

# DC Motor Speed Control Based on System Identification and PID Auto Tuning

Wei-Jie Tang<sup>1</sup>, Zhen-Tao Liu<sup>2,3</sup>, Qian Wang<sup>1</sup>

1. School of Industrial Automation, Beijing Institute of Technology, Zhuhai, Guangdong 519088, China  
E-mail: [tangwei6474@163.com](mailto:tangwei6474@163.com)

2. School of Automation, China University of Geosciences, Wuhan, Hubei 430074, China

3. Hubei key Laboratory of Advanced control and Intelligent Automation for Complex Systems  
E-mail: [liuzhentao@cug.edu.cn](mailto:liuzhentao@cug.edu.cn)

**Abstract:** Speed control mode of a DC motor without knowing the specific parameters of the motor is discussed. The approximate mathematical model of the control system is obtained by the system identification when the output of the system is measured by loading the specific input signal. The PID control algorithm is adopted and the P, I, and D parameters are obtained by auto tuning. Hardware in the loop (HIL) experiments are carried out on MATLAB and Arduino platform, in which the experimental results demonstrate the feasibility of the proposal.

**Key Words:** DC motor, System Identification, PID control, HIL Experiment, Automatic Code Generation

## 1 Introduction

To analyze the performance of a control system, it needs to build the mathematical model of the system through various kinds of theorems [1], which is not so easy and applicable. In view of this, it becomes important to identify the unknown system and obtain the approximate mathematical model of the system by experimental methods. If the mathematical model of the control system can be easily obtained and the control algorithm automatically generates the code, the control system design could be greatly accelerated [2-3].

To solve the above problems, a fast and efficient way to design control systems is proposed. In order to get the approximate mathematical model of the system, it is required to load a certain input signal, and to detect the output of the system. On the basis of this, the speed closed-loop simulation control system model is built, and the PID controller is added to the system. The PID parameters are obtained by the self-tuning function of PID controller. Then, the tuned PID controller is added to the hardware in the loop (HIL) control system, and the response performance of the system is analyzed. Finally, the embedded C code is generated automatically, and HIL experiments are carried out. The experimental results show that the proposed method can shorten the design process control system.

The rest of paper is organized as follow. The system structure of DC Motor Speed Control is introduced in Section 2. System identification and analysis are presented in Section 3. And the HIL experiments are given in Section 4.

## 2 System Structure

The control system consists of a DC motor with an encoder and a Arduino mega 2560 controller as well as a DC motor drive and the power supply, as shown in Fig. 1. Arduino mega 2560 detects the motor encoder signal, calculates the motor speed, and the error compared to the desired speed. The duty cycle of the PWM signal is adjusted by the deviation to obtain the voltage signal of the control motor speed. The speed of DC motor is calculated in Eq. (1).

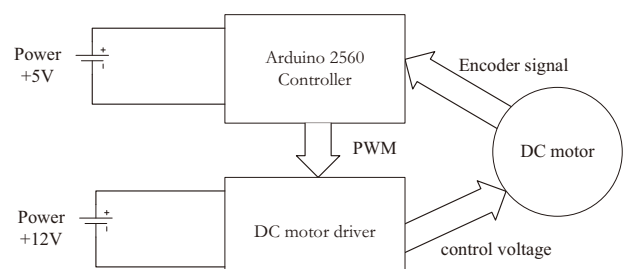


Fig. 1: System Structure

As seen in Eq. (1), it is easy to adjust the speed of the DC motor through changing the voltage.

$$n = \frac{U - IR}{C_e \phi} \quad (1)$$

## 3 System Identification and Analysis

To control the speed of DC motor, it needs to construct the speed closed loop control, at the same time, to join the PID controller. PID controller involves P, I, and D parameters' setting. The PID Simulink module provides a method of automatic parameter tuning as long as the tune button is

clicked. Such steps can be carried out. Firstly, the output is measured when the system loaded a given input signal. To establish the system model as shown in Fig.2, speed command module generates a step signal with a range of 0-255. Arduino mega 2560 digital pin 34 collect encoder signal and then calculate the motor speed. The encoder is the AB phase incremental encoder, each circle outputs 390 pulses. By detecting the time between adjacent pulses, we can calculate the motor speed according to Eq. (2).

$$v = \frac{360}{t} \times 60 = \frac{60}{390} \times \frac{1}{t} (r/min) \quad (2)$$

Where  $t$  is the time between adjacent pulses,  $v$  is the speed of the motor. The Simulink model is shown in Fig. 3.

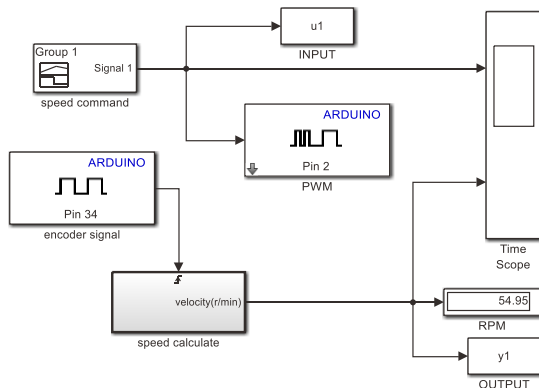


Fig. 2: Simulink mode of velocity measurement

Note that the sample time of the pulse signal detection module can't be too large, because it will lose pulse. And it could not be too small, because it will cause the Arduino overload. According to the Shannon sampling theorem, if the signal frequency is greater than or equal to two times by the sampling frequency of the signal, it can be well the sampling signal repetition. Thus, we need to calculate the frequency of the sampled signal. The motor rated speed is  $366 \pm 10$  r/min, we take 366 r/min for calculation, and we know the number of output pulses per turn is 390, thus, we can calculate the frequency of the sampled signal is 2379 Hz, then the sampling frequency can be set to 5000 Hz.

The input and output signals are exported to MATLAB work area, and we set  $u_1$  for input variable,  $y_1$  for output variable. Next, the control system by system identification is identified as shown in Fig. 4. Input and output signals are shown in Fig. 5. A variety of model, such as transfer function model, state equation model, nonlinear model could be used to estimate the system model [4-5]. In this paper, we use the transfer function model and default parameter.

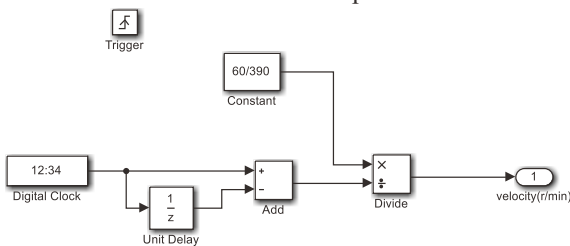


Fig. 3: Speed calculate

The identification process is shown in Fig. 6, where the model accuracy reaches 68.71%. Result of the identification

is shown in Fig. 7, which shows that the results of identification are good. The transfer function of the identified system is

$$G(z) = \frac{0.584841914225517}{z^2 - 0.682943586702267z + 0.069821011375924} \quad (3)$$

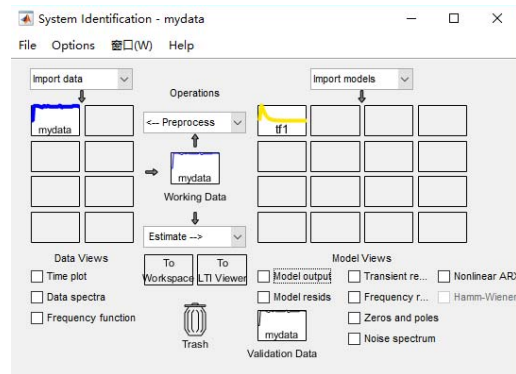


Fig. 4: System identification

Model in Simulink is established as shown in Fig. 8, which is an open-loop control system. The input and output waveform are shown in Fig. 9. From the waveform, it can be found that there is always a certain error between the output and the input. If precise control is to be achieved, a closed-loop control system must be constructed.

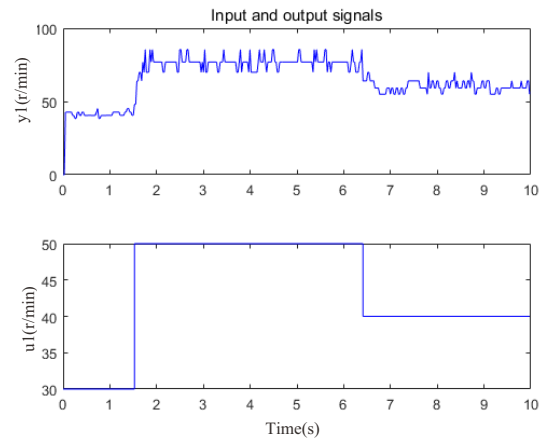


Fig. 5: Input and output signals

The simplest control method of closed loop control is PID control. There are three parameters  $P$ ,  $I$ , and  $D$  that need to be set.  $P$  accounts for present values of the error, if the error is large and positive, the control output will also be large and positive.  $I$  accounts for past values of the error, if the current output is not sufficiently strong, the integral of the error will accumulate over time, and the controller will respond by applying a stronger action.  $D$  accounts for possible future trends of the error, based on its current rate of change. Some applications may require using only one or two terms to provide the appropriate system control. This is achieved by setting the other parameters to zero. A PID controller is called as PI, PD, P or I controller in the absence of the respective control actions. PI controllers are fairly common, since derivative action is sensitive to measurement noise, whereas the absence of an integral term may prevent the system from reaching its target value.

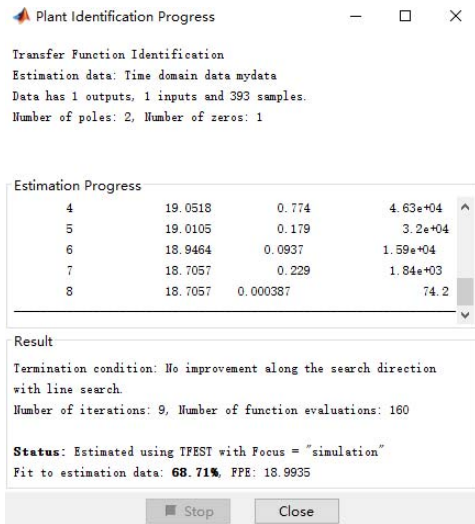


Fig. 6: Identification process

MATLAB provides a convenient method for tuning the parameters of the PID controller. MathWorks algorithm for tuning PID controllers meets the stability, dynamic performance and robustness of the control system by tuning the PID gains to achieve a good balance between performance and robustness. By default, the algorithm chooses a crossover frequency based on the plant dynamics, and designs for a target phase margin of 60°. When you interactively change the response time, bandwidth, transient response, or phase margin using the PID Tuner interface, the algorithm computes new PID gains.

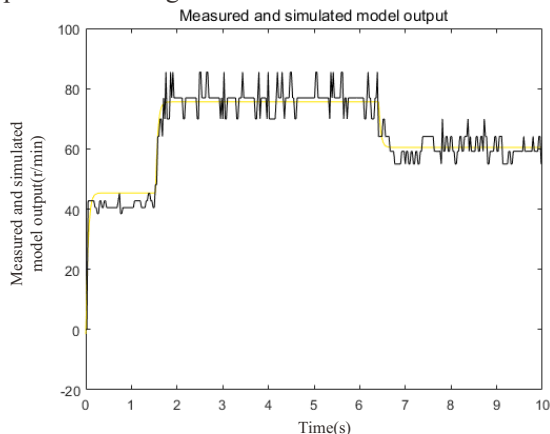


Fig. 7: Result of the identification

The closed loop control system based on PID in Simulink is established as shown in Fig. 10. Behind the PID controller it is a maximum 0-255 amplitude limiter. Just click the tune button, the PID parameters have been carried out automatic tuning, the designer can also through the panel on the response time and transient behavior to adjust, if the goal is reached, then click on the update block which will update the PID controller parameter tuning module. The PID parameters obtained after auto tuning are  $K_p=0.730$ ,  $K_i=1.247$ ,  $K_d=0.008$ .

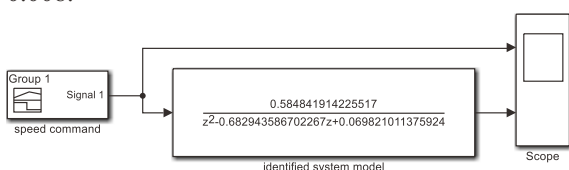


Fig. 8: System test after identification

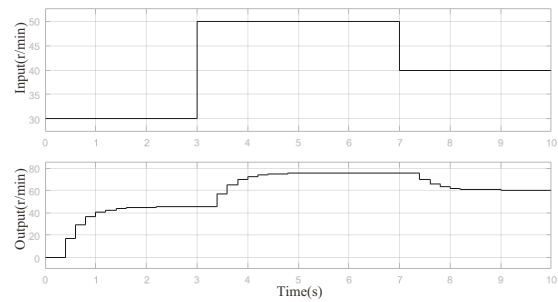


Fig. 9: Result of the system test

The input and output waveform through tuned PID control system is shown in Fig. 11.

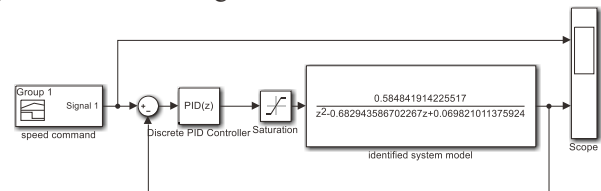


Fig. 10: PID tuning on identification system

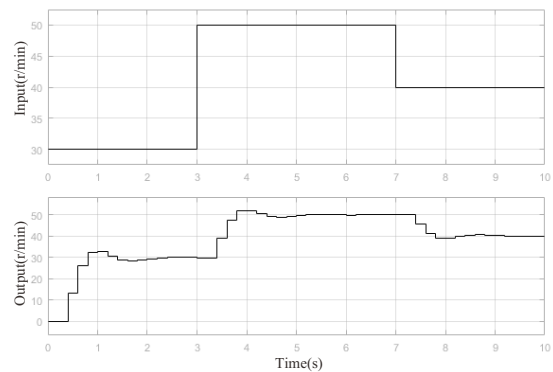


Fig. 11: Result of the tuned PID control system

From the output waveform in Fig. 11, although the output lags behind the input, the output can follow the input well. The results show that the closed-loop control system with PID controller is better than the open-loop system without PID controller.

## 4 HIL Experiment

After the above work, the identification system model is obtained, and the PID controller is designed. The PID control system is established based on Arduino mega 2560 in Simulink as shown in Fig. 12.

Computer and Arduino mega 2560 board are connected by USB cable. Serial communication port is set to make sure the connection is fine. The model in external mode is performed as shown in Fig. 13. We can observe real-time input and output waveform of the system by open scope.

In the HIL experiment, the random PID parameters are used to carry out firstly. PID parameters are set as  $K_p=5$ ,  $K_i=5$ ,  $K_d=0$ . In the experiment, when the speed is stable at 50 r/min, suddenly the speed command increased to 150 r/min, stable operation for a period of time, and then the speed requirement is suddenly reduced to 50 r/min. The real-time speed signal and error signal of the system are shown in Fig. 14. The PID parameters are set to the value obtained by auto tuning in the

third part of the article. The experiment is done again. The result of the HIL experiment is shown in Fig. 15.

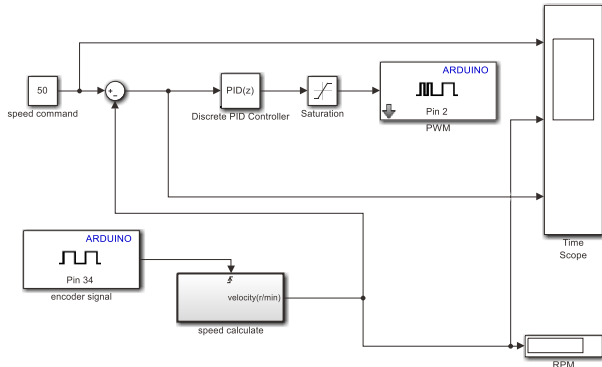


Fig. 12: PID control system based on Arduino

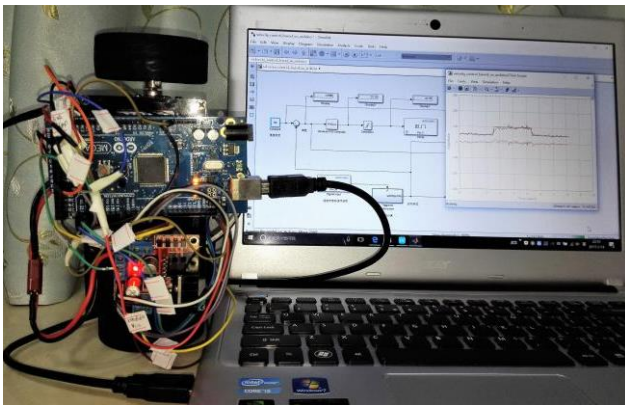


Fig. 13: HIL experiment

It can be seen from the two groups of experimental results that the motor actual speed can follow the desired speed signal. At low speed, the precision is high, but the accuracy decreases with the increase of speed.

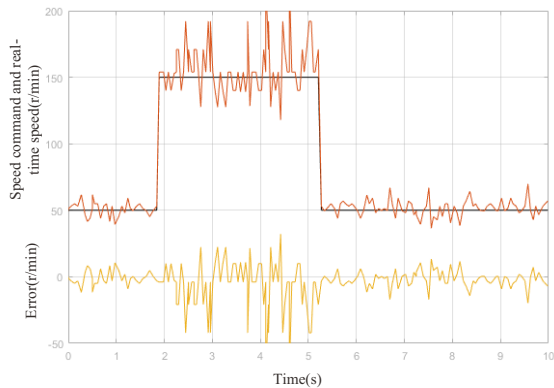


Fig. 14: Result of the HIL experiment ( $K_p=5, K_i=5, K_d=0$ )

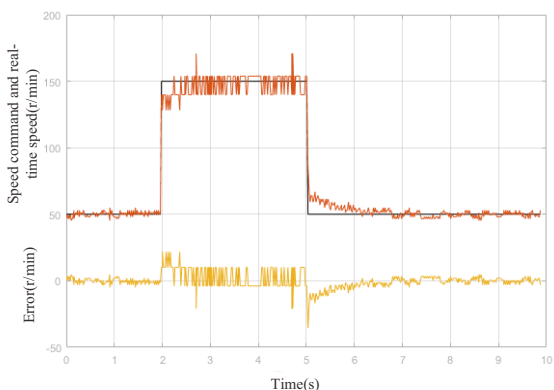


Fig. 15: Result of the HIL experiment(Auto tuning PID parameters)

By comparative analysis, the dynamic response of the system shown in Fig. 14 is faster than the system shown in Fig. 15. However, the system is more oscillation and low precision. The experimental results show that the performance of the system is better after PID parameters auto tuning.

## 5 Conclusion

In many cases, it is not easy to get the exact mathematical model of a control system. Through the experiments, we found that it is not necessary to know the exact mathematical model of the DC motor, but also can control it better. The system identification enables you to identify model of dynamic system from measured input-output data. PID control is a very practical and effective control method in the control system. The tuning of the PID parameters is also easy to get.

This paper studies the speed control of DC motor, using system identification to identify the mathematical model of the control system. Simulink model of PID closed-loop control is established for identification system, and the PID parameters are auto tuned. Finally, experiments are carried out on the Arduino Mega 2560 board. By comparison, it is found that the method used in this paper is very convenient and effective for the design of control system. The control system has good dynamic performance and precision. Through the experiments, we also found that with the increase of speed, the accuracy will become worse. Considering the collected encoder signal is not smooth enough. Low-pass filter can be considered for processing in the future research.

## References

- [1] Othman K A, Kamal M M, Mamat N, et al. System identification of discrete model for DC motor positioning[M]. 2009.
- [2] Munadi, Akbar M A. Simulation of fuzzy logic control for DC servo motor using Arduino based on MATLAB/Simulink[C]. IEEE International Conference on Intelligent Autonomous Agents, Networks and Systems. IEEE, 2014:42-46.
- [3] Wang X, Yin T, Chen J, et al. Rapid design of DC motor speed control system based on MATLAB[J]. Applied Mechanics & Materials, 2015, 743:168-171.
- [4] Mathworks, MATLAB users guide, [Online], Available: <http://www.mathworks.com>, The Mathworks, 2016.
- [5] MathWorks, "System Identification Toolbox," User's Guide, Version 9.4, 2016.