECS, (2012), Vol. - II, Hong Kong.