Advanced PID Simulation for DC Motor using Scilab

Jimmy Linggarjati
Computer Engineering Department, Faculty of Engineering, Bina Nusantara University, Jakarta, Indonesia, 11480
Email: jimmyl@binus.edu

Abstract. DC motor is simulated in this paper, in which Scilab simulator is used. The most advantage of using Scilab, as a simulation tool is that it is an open source software. Thereby it is very suitable for our laboratory condition, where budgets are very tied and our students would certainly could not afford a proprietary simulation software such as Matlab, i.e. already well-known in academia and industry. In this paper, a full feature PID Model, namely advanced PID Model is described and simulated, to introduce the importance of PID control in industrial. By understanding these advances in PID control, students are aware of their usage as a prime mover in industrial world. This paper starts with anti-windup, command weightings, derivative term replacement, and autotuning relay-feedback using descriptive function, that finally form a full feature advanced PID model. By using this advanced PID model, it will promote green technology for industrial applications, by consuming fewer energy due to highly efficient design.

Keywords: DC Motor, Scilab, Simulation, advanced PID, autotuning relay, anti-windup, green technology.

1. Introduction
In modern society, unconsciously we are surrounded by automation that make our products and needs. In the industrial world, a computer engineering students should have a firm ground of understanding about how a motor works and how it is being controlled. Using an already known DC motor characteristic as shown in Table 1, we can develop a model in Scilab, as shown in Figure 1.

Table 1. DC Motor characteristics

| Name            | Symbol | Values     |
|-----------------|--------|------------|
| Resistance      | R      | 5 ohms     |
| Inductance      | L      | 10 mH      |
| Voltage Constant| \( K_v \) | 0.065 V/rad/s |
| Torque Constant | \( K_t \) | 0.065 N.m/A   |
| Total Inertia   | J      | 3 kg.m²    |
| Friction        | B      | 0.1 N.m/rad/s |
Because the time constant of J/B (30) is larger than the electrical time constant L/R (0.002), we can neglect the L/R constants.

\[ s^2 + \frac{(B+K_d)}{J} s + \frac{K_p}{J} = \frac{s^2}{J} + 2\xi\omega_s + \omega^2 \quad (1) \]

which gives,

\[ K_p = \omega^2 J, \quad K_d = 2\xi\omega J - B \quad (2) \]

By selecting critical-damped system (\(\xi = 1\)), Table 1 shows \(K_p\) and \(K_d\) values, for a 3 given natural frequencies.

**Table 2. Calculated \(K_p\) and \(K_d\) for 3 natural frequency**

| \(\omega\) | \(K_p\) | \(K_d\) |
|--------|--------|--------|
| 4      | 48     | 24     |
| 8      | 192    | 48     |
| 12     | 432    | 72     |

**Figure 1.** PID with disturbance

**Figure 2.** Output PID with disturbance
There is a steady-state error (see Figure 2). In order to eliminate this steady-state error, $K_i$ is applied. Figure 3 shows that the steady-state error is reduced, but at a cost of overshoot. Once the $K_i$ is used, the closed loop system is now in third order, as shown in (3), with characteristic polynomial in (4)

$$\theta(s) = \frac{K_ds^2+K_ps+K_i\theta_d(s)}{\lambda(s)} \frac{s}{\lambda(s)} D(s)$$ (3)

$$\lambda(s) = js^3 + (B + K_d)s^2 + K_p s + K_i$$ (4)

By applying Routh-Hurwitz criteria, the feedback system is stable if

$$K_i < \frac{(B+K_d)K_p}{j}$$ (5)

By choosing $K_p = 432$ and $K_d = 72$ (based on Table 2) for $\omega=12$, Figure 3 shows how the step responses behave in PID control.

![Figure 3. K_i Eliminating Steady-state Error](image)

If $K_i$ is bigger than 432, than the steady-state error is eliminated, but the overshoot is higher.

2. **Advanced Simulation Model of DC Motor**

In real world application of DC Servo motor, Figure 1 has some drawbacks, listed as follows:

1. Unknown DC Motor Specifications
2. Saturation
3. Derivative Term Replacement [1]
4. Command Weightings

2.1. **Unknown DC Motor Specifications**

A real DC Motor device might not have listed their electrical and mechanical specifications. Therefore, there is a need to do auto-tuning for the PID terms [2]. Fortunately, a technique called descriptive function has defined a mechanism called relay-feedback, to calculate PID terms using ZNFD method [7].

Figure 4 shows the autotuning model, it uses a SIGN block, consult [3] for further study on autotuning. Figure 5 shows the sinusoidal oscillation at the output. The voltage peak (a) is 0.45,
therefore using equation (6), we get $K_u = 282$, while $T_u = 0.7$ (by observing one period of sinusoidal signal). And then using ZNFD method [4], $K_p = 0.6K_u = 169$, $K_i = 1.2K_u/T_u = 483$, and $K_d = 0.075K_uT_u = 15$. Setting the PID terms, we get the result shown in Figure 5.

$$K_u = N(a) = \frac{Ad}{\pi a} \quad (6)$$

**Figure 4.** Relay-feedback for autotuning PID model

**Figure 5.** Output Oscillation due to Relay Signal
2.2. Saturation
Due to non-linear components in real-world DC Motor driver, the feedback loop is affected, especially in the integrator component. This is known as “integrator-windup” effect. Figure 6 simulated how a limiting input component has effect on the output responses (shown in Figure 7 and 8).

![Figure 6. Scilab model for comparing saturation effect](image)

![Figure 7. Effect of Saturation in Output Response](image)

![Figure 8. Effect of Saturation in Input DC Motor Plant](image)
2.3. Derivative Term Replacement

One disadvantage of Derivative term, is that it amplifies high frequency noise. Suppose, at the controller input, we have some noise with unity amplitude at frequency 1000 rad/sec.

\[ \hat{e}(t) = \sin(1000t) \]  

(7)

With \( K_d=1 \), this noise is amplified 1000 times.

\[ \hat{u}(t) = \frac{d}{dt} \sin(1000t) = 1000\cos(1000t) \]  

(8)

Filter N (2 to 10) is used to limit the high frequency component, as shown in (9)

\[ C(s) = K_p + \frac{K_i}{s} + \frac{N K_d}{1+N/s} \]  

(9)

2.4. Command Weighting

The PID control structure is made flexible by differentiating error feedback between proportional and differential terms, while keeping the integral term the same (pure difference error feedback).

\[ u(t) = K_p e_p(t) + K_i \int_0^t e(r) \, dr + K_d \frac{de_d(t)}{dt} \]  

(10)

\[ e_p(t) = b r(t) - y(t) \]  

(11)

\[ e_d(t) = c r(t) - y(t) \]  

(12)

In practice, \( r(t) \) is piecewise signal that triggers high derivative, therefore \( c \) is set to 0, so that \( e_d(t) \) is only affected by the negative of DC motor plant output.

3. Full Feature Advanced PID Simulation

Having discuss all of the features of Advanced PID model, we can now combine them all in one model as shown in Figure 9, for the anti-windup, we use a technique called back-calculation. Also note that Figure 9, contains two sets of PID model, the upper one is basic PID model with limit input, while the lower one is advanced PID model. To ease the configuration setting, a GUI is developed [6], as shown in Figure 10.

![Figure 9. Advanced PID Model](image-url)

Figure 9. Advanced PID Model
Figure 10. PID GUI in Scilab

Figure 11 shows the importance of anti-windup in dealing with input limit in real device. The signal without anti-windup is not stable, while the advanced PID model with anti-windup is stable.

Figure 11. Output Signal with Different $K_i$
4. Conclusion

This paper illustrates how a Scilab is used as a teaching tool in control system, especially in DC Motor application. The Scilab software is by no doubt the best open-source tool in control system. By having a full feature PID model, called Advanced PID model, we get simulation results that can be compare to experimental data. This comparison will be done in Arduino to promote hands-on experience for undergraduate students in Computer Engineering Bina Nusantara University, Indonesia, who has a Control System Course.

Advanced PID Model consists of four (4) important components, i.e. autotuning relay-feedback, anti-windup using back calculation gain [5], command weightings, and last but not least, derivative term replacement. With these full features of PID model, students can further learn how to implement this model in real devices, by studying the relation between continuous time system and discrete time system.

References

[1] Hagglund T. 2012 Signal filtering in PID control *IFAC Proceedings Volumes*, vol. 45, no. 3, pp. 1-10.
[2] Mattavelli P., Tubiana L. and Zigliotto M. 2005 Simple control autotuning for PMSM drives based on feedback relay *European Conference on Power Electronics and Applications*, IEEE, , pp. 10
[3] Astrom K. J., Hagglund T. and Astrom K. J. 2006 Advanced PID control *ISA-The Instrumentation, Systems, and Automation Society Research Triangle*.
[4] Putra Y. H. Tuning Methods of PID Controller [Online]. Available: https://elib.unikom.ac.id/files/disk1/389/jbptunikomppgdl-yeffryhand-19448-13-bab13.pdf. [Accessed 31 08 2019].
[5] Kothare M. V., Campo P. J., Morars M. and Nett C. N. 1994 A unified framework for the study of anti-windup designs *Automatica*, vol. 30, no. 12, pp. 1869--1883.
[6] Toochinda D., June 2011 Digital PID Controllers. [Online]. Available: https://openlab.citytech.cuny.edu/?get_group_doc=4784/1393258757-DigitalPIDControllers_2011.pdf. [Accessed 31 08 2019].
[7] Ziegler-Nichols’ Open-Loop Method 17 July 2010. [Online]. Available: http://techteach.no/publications/articles/zn_open_loop_method/zn_open_loop_method.pdf. [Accessed 31 8 2019].