Performance analysis of PID control in DC Brushless motor using trial and error method

K Sekarsari* and T Tata
Department of Electrical Engineering – Pamulang University, South Tangerang, Banten Indonesia

*kartika.sekarsari.03@gmail.com

Abstract. This research presents an overview of the performance of the PID control technique applied to the Brushless DC motor to regulate the speed using the Arduino microcontroller and LabView software. Brushless DC motors are chosen because they have high efficiency both in performance and longer durability. Kp, Ti and Td parameter tuning techniques use the Trial and error method and the results obtained from the value of Kp = 0.050, Ti = 0.001 and Td = 0.001 indicate a minimum error compared to controls without using PID. The percentage of overshoot still exists but the rise time and settling time indicate a faster response speed to the setpoint.

1. Introduction
Brushless DC motors are included in the type of permanent magnet synchronous motor. In this type of motor, permanent magnets are found in the rotor, while winding is in the stator to produce a magnetic field. The magnetic field produced by the rotor and stator will rotate at the same frequency. Therefore, the Brushless DC Motor is an electric engine that drives electric cars and electric vehicles. DC Brushless motors are also widely used in industry as servo, drive and variable speed applications with precise motion control and stable operation in industrial processes. The advantage of Brushless DC motors in the industry is huge: it can save costs and time in almost all industries. Compared to DC motors, Brushless DC motors have better performance in terms of higher efficiency, higher torque under the low speed range, higher power density, lower maintenance, and lower noise levels [1,2]. Changing the polarity of a BLDC motor is done electronically using a hall-effect sensor or a rotary encoder [3].

In the use of DC Brushless motors sometimes desired rotation that can be changed in accordance with the load rotation with a smooth displacement setting. It aims to reduce the amount of current when starting, damping vibrations and mechanical pounding at start. Therefore, there are many ways to regulate the speed of the motor, one of which is the PID control, which is a control technique that consists of Proportional, Integral and Derivative controls [5].
In this paper PID control is applied by tuning using trial and error methods to regulate the speed of a Brushless DC motor. The controller equipment uses Arduino UNO and the monitoring system uses LabView 2018 software.

2. PID controller

PID controllers are widely used because of the simplicity of use and strong operation. The only drawback of using a PID controller is that it needs to be adjusted for optimal performance [6]. PID Controller has a parameter that is mutually combined, namely P (Proportional) Controller, D (Derivative) Controller, and I (Integral) Controller [7]. Figure 1 shows of PID control systems.

![Figure 1. PID controlled system [8].](image)

The equation can be written as follows:

\[
U_{PID} = K_P e(t) + K_I \int_0^t e(t) dt + K_D \frac{de}{dt}
\]  

Where \(u(t)\) is the control signal (control), \(K\) is the gain, \(e(t)\) is the error signal obtained from the difference between the value of the output signal - setpoint signal [7].

In this case, Proportional controller (P) has an output that is proportional to the magnitude of the error signal (the difference between the desired quantity and the actual price). The Integral controller (I) functions to produce a system response that has a zero steady state error. Differential Controller (D) is generally used to speed up the initial response of a system, but does not minimize errors in steady state [9]. The PID controller can be adjusted by various methods. The most commonly used closed-loop manual tuning method is tuning Ziegler Nichols [10].

3. Design of systems

Block diagram of the system shown in Figure 2, consists of hardware and software devices. While prototype of PID control in BLDC motor shown in Figure 3.

![Figure 2. Block diagram of the system.](image)  

![Figure 3. Prototype of PID control.](image)
As shown in Figure 2, the system parts consist of: the optical encoder sensor functions to detect motor rotation speed, PC to run LabVIEW software and program components, Arduino functions as a controller to control the device in accordance with the program written on it, and The L298N driver functions to regulate the speed and direction of rotation of the motor.

The system flow diagram is shown in Figure 4. Initialization, entering the set point (SP) value in the Lab View front panel in the form of the voltage input value used to supply the microcontroller. Trial and error tuning is done to find the parameters kp, ki and kd with the smallest or near zero error. In this case, the Setpoint is determined in advance with the desired motor speed value.

Next, enter the best parameters of the experiment results into LabView. Arduino will read the data set point from LabView software, will respond to the data, and receive information from the encoder according to the program written on it. The PID calculation process which is done through Lab View software is then forwarded to Arduino to then send PWM signals to the motor driver to drive the DC motor. If the set point value matches the controller output shown by the RPM meter, the motor will rotate according to the set point input. The motor movement will be detected by the encoder which then sends a signal as feedback. All output signals will be displayed in the LabView front panel screen. Labview of the system is shown in Figure 5.

4. **Results and discussion**

Some of the tests conducted in this paper consist of: voltage to speed testing, PWM testing to voltage, error percentage, system testing without PID controller, and system testing with PID controller.

4.1. *Voltage testing to a speed and PWM testing to voltage*

Table 1 shows the voltage measurements on the L298N motor driver using a digital voltmeter. Testing is done by running the Labview program that has been made and entering a setpoint value in stages without motor load based on the data sheet that was made.
### Table 1. Voltage test data for speed.

| No | Voltage (V) | RPM | No | Voltage (V) | RPM |
|----|-------------|-----|----|-------------|-----|
| 1  | 0           | 0   | 27 | 5.8         | 900 |
| 2  | 0           | 0   | 28 | 6           | 921 |
| 3  | 0           | 0   | 29 | 6.2         | 942 |
| 4  | 0.1         | 0   | 30 | 6.4         | 962 |
| 5  | 0.3         | 0   | 31 | 6.6         | 979 |
| 6  | 0.5         | 0   | 32 | 6.8         | 995 |
| 7  | 1.3         | 0   | 33 | 7           | 1012|
| 8  | 1.5         | 0   | 34 | 7.1         | 1026|
| 9  | 1.8         | 41  | 35 | 7.3         | 1040|
| 10 | 2           | 110 | 36 | 7.5         | 1054|
| 11 | 2.3         | 170 | 37 | 7.7         | 1065|
| 12 | 2.5         | 234 | 38 | 7.9         | 1068|
| 13 | 2.7         | 299 | 39 | 8.1         | 1079|
| 14 | 3.1         | 363 | 40 | 8.3         | 1089|
| 15 | 3.2         | 427 | 41 | 8.5         | 1097|
| 16 | 3.4         | 485 | 42 | 8.7         | 1106|
| 17 | 3.6         | 540 | 43 | 8.9         | 1114|
| 18 | 3.8         | 594 | 44 | 9.1         | 1122|
| 19 | 4           | 637 | 45 | 9.2         | 1129|
| 20 | 4.2         | 682 | 46 | 9.4         | 1137|
| 21 | 4.4         | 721 | 47 | 9.6         | 1145|
| 22 | 4.6         | 755 | 48 | 9.8         | 1152|
| 23 | 4.8         | 790 | 49 | 9.9         | 1162|
| 24 | 5           | 822 | 50 | 10.1        | 1173|
| 25 | 5.2         | 850 | 51 | 10.2        | 1225|
| 26 | 5.4         | 878 | 52 | 10.4        | 1275|

### Table 2. PWM test data for voltage.

| No | PWM | Voltage (V) | No | PWM | Voltage (V) |
|----|-----|-------------|----|-----|-------------|
| 1  | 0   | 0           | 15 | 131 | 5.4         |
| 2  | 9   | 0           | 16 | 140 | 5.8         |
| 3  | 18  | 0.2         | 17 | 150 | 6.2         |
| 4  | 28  | 1.3         | 18 | 159 | 6.8         |
| 5  | 37  | 1.8         | 19 | 168 | 7.2         |
| 6  | 46  | 2.2         | 20 | 178 | 7.7         |
| 7  | 56  | 2.6         | 21 | 187 | 8.1         |
| 8  | 65  | 3           | 22 | 197 | 8.5         |
| 9  | 75  | 3.4         | 23 | 206 | 8.8         |
| 10 | 84  | 3.8         | 24 | 215 | 9.1         |
| 11 | 93  | 4.2         | 25 | 225 | 9.5         |
| 12 | 103 | 4.5         | 26 | 234 | 9.8         |
| 13 | 112 | 4.7         | 27 | 243 | 10          |
| 14 | 122 | 5.3         | 28 | 255 | 10.4        |

Table 2 shows the PWM testing to the voltage, the testing process is carried out by measuring the 12 VDC input voltage from the L298N motor driver as the motor drive. Then compared with the PWM.
value given in the range of 0-255. In this case, when the PWM value increases, the voltage also rises, so it can be concluded that the voltage regulation based on the given pulse width is successfully carried out. At a maximum PWM of 255 or 100% duty cycle shows the result of a voltage of 10.4 VDC, this is caused by a decrease in the voltage that occurs in the motor driver when the supply voltage is 12 VDC.

4.2. Percentage of error
Table 3 shows percentage of errors between measurements using a tachometer and data readings made by the encoder sensor. The percentage of error itself is done with the following calculation:

\[
\text{The percentage of error} = \frac{\text{tachometer RPM} - \text{encoder RPM}}{\text{encoder RPM}} \times 100\% \tag{2}
\]

| No | Speed (RPM) | Difference (RPM) | Percentage % |
|----|-------------|------------------|--------------|
| 1  | 0           | 0                | 0            |
| 2  | 50          | 0                | 0            |
| 3  | 100         | 35.4             | 2.1          | 5.93 |
| 4  | 150         | 144.2            | -6.7         | -4.65 |
| 5  | 200         | 198.6            | 1.4          | 0.7  |
| 6  | 250         | 251.5            | -1.5         | -0.6 |
| 7  | 300         | 295.2            | 4.8          | 1.63 |
| 8  | 350         | 346.7            | 3.3          | 0.95 |
| 9  | 400         | 397.1            | 2.9          | 0.73 |
| 10 | 450         | 446.1            | 3.9          | 0.87 |
| 11 | 500         | 496.5            | 3.5          | 0.7  |
| 12 | 550         | 534.6            | 2.9          | 0.54 |
| 13 | 600         | 584.2            | 3.3          | 0.56 |
| 14 | 650         | 634.5            | 3            | 0.47 |
| 15 | 700         | 674.4            | 0.6          | 0.09 |
| 16 | 750         | 734.3            | 3.2          | 0.44 |
| 17 | 800         | 743.2            | 6.8          | 0.91 |
| 18 | 850         | 789.6            | -2.1         | -0.27|
| 19 | 900         | 826.5            | 11           | 1.33 |
| 20 | 950         | 874.1            | 0.9          | 0.1  |
| 21 | 1000        | 935.6            | 1.9          | 0.2  |
| 22 | 1050        | 970.6            | 4.9          | 0.5  |
| 23 | 1100        | 995.3            | 4.7          | 0.47 |
| 24 | 1150        | 1044             | 6            | 0.57 |
| 25 | 1200        | 1115             | 10           | 0.9  |
| 26 | 1250        | 1174             | 1            | 0.09 |
| 27 | 1300        | 1186             | 14           | 1.18 |
| 28 | 1350        | 1230             | 8            | 0.65 |
| 29 | 1360        | 1240             | 10           | 0.81 |
Figure 6 Graphic display of setpoint with encoder sensor.

As shown in Table 3 there is still an error difference between the encoder sensor with measurements using a tachometer in the range of 0 to 5%. The difference between setpoint and encoder sensor readings in table is still quite high because the setpoint value still uses the maximum motor speed with 12V DC supply. When the test was carried out it turned out that the ability of the motor driver L298N only reached a voltage of 10.4 VDC.

In Figure 6 shows, the maximum setpoint value of 1360 rpm DC motor will create a voltage drop on the motor driver. Therefore, the setpoint value must be adjusted to the maximum capability of the driver at 10.4 VDC, i.e. at an average motor speed value of 1275 rpm.

4.3. Control testing without PID

The BLDC motor control test is done by running the LabView program that was created without turning on the PID switch, testing the speed value input at the setpoint by stepwise from the smallest value of 0 rpm up to the maximum speed value of the previous calibration result which is 1275 rpm, with multiple data collection range 50 rpm to see the control accuracy. Test results are shown in Table 4.

| No | SP (RPM) | PV (RPM) | Error (SP-PV) | No | SP (RPM) | PV (RPM) | Error (SP-PV) |
|----|----------|----------|---------------|----|----------|----------|---------------|
| 1  | 0        | 0        | 0             | 15 | 700      | 737      | -37           |
| 2  | 50       | 0        | 5             | 16 | 750      | 775      | -25           |
| 3  | 100      | 0        | 100           | 17 | 800      | 837      | -37           |
| 4  | 150      | 157      | -7            | 18 | 850      | 887      | -37           |
| 5  | 200      | 212      | -12           | 19 | 900      | 925      | -25           |
| 6  | 250      | 275      | -25           | 20 | 950      | 975      | -25           |
| 7  | 300      | 325      | -25           | 21 | 1000     | 1025     | -25           |
| 8  | 350      | 375      | -25           | 22 | 1050     | 1088     | -38           |
| 9  | 400      | 425      | -25           | 23 | 1100     | 1125     | -25           |
| 10 | 450      | 475      | -25           | 24 | 1150     | 1175     | -25           |
| 11 | 500      | 537      | -37           | 25 | 1200     | 1212     | -12           |
| 12 | 550      | 587      | -37           | 26 | 1250     | 1248     | 2             |
| 13 | 600      | 637      | -37           | 27 | 1275     | 1250     | 25            |
| 14 | 650      | 687      | -37           |
Figure 7. Curve of setpoint to variable process without PID control.

From the data in Table 4, it can be seen that the error value is still quite large, namely the speed value is still above the setpoint value. Therefore, the best error value is obtained from the setpoint (SP) value is reduced by the process value (PV) as the output of the control without PID. Figure 7 shows the results curve from Table 4.

| Type  | Control | Kp  | Ti     | Td  |
|-------|---------|-----|--------|-----|
| P     | T/L     | ~   | 0      |     |
| PI    | 0,9 T/L | L/0,3 | 0     |     |
| PID   | 1,2 T/L | 2 L | 0,5 L  |     |

As a reference to find the right PID parameters, in Figure 7 the first Ziegler Nichols curve tuning method is used. By drawing a tangent to the turning point of the s-shaped curve, two constants will be obtained consisting of a time delay L and a time constant T. Furthermore, the parameters Kp, Ti and Td are determined from the following Table 5. The initial calculation results are obtained value of Kp = 0.039 Ti = 0.008 and Td = 0.002. When the test is applied to the setpoint value of 1000 rpm, the results obtained are an error of 62.5 at rise time and reach a setpoint value with a considerable amount of time.

4.4. The testing with PID control

Trial and error tuning methods to get the parameters Kp, Ti, Td are done in real time using the Labview program connected to the Brushless DC motor.
Figure 8. Curve results of the PID control at 1000 rpm setpoint with parameters $K_p = 0.050$, $T_i = 0.001$, $T_d = 0.001$ using trial and error method.

The $K_p$, $T_i$, and $T_d$ parameters are entered into the Labview program in stages in several experiments to find the best parameters with a data capture time of 5 seconds. From this experiment the parameter values obtained $K_p = 0.050$, $T_i = 0.001$ and $T_d = 0.001$ that meet the smallest error and the closest setpoint.

Furthermore, re-testing is done by running the program by entering the three PID parameters to see the actuator's response in the form of overshoot, increase in time, set time, and error value. Testing with 1000 rpm setpoint value, shown in Figure 8. This PID parameter value has a calibrated setpoint capability to the maximum motor driver voltage, which is 0 to 1275 rpm with a speed range of 50 rpm. Has an error difference between 0-21 rpm and the fastest time setting value between 0 to 11 milliseconds.

5. Conclusion
From the results of the research it was found that the technique of tuning the parameters $K_p$, $T_i$ and $T_d$ using the Trial and error method produces the parameter values $K_p = 0.050$, $T_i = 0.001$ and $T_d = 0.001$. This Parameter has the ability to test setpoint values that are calibrated to the maximum voltage of the motor driver, which is 0 to 1275 rpm with a speed range of 50 rpm. The fastest value setting time ranges from 0 to 11 ms and has the difference in error value in the smallest steady state conditions.

References
[1] Varshney A, Gupta D and Dwivedi B 2017 Speed response of brushless DC motor using fuzzy PID controller under varying load condition J. Electr. Syst. Inf. Technol. 4 310–21
[2] Pillay P and Krishnan R 1989 Modeling, simulation, and analysis of permanent-magnet motor drives. II. The brushless DC motor drive IEEE Trans. Ind. Appl. 25 274–9
[3] Jatmiko J, Basith A, Ulinuha A, Muhlasin M A and Khak I S 2018 Analisis Performa Dan Konsumsi Daya Motor Bldc 350 W Pada Prototipe Mobil Listrik Ababil Emit. J. Tek. Elektro 18 14–7
[4] Masudi N 2015 Desain Controller Motor Bldc Untuk Meningkatkan Performa Output (Daya) Pada Sepeda Motor Listrik (Institut Teknologi Sepuluh Nopember)
[5] Hibban H 2015 Desain Kontroler Fuzzy PID Gain Scheduling Untuk Pengaturan Kecepatan Motor DC Tanpa Sikat
[6] Pandit A and Hingu B 2018 Online Tuning of PID controller using Black Box Multi-Objective
Optimization and Reinforcement Learning *IFAC-PapersOnLine* **51** 844–9

[7] Wiranto W, Setiawan B I and Saptomo S K Sistem Kontrol Irigasi PID *J. Keteknikan Pertan.* **2** 21963

[8] Paz R A 2001 The design of the PID controller *Kllipsch Sch. Electr. Comput. Eng.* **8**

[9] Arindya R 2016 Penelaan kendali PID untuk pengendali proses *PROSIDING SEMINAR NASIONAL CENDEKIAWAN*

[10] Gowthaman E, Vinodhini V, Hussain M Y, Dhinakaran S K and Sabarinathan T 2017 Speed control of permanent magnet brushless DC motor using hybrid fuzzy proportional plus integral plus derivative controller *Energy Procedia* **117** 1101–8