Design of DC motor controller based on MBD

Xu Hu¹, Peng Lu¹,²*, Yan Yang¹, Chunpeng Pan¹, Guoguo Wu¹ and Yan Ping³
¹School of Intelligent Manufacturing Engineering, Chongqing University of Arts and Science, Chongqing, 402160, China
²School of Information Engineering, Southwest University of Science and Technology, Sichuan, 621010, China
³School of manufacturing Science and Engineering, Southwest University of Science and Technology, Sichuan, 621010, China
*Corresponding author’s e-mail: 20140004@cqwu.edu.cn

Abstract. This paper designs a direct current (DC) motor controller based on Arduino and Simulink to solve the problems of poor openness, complicated programming and difficult signal detection of the existing teaching experimental equipment. The controller integrates the Model-Based Design (MBD) concept, involving single-chip microcomputer, system modeling, micro-motor, automatic control and other technologies. The hardware system includes Arduino, encoder, driver, input/output and other interfaces. The software system uses Simulink as the software development platform of the experimental device. It has the functions of algorithm design, simulation, execution, real-time data observation, etc. The limiting incremental PID control algorithm is used to control the speed of the DC motor, and the experimental results verify the design requirements. The hardware and software systems have the characteristics of good openness, fast implementation, safe and reliable, low cost, can complete the verification and improvement of various algorithms such as motor control, and can meet the teaching and research needs of the training of new engineering and technology.

1. Introduction

With the development of the high-end manufacturing industry, motor speed control systems based on digital control have been widely used in many fields. Mastering the principles and applications of motor control systems has become an urgent need for relevant technical personnel in intelligent manufacturing, robotics, unmanned driving, aerospace and other industries.

Over the past decade, the V-shaped development process based on model design (MBD) has been widely recognized in advanced manufacturing industries such as Huawei and Boeing. This design idea is to establish a corresponding system model in different stages of system development and use high-level language to design, simulation and system implementation. As shown in Figure 1, this method connects theory and practice closely. Developers can perform hardware-in-the-loop (HIL) simulation in all aspects, to reduce the difficulty of development, and greater flexibility [1].

The real-time simulation systems developed by dSPACE and the Quanser system have widely used MBD real-time HIL development platforms at present. However, the above equipment is relatively expensive and is mainly used for scientific research. It is still difficult to be used as experimental equipment for ordinary colleges. This paper introduces an open-source and low-cost DC motor control teaching experimental device based on the Matlab/Simulink software platform.
2. System hardware design

2.1. Introduction of the Main Controller

The hardware platform of the main controller uses the Arduino circuit board based on the ATmega 2560 single-chip microcontroller unit (MCU), which has the advantages of high performance, high integration, good reliability, and low cost. To realize the connection with the controlled object such as the motor, the experimental device has interfaces such as encoder and general I/O. It also provides peripherals such as buttons, LED, and RS-232 communication interface, which is convenient for human-computer interaction and data collection.

2.2. DC geared motor

DC geared motors are widely used in balance cars, inverted pendulums, and smart cars. This article chooses a micro DC geared motor with an encoder. The specific models are shown in Table 1.

| Parameter           | Value   | Parameter           | Value   |
|---------------------|---------|---------------------|---------|
| Brand               | SZCMMOTOR | Model               | JGA25-370 |
| No-load speed (shaft) | 1360 rad/min | Voltage             | 6 V     |
| Rated current       | 0.45 A   | Locked-rotor current | 1.3 A   |
| Power               | 1 W      | Weight              | 100 g   |

As shown in Figure 3, the DC motor is composed of brushes, commutators, armature windings, and main magnetic poles. By supplying direct current to the outside of the motor, the commutator switches according to the position of the armature. According to Faraday's law of electromagnetics, the charged winding will be rotated by the Lorentz force in the main magnetic pole. According to the knowledge of electrical machinery, the speed of the motor is shown in Formula (1). It can be seen that there are three ways to change the speed: changing the magnetic flux, armature voltage or armature loop resistance. The most common electronic speed controller is to change the armature voltage to change the speed. This article also uses the method of controlling the armature voltage to achieve speed regulation. Specifically, the Arduino 2560 outputs PWM waveforms with different duty cycles to the drive according to the control amount. The circuit board TB6612FNG, the core of the drive circuit board is an H-type full-bridge circuit, which changes the size and direction of the effective value of the output voltage by controlling the on-time of the power tube of the bridge arm to achieve the purpose of controlling the speed and direction of the DC motor.

\[ n = \frac{(U - i \cdot R_s)}{C_i \phi} \]  

(1)
For a decelerated DC motor, the output shaft of the motor is connected to the reducer, and the Hall encoder is directly installed at the tail of the motor. As shown in Figure 4, the colors of the positive and negative power cords are red and white, respectively. The other four colors are the power and signal wires of the encoder respectively. The black is the negative pole of the encoder power supply; the blue is the positive pole of the encoder power supply. The yellow wire and the green wire are the motor feedback signal wires (1 revolution of the motor can generate 11 pulse signals), which are used to measure the speed of the controller. The reduction ratio of the reducer is 4.4.

2.3. Drive circuit
In this work, a kind of mature driver chip, TB6612FNG, is chosen to realize driver for motor, which with small volume and widely used in the current electronic design.

As shown in Figure 5, the ports of TB6612 are divided into two groups (motor A and motor B). Taking motor A port as an example, the motor speed control PWM signal output by the single-chip microcomputer is connected to the PWMA port; STBY is the enable of the entire drive chip Signal, when a low level is fed into this pin, all the motors are driven by which will stop; the direction control signal terminals are AIN1 and AIN2.
3. System algorithm and software implementation

3.1. Principle of Limiting Incremental PID

PID control, namely proportional-integral-derivative control, is a control method defined in the principle of automatic control and plays a pivotal role in modern industrial control. PID control is mainly to combine the three operations of proportional, integral, and derivative of the system output deviation to form the output control quantity to reduce the system error and improve the system response speed and response effect [2-5].

PID can be divided into analog PID and digital PID. In digital PID, the calculation formula of its control quantity is:

\[
\Delta u(n) = \Delta u(n-1) = K_p \Delta e(n) + K_i \sum_{i=0}^{n} e(i) + K_d [e(n) - e(n-1)]
\]

where \( u(n) \) is the output of the controller; \( e(n) \) is the input of the controller, which is the difference between the expected value of the system and the actual measured value. As shown in Figure 7, digital PID uses instead of continuous PID.

Since Formula (2) will accumulate all the error values when it is implemented, the amount of calculation is relatively large, so an incremental algorithm appears. Incremental PID control is a basic form of digital PID control algorithm. It is a control algorithm that performs PID control on the increment of the control quantity (the difference between the current control quantity and the previous control quantity). The output of the control system is the change in the control quantity. Since the control quantity will be very large at the beginning, generally, it will exceed the maximum value of the system memory. The control quantity used in this system is 8 bits, so the final limit is 255.

\[
\Delta u(n) = \Delta u(n-1) - \Delta u(n-2) = K_p \Delta e(n) + K_i e(n) + K_d [e(n-2) - 2e(n-1) + e(n-2)]
\]

3.2. Algorithm software implementation

The software development environment of this article is as follows: Win10 operating system, Matlab2015a, and the hardware support package of Arduino matching Simulink.

As shown in Figure 8, this article uses the soEncoder module in Simulink to count motor encoder pulses, and this module uses quadruple frequency technology to count encoder pulses. The module outputs 0 from power-on, that is, the current position of the motor is the initial point, and the pulses of the encoder are counted when the motor is rotating. The y value is a 32-bit signed integer, and positive and negative represent different rotation directions. The method to obtain the speed is to use the delay link to calculate the difference between the data output by the soEncoder module at the two moments, and then divide the difference by the time interval between the two moments. Due to the interference of the motor encoder itself and the instability of the motor itself, the speed data obtained will have large noise fluctuations, and direct use will lead to inaccurate information detection, so filtering processing is also required. The data type conversion module in Figure 8 uses a first-order digital low-pass filter to filter the speed signal.

\[
Y_n = aX_n + (1-a)Y_{n-1}
\]

where \( X_n \) is the current sampled value, \( Y_{n-1} \) is the previous filtered output value, a is the filter coefficient, which is much less than 1, is the current filtered output value.
Figure 9 is the main program diagram of DC motor speed control Simulink, where the motor speed and filter module are described in Figure 9, and then a speed signal is obtained. The constant 15 is the set speed of the motor spindle, and the output of the filter below is the actual speed of the motor. Because after the reducer, the power output of the motor reaches the spindle, the speed of the spindle needs to be divided by the measured speed of the motor encoder A reduction ratio. Therefore, the encoder output at the bottom in Figure 9 is multiplied by the constant term $K$. The set value and the actual value are subtracted to obtain the error, which is the input of the motor speed PID controller, and the system output is the PWM control value of the motor driver. The sign and absolute value of the motor drive determine the direction and speed of the motor.

The function of the switch module in the upper part of Figure 9 is to directly send “1” high level to the motor module when the input speed value is not equal to 0, as the enable signal of the motor driver; when the input speed value is 0, it will send “0” low level to the motor module, and the motor stops rotating. Refer to Figure 10 for details of the motor module. In1 is the enable signal input of the motor driver, and In2 is the PWM control input, which is taken as an absolute value and limited in the program. This article uses the PWM output module in the Arduino hardware package of Simulink. If the value 255 is input to this module, the module will automatically output a PWM waveform with a duty cycle of 100%; and when the input value is 127, a PWM waveform with a duty cycle of 50% will be generated.

Figure 11 is a program diagram of the PID controller for the motor speed, which is a subroutine of the PID controller in Figure 9. According to the empirical debugging method, set the corresponding control parameters of PID to [5, 0.1, 2], the output of the controller is the PWM control value, and its sign will determine the positive and negative rotation of the motor, and the absolute value determines the motor speed, And finally connected to In2 in Figure 9.
4. **Analysis of experimental results**

After the software and hardware design in the previous section, this section will conduct a control test on the system. The system control sampling rate is 50 ms, and the program download mode is in external operation mode, which is the HIL control mode.

![Figure 12. System speed control step response under external interference](image1)

As shown in Figure 12, a total of 700 steps were sampled. It can be seen that in less than 5 steps (250 ms), the motor output reaches the set value, and the system has no overshoot. At 500 steps, the motor spindle inputs a random load, and the system restores to the set value within 50 steps. The sub-picture in Figure 12 is a partially enlarged picture.

![Figure 13. The PWM control value output by the PID module](image2)

To further study the internal control principle of the system, this paper tests the output of the PID module. As shown in Figure 13, it can be seen that the speed difference at the beginning of the system is the largest so that the PID module output value reaches 80, but it still does not reach the system threshold of 255, therefore, the set speed of the system can still be increased. The partially enlarged diagram shows that when the system speed is interfered with by an external load, the speed will decrease, and the output of the PID controller will increase immediately, it realizes the real-time control of the speed of the DC motor.

5. **Conclusion**

In this paper, a motor speed control platform based on low-cost hardware such as Simulink and Arduino is constructed, and the realization process of software programs such as encoder speed acquisition, PID control module, and motor control module is analyzed in detail. It has been verified that the system has
strong tracking ability, small overshoot, good anti-interference ability, and the control effect meets the requirements.

The biggest feature of this experimental platform is low cost, good openness, and visualization of control process information. Based on this platform, the hardware can continue to add modules such as gyroscope, Bluetooth, and Internet of Things (IoT) communication; and the algorithm level can verify many control algorithms such as multi-loop PID, active disturbance rejection control, fuzzy control, and deep reinforcement learning. Using this experimental platform will help researchers to realize complex control algorithms quickly and improve teaching efficiency; furthermore, the experimental platform adopts the MBD design concept, which helps to develop the ability of students to explore an innovative spirit.

Acknowledgments
This work was supported by the Innovation and Application Development Program of Chongqing (Grant No. cstc2020jscx-msxmX013, cstc2019jscx-fxydX0040), the Science and Technology Research Program of Chongqing Education Commission of China (Grant No.KJQN202101333), and the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJQN201801342), the Foundation for High-level Talents of Chongqing University of Arts and Sciences (Grant No. R2018JD05).

References
[1] Croce Sipontina, Neu Julian, Hubertus Jonas. (2021) Model-Based Design Optimization of Soft Polymeric Domes Used as Nonlinear Biasing Systems for Dielectric Elastomer Actuators. Actuators, 10(9): 87–94.
[2] Veinović Slavko, Stojić Djordje. (2021) Optimized four-parameter PID controller for AVR systems with respect to robustness. International Journal of Electrical Power and Energy Systems, 135: 40–46.
[3] Wang Qianru, Wang Caixia. (2021) Research on fuzzy PID droop Control method for DC microgrid in island mode. Journal of Physics: Conference Series, 2010(1): 124–130.
[4] Zhang Mingli, Zhang Yijie, He Xiaolonge. (2021) Adaptive PID Control and Its Application Based on a Double-Layer BP Neural Network. Processes, 9(8): 78–83.
[5] Demirtas Metin, Calgan Haris, Amieur Toufik. (2021) Small-signal modeling and robust multi-loop PID and formula omitted controllers synthesis for a self-excited induction generator. ISA Transactions, 117: 327–332.