Eng.J.CMU.[2021] 28 [1], 195-211

วารสารวิศวกรรมศาสตร์
มหาวิทยาลัยเชียงใหม่
Engineering Journal
Chiang Mai University

Comparison of PI and PID control performances coupled with Kalman Filter for DC motor speed control via MATLAB/Simulink

Sirapat Boonsing and Pornsiri Kaewpradit*

Department of Chemical Engineering, Prince of Songkla University, Songkhla, 90112, THAILAND

* E-mail: pornsiri.k@psu.ac.th, Tel.: +66-074-287283

ABSTRACT

Efficient mixing is required in various industries to require homogeneity and system consistency. DC motor speed control is important for mixing processes. In this study, proportional integral (PI) controllers and proportional integral derivative (PID) controllers with simple first order filters designed in MATLAB/Simulink are compared using three different tuning methods such as Ziegler-Nichols (ZN) step response, Chien-Hrones-Reswick (CHR) step response and Closed-loop Pole placement design methods. The purpose of DC motor speed control is to respond quickly without overshoot to prevent damage to the DC motor. Therefore, Kalman filter is considered to eliminate the occurrence of excessive overshoot. The motor output signal is detected by an encoder and is then converted to revolutions per minute unit (rpm) before going through the noise reduction and the controller respectively. The motor input is a pulse width modulation (PWM) cycle signal which is set in the range of 0-255. Experimental results show that the Kalman filter is excellent in all cases the noise and excessive overshoot of the DC motor speed. It can also accurately assess its state and can be used to reduce the aggressiveness of DC motor responses. The DC motor speed control performance of the controllers based three different tuning methods are also compared. It found that the PID controller based pole placement tuning method with the Kalman filter provides the most efficient response without overshoot for all desired speed values.

Keywords: DC motor; Overshoot rejection; Ziegler-Nichols Method; CHR Method; Pole placement Method; Kalman Filter

1. INTRODUCTION

The mixing process is widely used in various industry. The mixing is an important process for improving homogeneity and system consistency. The mixing occurs when the materials are moved from one area to another in a vessel. Inefficient mixing causes waste of energy [1]. One of the improvements of the mixing process is motor speed control. The reason that the motor speed must be controlled in order to prevent excessive

Received 24 December 2020
Revised 25 March 2021
Accepted 31 March 2021

energy loss and damage to the machine. A direct current (DC) motor is a direct current electric machine that transforms electrical energy into mechanical energy. Advantages of DC motors are suitable for small jobs, easy to use and less maintenance costs compared to alternating current (AC) motors. The DC motors are then preferred when extensive speed range control is desired [2]. The main difficulty of the DC motor control is to maintain at the set point value, especially in case of speed set point change (known as servo-control operation). There is a chance that the overshoot will be too high to accept which causes the motor swing aggressively resulting in damage to the motor. Therefore, appropriate motor speed control is then desired.

There are many methods to control DC motor presented in previous research works. Vikhe, Punjabi and Kadu (2015) used the PID controller in Lab View to control the speed of DC motor. The method introduced in their work is a low-cost method for the DC motor speed control. IR sensor is used to measure the motor speed by sending the signal back to the controller to compare with the desired speed. The signal is sent to the pulse width modulation (PWM) pins. Then the microcontroller sends PWM signal to the motor to drive the motor to spin. Hamoodi, Mohammed and Salih (2018) used the PID controller by Simulink to control the DC motor speed and can be used. Modeling of DC motors is an equation of current (I), torque (T), and speed (ω). PID parameters are tuned by the Ziegler-Nichols method. The output efficiency obtained from the normal values in the PID is very close to accuracy. Moreover, the results obtained from actual use are close to the simulation results. Husnaini, Krismadinata and Hastuti (2019) provided a step-by-step design and comparison of PI and PID controller responses. These controllers are used to control the speed of a DC shunt motor in open-loop and closed- loop conditions without controller. The simulation shows that the PID controller used DC shunt motor speed control is better than PI controller.

Another method used to improve controller parameters is self-tuning based on a Kalman filter, which is known to be effective because it provides good estimation results under noisy environments. The Kalman filter is used with PID parameters estimation problems to increase the performance of parameters estimation that presented in previous research works. Kumar and Murthy (2016) used the Kalman filter to control speed of DC motor for robotic system in wheelchair. The state and parameters were estimated using armature current and rotor shaft rotation speed measurements. The Kalman filter has the capability to evaluate the states and model parameters of the best, even where is noise. And Abdulameer et al (2016) controlled the motor speed using PID controller based tuning methods that are the Ziegler-Nichols method (ZN) and the CHR method. Ziegler-Nichols method provides a faster response than CHR method in case where overshoot is acceptable. If the system considers the overshoot to be more important than the fast response, the CHR method is better because the overshoot value is less than the Ziegler-Nichols method. Currently, PID controllers are mostly used in various industries because it can increase production efficiency, prevent system damage, reduce production cost and be used for continuous process [2].

According to previous research studies DC motor control model with PI and PID controllers focusing on the equation of the relationship between current (I), torque (T), and speed (**ω**). And find the controller parameters from random to get the most suitable values. Therefore, this research focuses on practical applications and further studies the design and comparison of PI and PID controllers by using different methods to control DC motor speed via MATLAB/Simulink, aiming for the fastest and most accurate response without overshoot. The MATLAB is an easy-to-use program with Simulink that is a modeling support software, along with multidisciplinary block sets for the analysis of different systems. The encoder is a speed sensor that sends a PWM signal and converted it into revolution per minute (rpm). Additional studied methods used to design the PI and PID controllers are Ziegler-Nichols method, CHR method, and pole placement method, which has received the process transfer function from experiments. Since Ziegler-Nichols method is a very common method, CHR can eliminate overshoot, and the main advantages of the pole placement method are that it is easy to apply to the simple system and can be configured to meet design requirements. In literature, there are few studies have shown the comparisons of these three methods. In addition, the Kalman filter has been applied in this work to reduce noise sensitivity and to ensure the control system stability to prevent motor damage.

2. CONTROL SYSTEM DESIGN

2.1 PI controller

Nowadays, PI controller is widely used in industrial plants due to their simple structure, easy design, and low cost. But the PI controller is not suitable for controlling objects with high non-linearity and uncertainly [13]. In this controller, proportional gain (K_c) and integral gain (K_i) are the control parameters. The block diagram relating to the plant system is shown in Figure 1. The relationship of input or output to the error feedback for an ideal PI controller is equation as follow:

$$U(t)=K_c e(t)+K_i \int_0^t e(t) \quad (1)$$

$$U(s)=K_c + \frac{K_i}{s} \quad (2)$$

where U(t) and e(t) are the control variable and the error value by comparing the actual output and the desired input, respectively.

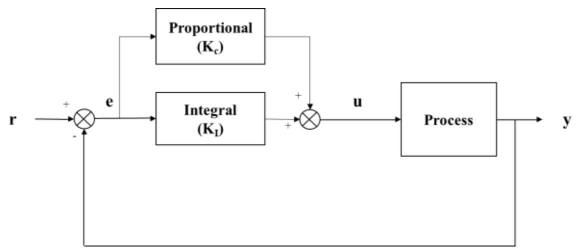


Figure 1 The structure of PI controller.

2.2 PID controller

PID controller is part of the control in a closed-loop system or feedback control system. The PID controller is popular due to its highly effective response and easy to use for an extensive range of operations. Inappropriate parameter values cause the system to become unstable [3]. In this controller, proportional gain (K_c), integral gain (K_i), and derivative gain (K_d) are the control parameters. The transfer function is shown as follow:

$$U(t)=K_c e(t)+K_i \int_0^t e(t) + K_d \frac{de(t)}{dt} \quad (3)$$

$$U(s)=K_c+\frac{K_i}{s}+K_d s \quad (4)$$

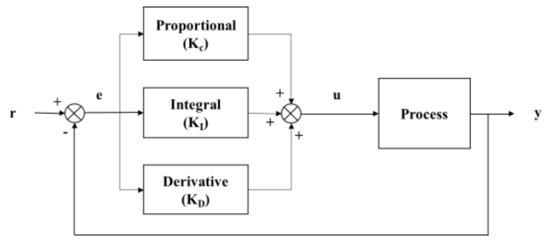


Figure 2 The structure of PID controller.

The error value (e) is the difference between the actual output value (y) and the desired input value (r). The signal U(t) is calculated by the controller and then sent to the process to the process output signal to the desired trajectory [8].

3. CONTROLLER TUNING METHOD

In this research, DC motor speed is controlled using PID controller based three different tuning methods such as Ziegler-Nichols (step-response) and Chien-Hrones-Reswick (CHR) (step-response) and closed-loop pole placement tuning methods. Noise reduction and speed-overshoot elimination of the DC motor response have also been investigated by applying Kalman filter. Details of those tuning methods to enable the process to meet its specified objectives are described in this section [9].

3.1 Ziegler-Nichols Method

This method applies to a process that has an open-loop step response. This method provides a normal response for the first order system with delays. The response of a step change in process input for most processes control is an "S-shaped curve" that begins gradually, sometimes with delays and then rises to a steady state. This type of process response can be estimated by a first order plus dead time (FOPDT) model as shown in Figure 3. The time constant (T) can be estimated in two ways that are at 100% and 63.2% of the change in measurement. For the delay time (L) can be estimated only one way. In this study considers the time constant at 63.2%. The controller parameters can be obtained from the Ziegler-Nichols formula as shown in Table 1. The process model is as follow:

$$P(s)=\frac{K_P e^{-Ls}}{Ts+1} \quad (5)$$

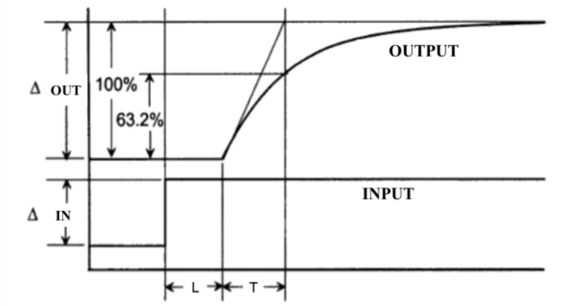


Figure 3 Response curve of FOPDT model for Ziegler-Nichols method [10].

Table 1 Ziegler-Nichols tuning method [11].

Type of controller	K_c	K_i	K_d
P	$\frac{T}{L}$	0	0
PI	$0.9\frac{T}{L}$	$\frac{K_c}{L/0.3}$	0
PID	$1.2\frac{T}{L}$	$\frac{K_c}{2L}$	$0.5L{\cdot}K_c$

3.2 Chien-Hrones-Reswick (CHR) Method

This method is modified from the original Ziegler- Nichols with superior response control to decrease overshoot [7]. This method focuses on the set point response and the disturbance response. The parameters used for tuning can be estimated from the FOPDT model such as gain (K_p), delay (L), and time constant (T). The tuning formulas for the system responses without overshoot (0% OS), and for the system responses with 20% overshoot (20% OS) are summarized in Table 2 and Table 3, respectively, where $a = \frac{K_P L}{T}$. In this study considers only 0% overshoot.

Table 2 Chien-Hrones-Reswick tuning method (0% OS) [7].

Type of controller	K_c	K_i	K_d
P	$\frac{0.3}{a}$	0	0
PI	$\frac{0.35}{a}$	1.2T	0
PID	$\frac{0.6}{L}$	T	$\frac{L}{2}$

Table 3 Chien-Hrones-Reswick tuning method (20% OS) [7].

Type of controller	K_c	K_i	K_d
P	$\frac{0.7}{a}$	0	0
PI	$\frac{0.6}{a}$	1.2T	0
PID	$\frac{0.95}{L}$	1.4T	0.47L

3.3 Pole Placement Method

The pole placement method is designed to control feedback in the system by placing the pole of the closed-loop system in the desired position on the S-plane. The pole placement refers to the configuration of the particular characteristics of the system, which is associated with the stability of the system directly according to theory. This method only applies to a system that can be controlled, which means that all status values can be measured from the sensors. The pole position is shown in Figure 10 (a). The 2nd order transfer function is shown as equation (6).

$$G(s)=\frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2} \tag{6}$$

$$P_{1,2}=\sigma\pm j\omega_d=-\xi\omega_n\pm j\omega_n\sqrt{1-\xi^2} \tag{7}$$

$$T_r=\frac{\pi-\beta}{\omega_d} \tag{8}$$

$$T_s=\frac{4}{\xi\omega_n} \tag{9}$$

$$T_d=\frac{1+0.7\xi}{\omega_n} \tag{10}$$

$$T_p=\frac{\pi}{\omega_d}=\frac{\pi}{\omega_n\sqrt{1-\xi^2}} \tag{11}$$

$$M_p=e^{-\pi\xi/\sqrt{1-\xi^2}}\times 100\% \tag{12}$$

Where ξ is Damping ratio, ω_n is the Natural frequency, $P_{1,2}$ is the pole of the system, σ is pole on real axis, ω_d is pole on imaginary axis, T_r is rise time, T_s is settling time, T_d is delay time, T_p is peak time, M_p is maximum overshoot, and $\beta = \cos^{-1}\xi$.

4. KALMAN FILTER DESIGN

The Kalman filter has many utilizations in technology. The general purpose is to be a guidance for vehicle control and navigation systems. Moreover, the Kalman filter or linear quadratic estimation (LQE) is an algorithm to estimate the disturbed state variables of a dynamic system. In some cases, the Kalman filter can be combined with the state variable data obtained from the sensor, which makes the data of the state variable more accurate than a single measurement [12]. The Kalman filter iterates in a noisy data stream to get an estimate of the best system status in a probability perspective. The function of the filter generates some data and compares the filtered response to the actual plant response, as seen in Figure 4.

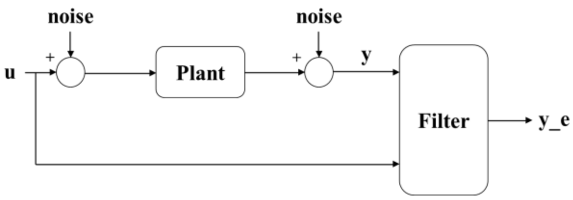


Figure 4 Structure of Kalman filter.

In this study, the continuous state space model was chosen as follow:

$$\dot{x}=Ax+Bu \quad (13)$$

$$y=Cx+Du \quad (14)$$

where x is the state vector, u is the input vector, A is the state transition matrix, B and C are the input and the output matrices, and D is the direct transition matrix which is typically zero. The state space model equation can also be approximated in a discrete-time system as,

$$x_k=Ax_{k-1}+Bu_{k-1}+w_{k-1} \quad (15)$$

where x_k is the state vector at k= 1, 2, 3, ..., n, u_k is the input vector, w_k is the system noise, v_k is the measurement noise. The output equation can be defined as

$$y_k=Cx_k+v_k \quad (16)$$

The Kalman filter algorithm has two steps including of prediction and correction steps. Both steps are applied in each k^{th} state. The equations of prediction are shown in equations (17)-(18), and the correction equations are defined as equations (19)-(21) [6].

$$\hat{x}_k^-=A\hat{x}_{k-1}+Bu_k \quad (17)$$

$$\overline{P}_k=AP_{k-1}A^T+Q \quad (18)$$

$$K_k=\overline{P}_kC^T\left(C\overline{P}_kC^T+R\right)^{-1} \quad (19)$$

$$\hat{x}_k=\hat{x}_k^-+K_k(z_k-C\hat{x}_k^-) \quad (20)$$

$$P_k=(I-K_kC)\overline{P}_k \quad (21)$$

where $\hat{x}$ and $\hat{x}^-$ are the estimated and predicted state vectors, respectively, and z is a measured value vector. K is a Kalman gain, P and $\overline{P}$ are the updated and predicted of the error covariance. I is a unit matrix, Q and R are the covariance matrixes of process noise and measurement or sensor noise, respectively.

5. EXPERIMENTAL SECTION

The activities are divided into two parts consisting of hardware and software implementation. The hardware part includes of a microcontroller, an actuator, and DC motor with encoder. The software part, which is implemented in Simulink, includes of sensors reading, noise filter, and PID controller. Figure 5 shows a block diagram for DC motor speed control.

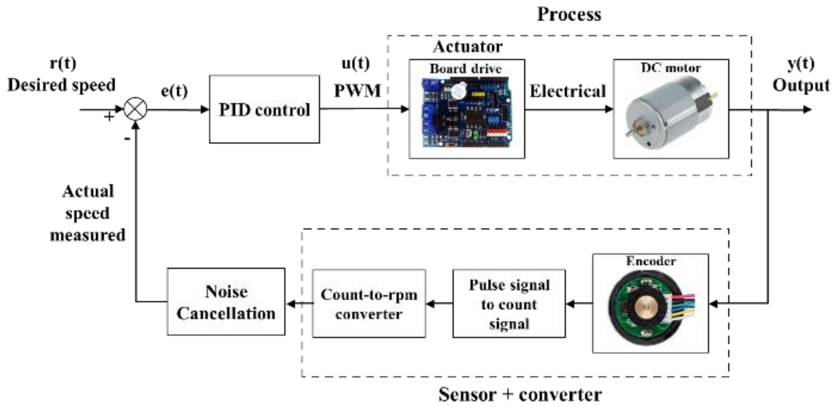


Figure 5 Block Diagram for DC motor speed control.

5.1 Implementation of Hardware

The microcontroller board used is Arduino Uno based on the microchip Atmega328P which has everything necessary to support a microcontroller. The Arduino Uno board was powered by electrical power through an analog to digital (A/D) converter for computer connection [13]. The L298P DC Motor Drive Shield acts as an actuator that allows to control the direction and speed of the DC motor, which is continuous rotation motors. The DC motors can be used with mechanical equipment that requires high speed (rpm) such as car wheels and fans. DC motor control by using a rapidly pulse switching technique can accurately measure the DC motor speed [14].

In this work, the DC motor is connected to the Simulink through Arduino Uno board that sends the input signal to the motor through the pulse width modulation (PWM) pin on Arduino Uno board and receives the output signal from the motor through the encoder [3]. The incremental encoder is a typical sensor that converts the movement of the shaft or angular position into digital or analog code, which will be further processed into data such as position and speed (rpm). In addition, it can specify the rotation direction of the encoder whether to rotate clockwise or counterclockwise. This study considers only the motor speed in one direction.

5.2 Implementation of Software

This experiment will create a model in MATLAB/ Simulink with the output and input sections. The output part is a block of signals from the encoder which is responsible for measuring the speed of the DC motor and converting it to revolution per minute (rpm). The input part is to create the desired speed block and the controller block.

5.2.1 Motor speed calculation from encoder

The encoder used provides the output signal format which is a characteristic of a square wave pulse signal. The number of pulses is related to the motor speed by connecting to the pulse meter. From figure 6, the pulse output from the encoder was counted by using “Rollover block” in Simulink with the sampling time (0.08 seconds). In this study, the encoder of gear motor has a reduction ratio of 21.3:1 (metal gearbox) providing a resolution of 55 counts per revolution. So, the counts per revolution (CPR) of the motor speed and gear ratio can then be calculated by using equations (22) and

(23) , respectively. The Gain block used for converting units is shown in Figure 7. Since the output signal has quite a lot of noise. The 1st order transfer function filter block was then applied here to reduce some noise to a certain extent. In this case, the 1st order filter used was defined as $\frac{1}{0.2s+1}$ by using the trial and error technique.

$$\text{CPR}=55\times21.3=1172 \tag{22}$$

$$\text{Gear ratio}=\frac{1}{1172} \tag{23}$$

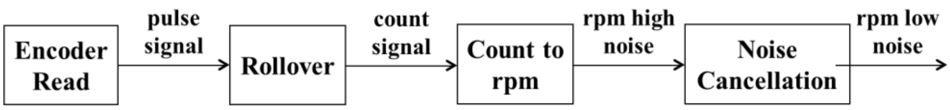


Figure 6 The structure of rpm conversion.

5.2.2 PI and PID Tuning

After being able to convert the pulse signal from the encoder to the motor speed in rpm. This step is a step test with the PWM signal being the input by a step from 45 and changing at 4 seconds to 255 which is sent to the Arduino Analog Write block set at pin 10 as shown in Figure 7.

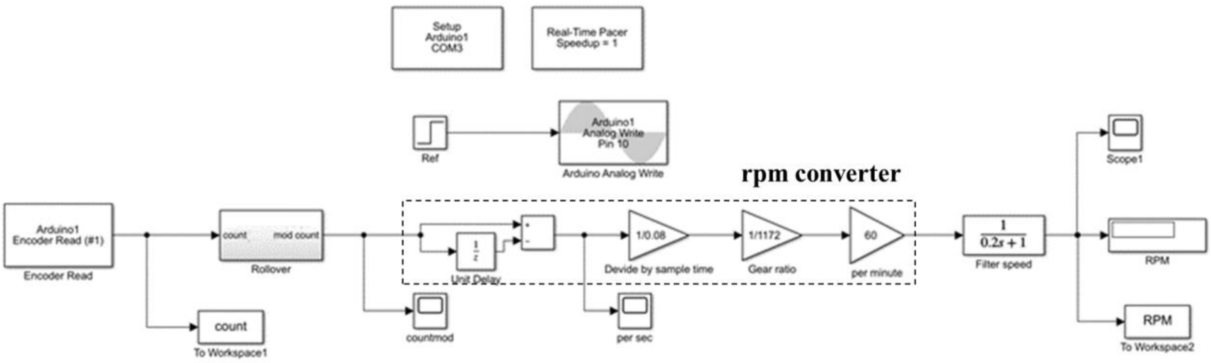
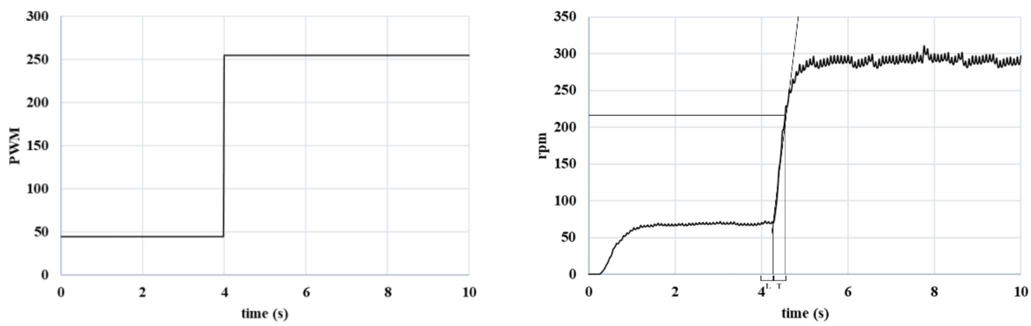


Figure 7 Simulink model of step test.



(a) (b)

Figure 8 (a) Step input and (b) The S-shaped step output response curve that time constant from 63.2%.

Figure 8 shows motor speed response in step change (from 45 to 255) of motor input (PWM) at time 4 sec, the motor speed was driven to the new steady-value of 290 rpm (from 67 rpm). From figure 8 (b), time delay (L) is 0.28 sec and process time constant (T) is 0.24 sec. Process gain (K_p) can be estimated by using equation (24). Therefore, the process model of the DC motor is obtained in equation (26). Then find the control parameters by using Ziegler-Nichols method and CHR method from Table 1 and Table 2, respectively.

$$K_P=\frac{\Delta output}{\Delta input} \qquad (24)$$

$$K_P=\frac{290-67}{255-45}=1.06 \qquad (25)$$

$$P(s)=\frac{1.06e^{-0.28s}}{0.24s+1} \qquad (26)$$

From Figure 1 and 2, the relationship between the input and the output in s domain but not considered delay term is given in the equation below.

$$\frac{y(s)}{r(s)}=\frac{C(s)P(s)}{1+C(s)P(s)} \qquad (27)$$

For PI controller substitute equations (2) and (26) in the equation (27) to get

$$\frac{y(s)}{r(s)}=\frac{\left[K_c+\frac{K_i}{s}\right]\left[\frac{1.06}{0.24s+1}\right]}{1+\left[K_c+\frac{K_i}{s}\right]\left[\frac{1.06}{0.24s+1}\right]} \qquad (28)$$

$$\frac{y(s)}{r(s)}=\frac{1.06(K_cs+K_i)}{0.24s^2+(1+1.06K_c)s+1.06K_i} \qquad (29)$$

The characteristics is

$$0.24s^2+(1+1.06K_c)s+1.06K_i=0 \qquad (30)$$

For PID controller substitute equations (4) and (26) in the equation (27) to get

$$\frac{y(s)}{r(s)}=\frac{\left[K_c+\frac{K_i}{s}+K_ds\right]\left[\frac{1.06}{0.24s+1}\right]}{1+\left[K_c+\frac{K_i}{s}+K_ds\right]\left[\frac{1.06}{0.24s+1}\right]} \qquad (31)$$

$$\frac{y(s)}{r(s)}=\frac{1.06\left(K_cs+K_i+K_ds^2\right)}{(0.24+1.06K_d)s^2+(1+1.06K_c)s+1.06K_i} \qquad (32)$$

The characteristics is

$$(0.24+1.06K_d)s^2+(1+1.06K_c)s+1.06K_i=0 \qquad (33)$$

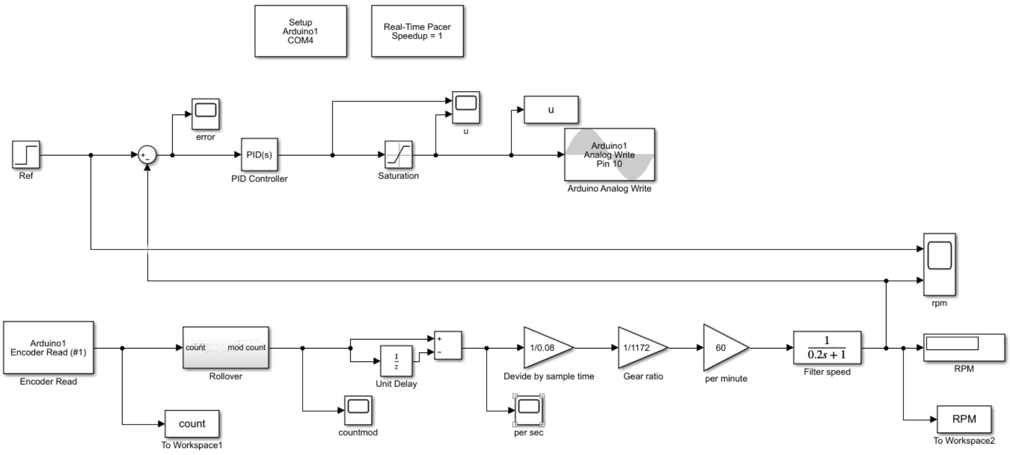


Figure 9 Controller model in Simulink.

In this study, the control specification was designed as follows.

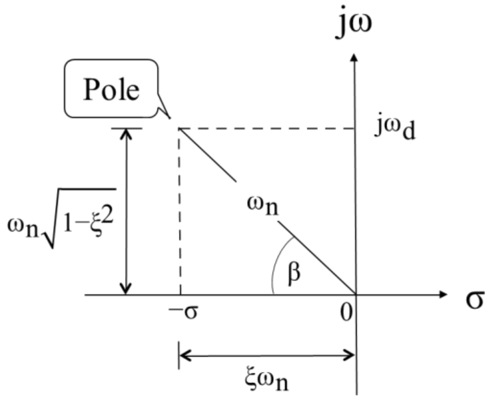
- K_d is equal to 0.15.
- Peak time is less than 2.5 sec: $T_p = \frac{\pi}{\omega_d} < 2.5$ so $\omega_d > 1.16$
- Settling time constant is less than 2.7 sec: $T_s = \frac{4}{\sigma} < 2.7$ so $\sigma > 1.33$
- Maximum overshoot is less than 45%: $\xi < 0.246$ so $\beta = \cos^{-1}\xi = 75.76^{\circ}$

Then took all these requirements to design to specify the pole positions as shown in Figure 10 (b).

Next, the speed control of DC motor is implemented via Simulink by adding the controller block as shown in Figure 9, where desired speed is 160 rpm. The controller in this study has three terms, the P term is proportional will result in decreasing the rise time (K_c), the I term is integral will result in the elimination of the error of the steady state (K_i), and the D term is derivative will result in increased system stability (K_d). The input signal used with the DC motor is the PWM cycle signal. The PWM is generated by the Arduino Analog Write set to pin10, which receives the input from 0-255. So, the saturation block must be specified in the range 0-255 only. In this research, to reduce the noise generated from the actual use and to make it easier and clearer to see the trend of the response, a simple filter (1st order transfer function filter) was added using randomized trial. And the motor is considered to spin in one direction only.

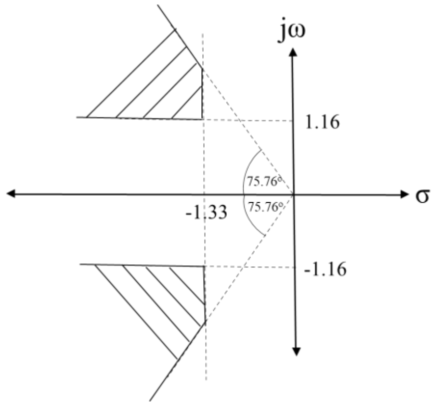
6. RESULTS AND DISCUSSION

6.1 PI and PID performances



(a)

Figure 10 (a) The region of pole location, (b) The region of pole location in this study.



(b)

Figure 10 (cont.) (a) The region of pole location, (b) The region of pole location in this study.

The pole position selection is selected in the shaded area and the response must meet the set requirements. From figure 10 (b), chose the closed-loop pole to equal -3±1.2j for PI controller and -1.8±1.2j for PID controller. Then find the parameters of PI and PID controllers from equations (30) and (33), respectively.

For PI
$$0.24s^2+(1+1.06K_c)s+1.06K_i = s^2+2\xi\omega_n s+\omega_n^2$$

So,
$$\frac{(1+1.06K_c)}{0.24}=2\sigma=6 \rightarrow K_c=0.42$$
$$\frac{1.06K_i}{0.24}=\sigma^2+\omega_d^2=10.44 \rightarrow K_i=2.36$$

For PID
$$(0.24+1.06K_d)s^2+(1+1.06K_c)s+1.06K_i = s^2+2\xi\omega_n s+\omega_n^2$$

So,
$$\frac{1+1.06K_P}{0.24+(1.06\times0.15)}=2\sigma=3.6 \rightarrow K_c=0.412$$
$$\frac{1.06K_I}{0.24+(1.06\times0.15)}=\sigma^2+\omega_d^2=4.68 \rightarrow K_i=1.76$$

The parameters of PI and PID controllers for the different tuning methods are shown in Table 4. From Figure 11. (a), shows the speed response of DC motor using PI controller by different tuning methods. For Ziegler-Nichols method (K_c= 0.77, K_i= 0.83) gives a good response performance with little overshoot. For CHR method (K_c=0.28, K_i=0.29) provides a good response performance with no overshoot to prevent damage. However, CHR has a very delayed response because of too small K_c. So, it is not appropriate to use in this study. When considering the pole placement method (K_c=0.42, K_i=2.36), it has low settling time (T_s) but has an excessively high overshoot of 67.5% while the requirement is no more than 45%. As this research requires the same specification for both PI and PID controllers, the K_c and K_i values are obtained from pole position selectors that are theoretically relevant and consistent. When choosing **σ** that leads to obtaining K_c, this **σ** is also used to find K_i, where this high K_i results in higher overshoot. Therefore, the PI controller based pole placement method does not meet the requirements.

From Figure 11. (b) and Table 5, the DC motor speed response using PID controller based on Ziegler- Nichols method (K_c=1.03, K_i=1.84, K_d=0.14) is highly aggressive and overshoot but tends to enter a steady state. For better response performance such as less rise time (T_r) and less overshoot, it can be tuned by decrease K_c to reduce rise time or increase K_i and K_d to eliminate error value and increase system stability, respectively. For CHR method (K_c= 0.48, K_i= 0.24, K_d=0.14) enters the slowest steady state due to no overshoot (0% OS). The requirements of the pole placement method are peak time is less than 2.5 sec, settling time is less than 2.7 sec and overshoot is less than 45%. The response tends to be conditional, which attains the fastest steady state and has an acceptable overshoot. From all the results, although the response of the DC motor quickly enters the steady state, the efficiency of the control is still not good enough to be used in this study. Therefore, efficiency and noise reduction were improved by using the Kalman filter instead of the 1st transfer function filter in the next step.

Table 4 Control parameters of PI and PID controller.

Controller	Method	K_c	K_i	K_d
PI	Ziegler-Nichols	0.77	0.83	-
	Chien-Hrones-Reswick	0.28	0.29	-
	Pole placement	0.42	2.36	-
PID	Ziegler-Nichols	1.03	1.84	0.14
	Chien-Hrones-Reswick	0.48	0.24	0.14
	Pole placement	0.412	1.76	0.15

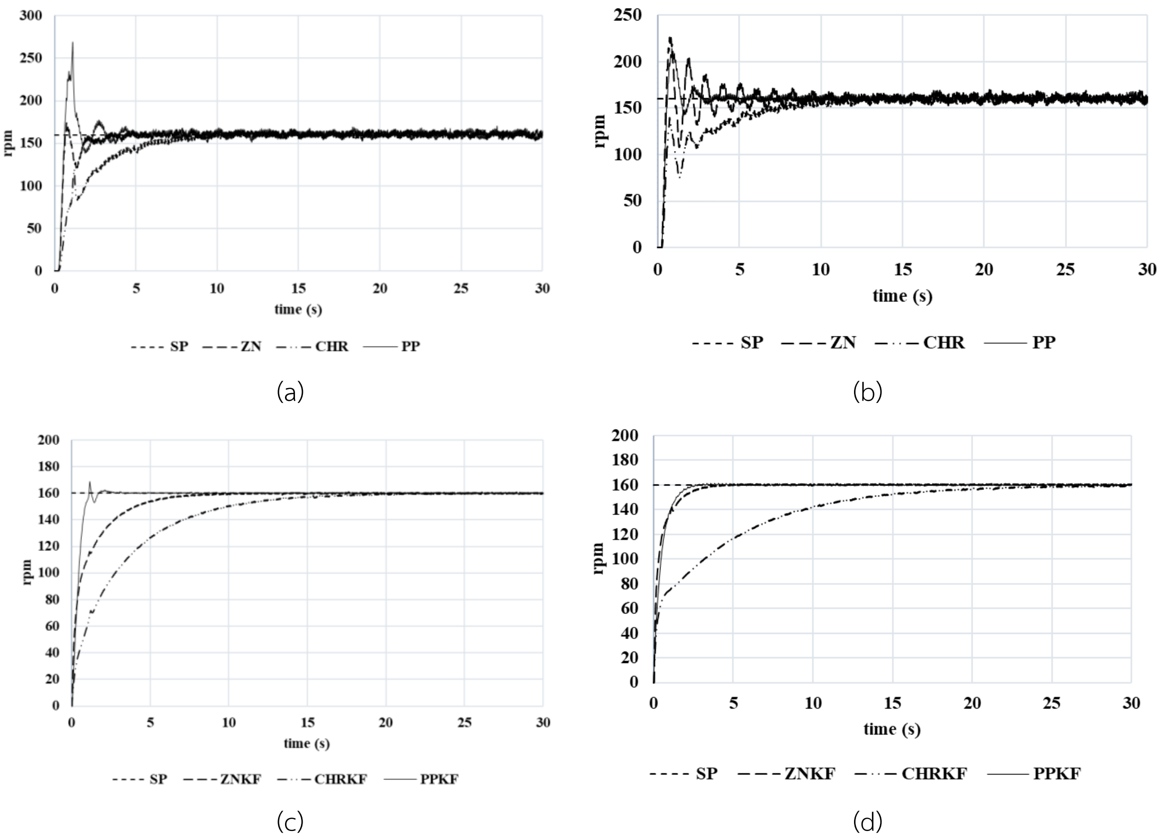


Figure 11 DC motor speed response of (a) PI without Kalman, (b) PID without Kalman, (c) PI with Kalman and (d) PID with Kalman.

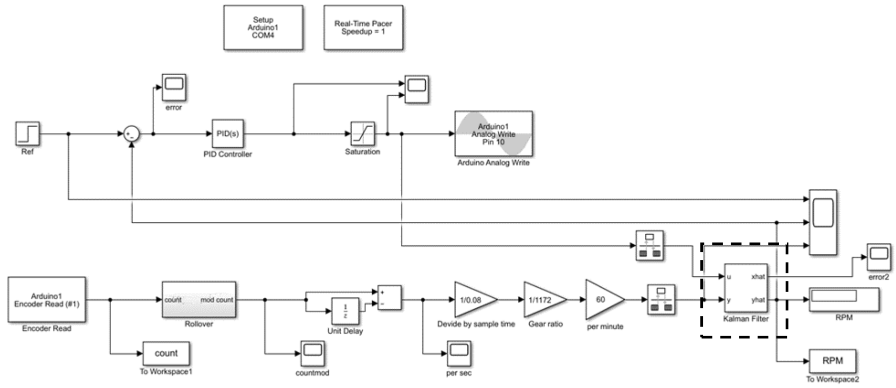


Figure 12 Controller model with Kalman filter in Simulink.

6.2 Controller with Kalman filter

Once the appropriate parameters are obtained. Next, filter the noise with Kalman filter. Since in the past it was reducing noise with 1st transfer function that the results were not good enough. The Kalman filter design should specify the following parameters or weight matrix through a trial-and-error process [12]. In this study, it is assumed that Q = 1 and R = 5. The Kalman filter block is replaced 1st transfer function block as shown in Figure 12.

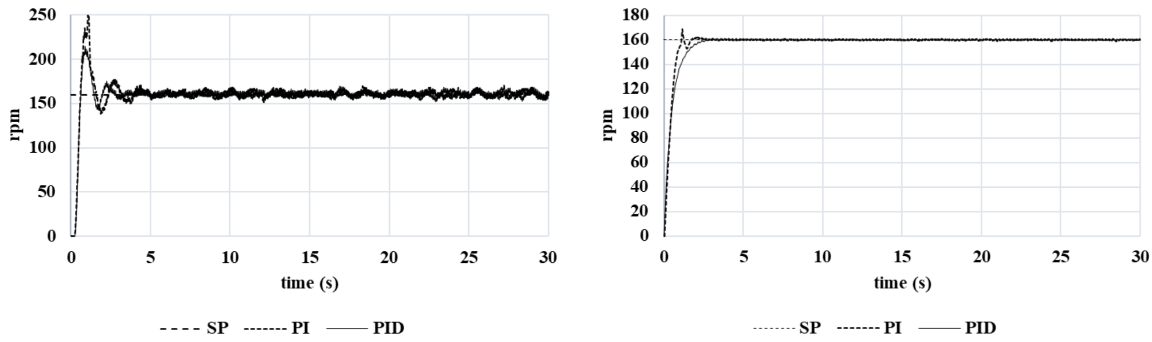
The Kalman filter can greatly reduce noise and eliminate overshoot for every tuning method. For PI controller using Ziegler- Nichols and CHR methods, when the Kalman filter is added, it reduces noise and overshoot, but the response is even more delayed. As for pole placement method, the Kalman filter is very good at eliminating overshoot but also has a non-smooth response as shown in Figure 11. (c). Considering the SAE value, the DC motor speed response using PI controller based pole placement method with Kalman filter gives the smallest error value. It is considered the best response compared to Ziegler-Nichols and CHR methods as shown in Table 5.

The results of the experiment as shown in Figure 11. (d) and Table 5, after using the Kalman filter instead of the 1st transfer function filter provides better performance. The Kalman filter is excellent in reducing noise and overshoot, which is obvious for Ziegler-Nichols and pole placement method. The PID controller based on pole placement method with Kalman filter provides the best response with minimal settling time (0.24 sec) and minimal overshoot (0 %OS). On the other hand, although the Kalman filter can actually reduce the noise, it results in a slower steady state for CHR method. Since the CHR method formula used in this study does not have overshoot (0% OS). When the Kalman filter was added, the overshoot was reduced from the original. Therefore, the settling time (T_s) was increased.

From Figure 13 shows the comparison of the responses of the PI and PID controllers. The PID controller offers a more efficient response than the PI controller, both without the Kalman filter and with the Kalman filter.

Table 5 Comparison between Ziegler-Nichols, CHR and pole placement response at 160 rpm.

Controller	Method	Rise time (s)	Maximum overshoot (%)	Settling time (s)	SAE
PI	ZN	0.62	8.75	3.24	8300
	CHR	8.75	0	8.75	18000
	PP	0.62	67.5	2.94	10000
	ZNKF	6.8	0	6.8	9000
	CHRKF	18	0	18	25000
	PPKF	1.24	3.75	1.68	4000
PID	ZN	0.52	39	8.5	11000
	CHR	11	0	11	18000
	PP	0.64	28	2.5	8000
	ZNKF	3	0	3	5000
	CHRKF	20	0	20	30000
	PPKF	2.24	0	2.24	4000



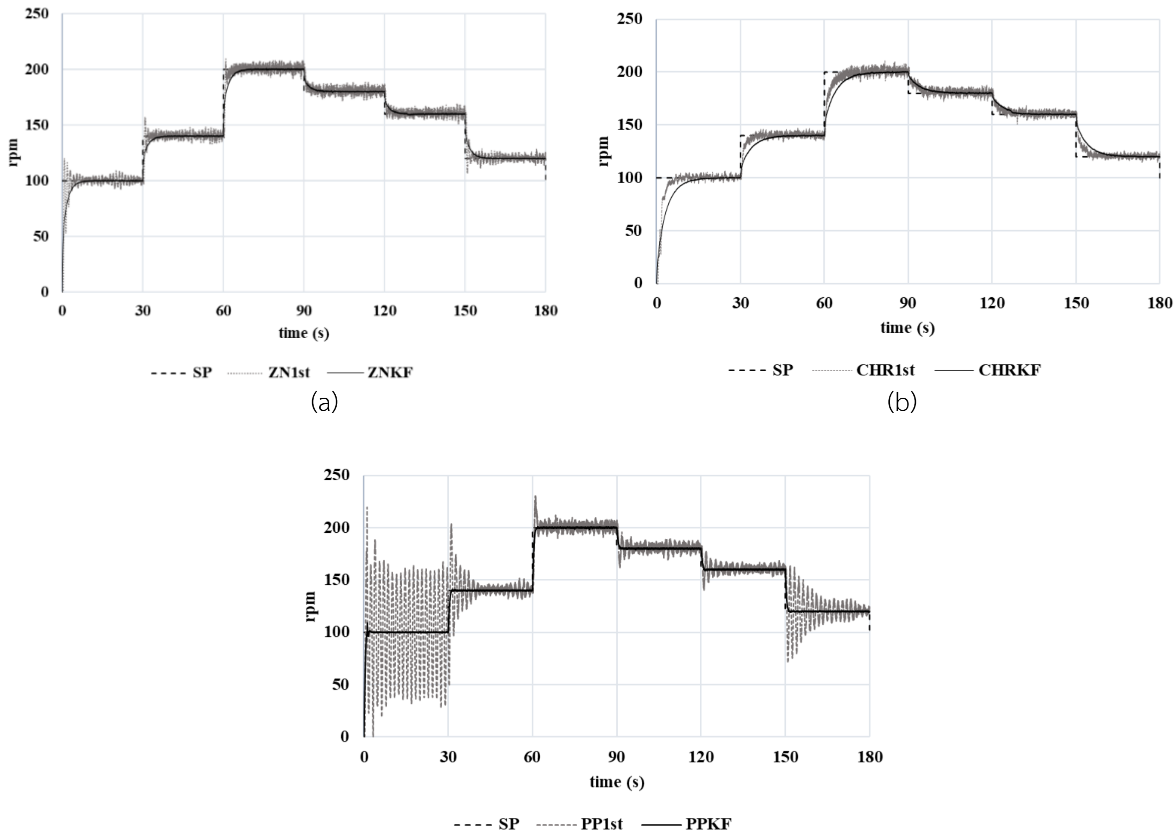
(a) (b)

Figure 13 Comparison between PI and PID (a) without Kalman filter, (b) with Kalman filter.

6.3 Set point tracking

This study considers only set point tracking. The objective of set point tracking is to equalize the set-point (SP) and process value (PV) while the controller is in manual mode. Therefore, when the controller is changed to automatic mode, it will start automatically without error (PV=SP).

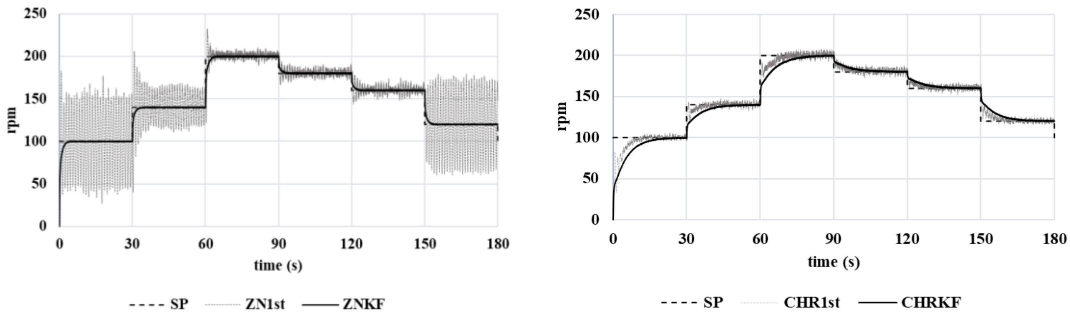
Figures 14 and 15 show a range of responses with different input ranges and up to 200 rpm. The dashed line is filtered using the first order transfer function. The solid line is filtered using Kalman filter that is the actual speed of the motor with the highly accurate response and steady state in seconds. For 1st order transfer function filter, the PI controller designs based on Ziegler-Nichols method and CHR method provide good performances at every desired value, but pole placement method only responds to certain desired values. For the PID controller designs based on CHR method and pole placement method provide good performances at every desired value but Ziegler-Nichols method only responds to certain desired values. For the Kalman filter both PI and PID controller give good responses and steady state error values close to zero for every desired value and every method. Considering only settling time (T_s), the pole placement method with Kalman filter gives the best response with settling time at 1.68 sec for PI controller and 2.24 sec for PID controller.



(a) (b)

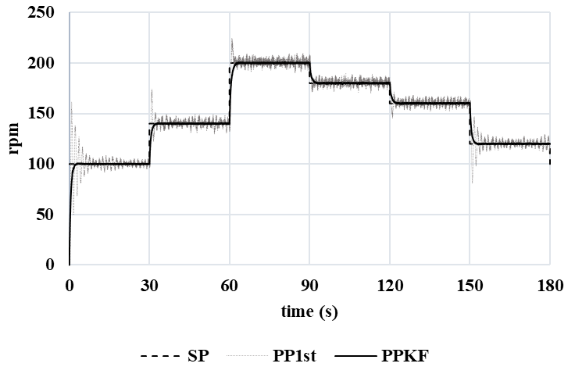
(c)

Figure 14 Ranges the output speed response of DC motor for PI (a) ZN vs ZNKF, (b) CHR vs CHRKF and (c) PP vs PPKF.



(a) (b)

Figure 15 Ranges the output speed response of DC motor for PID (a) ZN vs ZNKF, (b) CHR vs CHRKF and (c) PP vs PPKF.



(c)

Figure 15 (cont.) Ranges the output speed response of DC motor for PID (a) ZN vs ZNKF, (b) CHR vs CHRKF and (c) PP vs PPKF.

7. CONCLUSIONS

The experimental results show the control speed responses of DC motor using PI and PID controllers based on Ziegler-Nichols (ZN), Chien-Hrones- Reswick (CHR), and closed- loop pole placement tuning methods, which is implemented via MATLAB/Simulink. PI controller is often suitable for controlling first-order plant. On the other hand, PID controller is suitable for two or higher order plants. The control responses based all three tuning methods tend to enter a steady state. In case of the 1st order transfer function filter, the PI controller based Ziegler-Nichols method gives the best response compared to the PID controller with a very aggressive response with unacceptable speed-overshoot especially in low range of the motor speed. For better responses in a range of those speeds, it can be decreased K_c value or increase K_i and K_d to compensate the aggression. However, the PI and PID controller based CHR method give responses without overshoot (0% OS). Although they can be controlled at all the desired speed ranges, the slow performances are observed significantly. Then this method is suitable for processes that do not require oscillations or overshoot. The PID controller based pole placement method gives a better response than PI controller with acceptable overshoot and satisfies the specified conditions. In this study, the Kalman filter block was used to eliminate noise and overshoot by replacing the 1st order transfer function filter block. It is obvious that the Kalman filter can reduce the noise well and increase the response efficiency. It means that when you have a Kalman filter you can control the speed in every desired speed range. The objective of this study is to respond quickly without overshoot to prevent motor damage. Therefore, the final result shows the PID controller with the Kalman filter is the best and fastest response without overshoot, which is the most suitable to be used for this study.

REFERENCES

[1] Chau, K.T., Ye, S., Gao, Y. and Chen, J.H. Application of Chaotic-Motion Motors to Industrial Mixing Processes. *Institute of Electrical and Electronics Engineers,* 2004; 3: 1874-1880.

[2] Loucif, F. DC Motor Speed Control Using PID Controller. International Conference on Control, Automation and Systems, June 2-5, KINTEX, Gyeonggi-Do, Korea, 1–5.

[3] Vikhe, S. P., Punjabi, N. and Kadu, B. C. DC Motor Speed Control Using PID Controller in Lab View. *International Journal of Innovative Science and Modern Engineering*, 2015; 3: 38–41.

[4] Hamoodi, S.A., Mohammed, R.A. and Salih B.M., DC Motor Speed Control Using PID Controller Implementation by Simulink and Practical. *International Journal of Electrical Engineering*, 2018; 11(1): 39–49.

[5] Husnaini, T., Krismadinata, Asnil and Hastuti. PI and PID Controller Design and Analysis for DC Shunt Motor Speed Control. *International Journal of Recent Technology and Engineering*, 2019; 8(4): 144-150.

[6] Kumar, R. and Murthy, R. Kalman Filter for Speed Control of DC Motor for Robotic Safety Critical Application. 2nd National Conference on Emerging Trends in Science and Technology, May 12, Sapthagiri College of Engineering, Bangalore, 2015.

[7] Abdulameer, A., Sulaiman, M., Aras, M. and Saleem D. Tuning Methods of PID Controller for DC Motor Speed Control. *Electrical Engineering and Computer Science*, 2016; 3(2): 343 – 349.

[8] Allam, T., Raju, M. and Kumar, S.S. Design of PID controller for DC Motor Speed Control Using Arduino Microcontroller. *International Research Journal of Engineering and Technology,* 2016; 3(9): 791–794.

[9] Ali, M.A., Mohammed, A.H. and Alwan, H.M. Tuning PID Controllers for DC Motor by Using Microcomputer. *International Journal of Applied Engineering,* 2019; 14(1): 202–206.

[10] Wade, H. L. Basic and Advanced Regulatory Control: System Design and Application. The Instrumentation, System and Automation Society, North Carolina, 2004.

[11] Shabaan, A.R., El-Metwally, A.M., Farghaly, M.M.A. and Sharawi, A.A. PID and Fuzzy Logic Optimized Control for Temperature in Infant Incubators. *International Conference on Modelling, Identification and Control*, August 31-September 2, Cairo, Egypt, 2017; 53-59.

[12] Gaeid, K.S. Optimal Gain Kalman Filter Design with DC Motor Speed Controlled Parameters. *Journal of Asian Scientific Research*, 2013; 3(12): 1157–1172.

[13] Maung, M.M., Latt, M.M. and New, C.M. DC Motor Angular Position Control using PID Controller with Friction Compensation. *International Journal of Scientific and Research Publications*, 2018; 8(11): 149–155.

[14] Javiya, S.N. and Kumar, A. Comparisons of Different Controller for Position Tracking of DC Servo Motor. *International journal of Advance Research in Electrical, Electronics and Instrumentation Engineering*, 2016; 5(2): 966-974.