

# Research on closed-loop simulation system for digital AC servo system

HAO ShuangHui, ZHENG WeiFeng<sup>†</sup>, HAO MingHui, MA RuQi & LI Hong

Department of Mechatronics Engineering, Harbin Institute of Technology, Harbin 150001, China

**An intelligent and efficient closed-loop simulation system without any hardware is proposed for AC servo system. According to the characteristics of AC servo system and its control processor, the code executing platform combined with the reverse Polish notation algorithm is developed. To gain the accurate simulation results, the discrete motor model and IPM model are built to deal with the dead-time. Based on the simulation system, parameter identification can be implemented and control parameters can be optimized by using the exhaust algorithm. The parameters obtained by the simulation system were successfully applied to an experimental system, and the favorable control performance was achieved.**

control code, closed-loop simulation, discrete model, parameter identification, parameter optimization

## 1 Introduction

In digital AC servo systems, PID controllers are usually used to achieve the closed-loop control. The control performance of the system is determined by the control code and parameters<sup>[1]</sup>. With the help of MATALAB, researchers can get the best control models and results, but it is hard to verify the control code and obtain accurate parameters. On the other hand, the integrated development environment of the processors can be used to execute the code, but it can not realize the closed-loop simulation<sup>[2-5]</sup>. To improve the development of the control code, a closed-loop simulation system is proposed.

Methods for parameter identification and optimization have been proposed, but they are too complicated to implement. Based on the closed-loop simulation system, parameter identification and optimization can be achieved. Furthermore, the simulation system will greatly reduce the cost of development.

Figure 1 shows the digital AC servo system, which consists of motor, processor, IPM (intelligent power model) and encoder. As the core of the system, the processor accepts the feedbacks and generates 6-phase PWM (pulse width modulation) waveform with the dead-time

dead-time for the IPM to drive the motor<sup>[6]</sup>.

To develop the closed-loop simulation system, the code executing platform, motor model and IPM model should be built. Considering the dead-time effect on the real system, the discrete motor model and IPM model are designed.

The simulation system including IPM model, PMSM model, encoder model and sensor model (see Figure 1) executes the code to serve as the processor, which can carry out the closed-loop simulation for the whole system.

## 2 Code executing platform

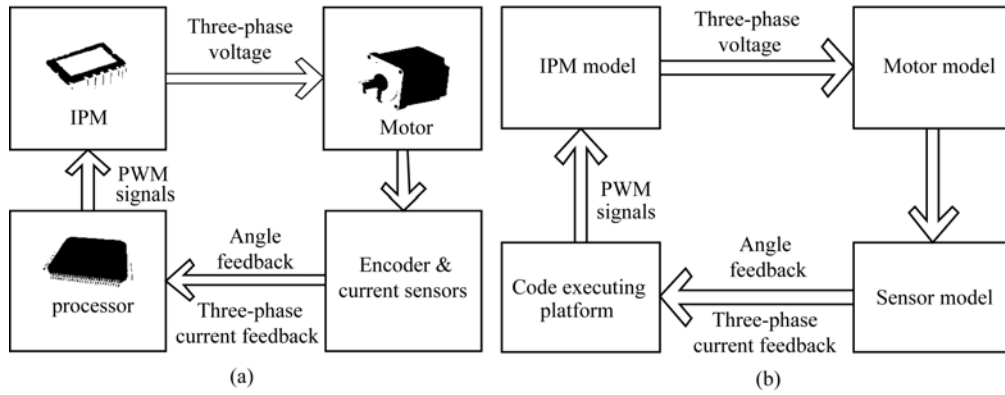
In a real system, the processor reads the registers (*ADA0CR0*, *ADA0CR1*, *ADA1CR0* and *CB0RX*) to receive the feedbacks (current and angle) and calculates the values of 3-phase voltages which will be saved in the registers (*TQ0CCR1*, *TQ0CCR2* and *TQ0CCR3*) to generate 6-phase PWM waveform with the dead-time.

In the simulation system, the control code is read into

Received March 3, 2009; accepted April 30, 2009

doi: 10.1007/s11431-009-0207-0

<sup>†</sup>Corresponding author (email: [hitren1982@163.com](mailto:hitren1982@163.com))



**Figure 1** The structure of AC servo system and the simulation system. (a) AC servo system; (b) the simulation system.

the code executing platform in the format of text and saved as character arrays. The closed-loop simulation system is developed by using the Visual C++.

### 2.1 Word analysis

Word analysis is to distinguish the tokens from the code, such as “if” and “+”. In the simulation system, a token is recorded by the class whose name is CSymbol. The class includes the attributes (name, type and value) of the token. Figure 2 shows the results of word analysis to deal with the sentence: “int i=1”. The platform scans the sentence from left to right and processes the tokens one by one.

### 2.2 Sentence analysis

Sentence analysis is to discriminate the sentences in the control code. The sentences will be classified as basic expressions, functions and special sentences. The sentence is made up of tokens. The platform should identify them and arrange the basic expressions together to be a function or a special sentence. All of them would be dealt with as the function module. The function module may include submodules. In the end, the platform gains the numbers of basic expressions together with the information of executing steps.

The results of sentence analysis are shown in Figure 3. By using the results, the platform can execute the basic expressions step by step.

The platform can not execute the basic expression such as “ $a = 2 \times 3 + 8 / 2$ ”. The basic expression should be

pretreated by the reverse Polish notation algorithm (RPN) in advance. The RPN is used to convert the prefix expression into the postfix expression. For example, the basic expression “ $(1+5) \times 6 / 3$ ” can be converted into the postfix expression “1, 5, +, 6, ×, 3, /”<sup>[7]</sup>. In the end, the RPN is used to execute the postfix expression from left to right and get the result of arithmetic.

## 3 Motor model

### 3.1 Motor model in the $d$ - $q$ coordinates

In the real system, the voltage of busbar capacitance is  $U_{PN}$ . The 3-phase voltages ( $U_a$ ,  $U_b$ , and  $U_c$ ) can be formulated as

$$U_a = \frac{U_{PN}}{2} - \frac{U_{PN} \times TQ0CCR1}{TQ0CCR0}, \quad (1)$$

$$U_b = \frac{U_{PN}}{2} - \frac{U_{PN} \times TQ0CCR2}{TQ0CCR0}, \quad (2)$$

$$U_c = \frac{U_{PN}}{2} - \frac{U_{PN} \times TQ0CCR3}{TQ0CCR0}, \quad (3)$$

where  $TQ0CCR0$  is the invariable in the real system,  $U_d$  and  $U_q$  are the voltages of  $d$ -axis and  $q$ -axis,  $\theta_e$  is the electric angle of motor<sup>[8,9]</sup>.  $U_d$  and  $U_q$  can be given by

$$U_d = \frac{2}{3} \left( U_a \cos \theta_e + U_b \cos \left( \theta_e - \frac{2}{3} \pi \right) + U_c \cos \left( \theta_e + \frac{2}{3} \pi \right) \right), \quad (4)$$

CSymbol Name: “int” Type: TOKEN_INT(41) Value:	CSymbol Name: “i” Type: TOKEN_ID (2) Value:	CSymbol Name: “=” Type: TOKEN_ASSIGN (17) Value:	CSymbol Name: “1” Type: TOKEN_NUM (3) Value: 3	CSymbol Name: “,” Type: TOKEN_SEMI (7) Value:
---	--	---	---	--

**Figure 2** The results of word analysis.

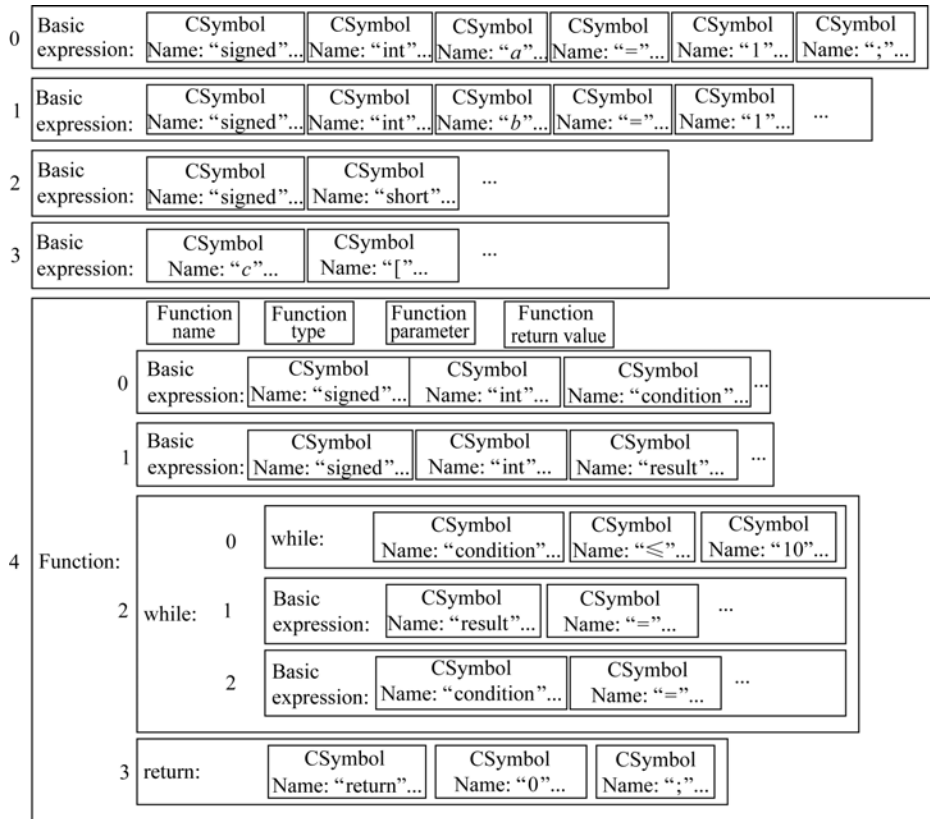


Figure 3 The results of sentence analysis.

$$U_q = -\frac{2}{3} \left( U_a \sin \theta_e + U_b \sin \left( \theta_e - \frac{2}{3} \pi \right) + U_c \sin \left( \theta_e + \frac{2}{3} \pi \right) \right). \quad (5)$$

According to the differential equations below, we can calculate the current and angle of motor.

$$\frac{d}{dt} i_d = \frac{1}{L_d} (U_d - R i_d + L_q p \omega_r i_q), \quad (6)$$

$$\frac{d}{dt} i_q = \frac{1}{L_q} (U_q - R i_q - L_d p \omega_r i_d - \lambda p \omega_r), \quad (7)$$

$$\frac{d}{dt} \omega_r = \frac{1}{J} (T_e - f \omega_r - T_m), \quad (8)$$

$$\frac{d\theta_r}{dt} = \omega_r, \quad (9)$$

$$T_e = 1.5 p [\lambda i_q + (L_d - L_q) i_d i_q], \quad (10)$$

$$\theta_e = p \theta_r. \quad (11)$$

In each control cycle, the 3-phase voltages can be calculated by the platform and the PWM waveform can be generated with the dead-time. By neglecting the dead-time effect, the 3-phase voltages can be calculated

according to eqs. (1), (2) and (3). By the algorithm of Rungekutta, the differential equations can be solved and the feedbacks (current and angle) can also be obtained. However, it is difficult to get the perfect simulation results because of the dead-time effect. Therefore, it needs the discrete models to deal with the dead-time for the simulation system.

### 3.2 Discrete motor model

We build the discrete models according to the characteristics of the digital AC servo system and its circuits. The computation cycle is the minimal resolving time of the processor. The discrete models are the motor model and IPM model.

3.2.1 Generation of 6-phase PWM waveform with the dead-time. There is a motor-control module in the processor and it can generate 6-phase PWM waveform with the dead-time. The control cycle is determined by the value of  $TQ0CCR0$ . The duty-cycles of the 3-phase PWM are determined by the values of  $TQ0CCR1$ ,  $TQ0CCR2$  and  $TQ0CCR3$ . The range of them is 0 to  $TQ0CCR0+1$ . In the processor, the value of the main timer and  $TQ0CCR1$  determine the output (high level or

low level) of PWM waveform in U phase. By the dead-time timer, the PWM waveform can be generated with the dead-time. The simulation system is designed to generate the PWM waveform with the dead-time as the processor. The process of generating 6-phase PWM waveform is shown in Figure 4.

In Figure 4, the  $CCR_0$  is  $TQ0CCR_0$ , and  $CCR_i$  ( $i = 1, 2, 3$ ) are  $TQ0CCR_1$ ,  $TQ0CCR_2$ , and  $TQ0CCR_3$ .

3.2.2 IPM model. The voltage between the P pole and the N pole of IPM is  $U_{PN}$ . In the U phase, when the upper arm is turned on, the voltage of U phase is  $U_p$ . When the lower arm is turned on, the voltage of U phase is  $U_N$ . The upper arm and lower arm are prohibited to be turned on at the same time, or else the IPM will be damaged. The dead-time is provided to protect them from short circuit. When both the upper arm and lower

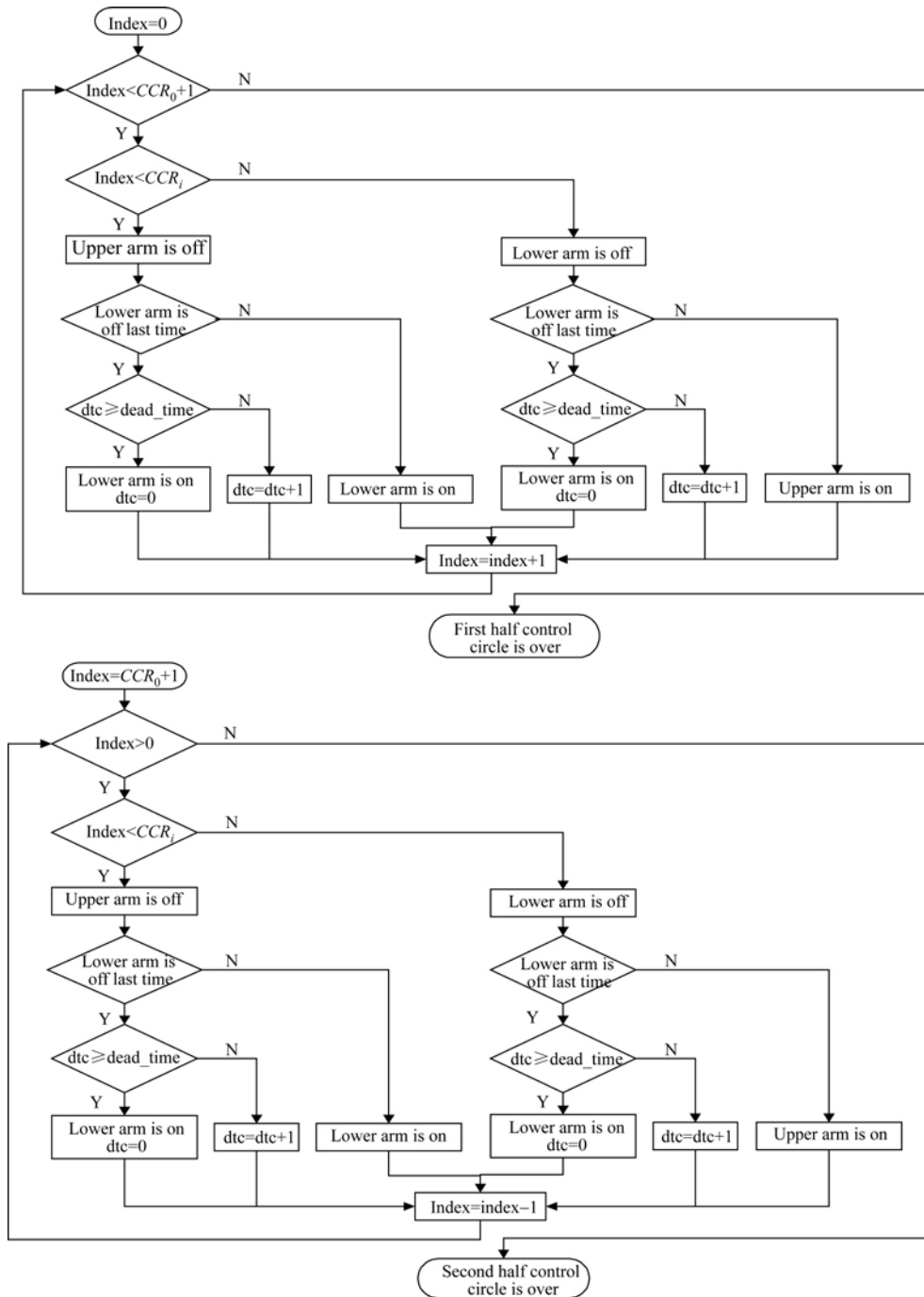


Figure 4 The process of generating 6-phase PWM waveform.

arm are turned off, the current of U phase would be maintained by the diode. If the current is from the N pole to the motor, the voltage of U phase is  $U_N$ . And if the current is from the motor to the P pole, the voltage of U phase is  $U_P$ . Figure 5 shows the model of U phase in the IPM.

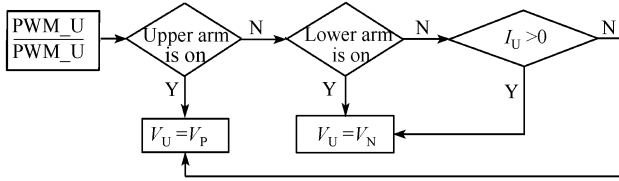


Figure 5 The model of U phase in the IPM.

The three phases of the motor are symmetrical, so the models of them are the same with each other. The simulation system can get the same outputs just like a real IPM. The outputs (PWM waveform, current and voltage) of U phase in the real system are shown in Figure 6.

The outputs (PWM waveform, current and voltage) of U phase in the simulation system are shown in Figure 7. From Figures 6 and 7, we can see that the dead-time of PWM waveform is achieved in the simulation system.

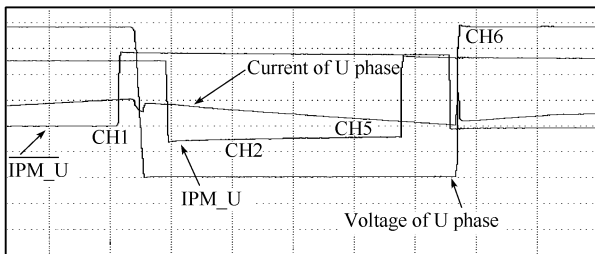


Figure 6 The PWM waveform and voltage of the real system.

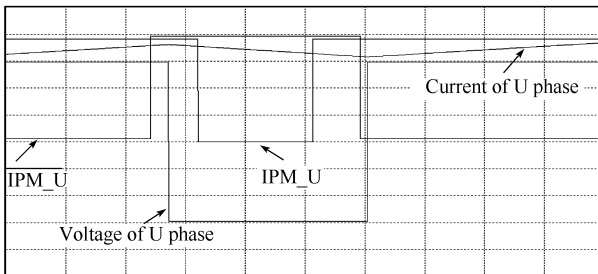


Figure 7 The PWM waveform and voltage of the simulation system.

3.2.3 Solution of discrete motor model. The 3-phase voltages ( $U_a$ ,  $U_b$ , and  $U_c$ ) can be provided by the IPM model. The feedbacks (current, velocity and angle) can be calculated in each computation cycle by the discrete motor model as follows.

$$\begin{cases} U_{d(k)} = \frac{2}{3} \left( U_{a(k)} \cos \theta_{e(k)} + U_{b(k)} \cos \left( \theta_{e(k)} - \frac{2}{3} \pi \right) \right. \\ \quad \left. + U_{c(k)} \cos \left( \theta_{e(k)} + \frac{2}{3} \pi \right) \right), \\ U_{q(k)} = \frac{2}{3} \left( U_{a(k)} \sin \theta_{e(k)} + U_{b(k)} \sin \left( \theta_{e(k)} - \frac{2}{3} \pi \right) \right. \\ \quad \left. + U_{c(k)} \sin \left( \theta_{e(k)} + \frac{2}{3} \pi \right) \right), \end{cases} \quad (12)$$

$$\begin{cases} \begin{bmatrix} i_{d(k+1)} \\ i_{q(k+1)} \end{bmatrix} = \begin{bmatrix} 1 - \frac{R_d T_s}{L_d} & \omega_{e(k)} \frac{L_q T_s}{L_d} \\ 1 - \omega_{e(k)} \frac{L_d T_s}{L_q} & -\frac{R_q T_s}{L_q} \end{bmatrix} \begin{bmatrix} i_{d(k)} \\ i_{q(k)} \end{bmatrix} \\ \quad + \begin{bmatrix} \frac{T_s}{L_d} & 0 \\ 0 & \frac{T_s}{L_q} \end{bmatrix} \begin{bmatrix} U_{d(k)} \\ U_{q(k)} - \omega_{e(k)} \lambda \end{bmatrix}, \end{cases} \quad (13)$$

$$\omega_{r(k+1)} = \omega_{r(k)} + \frac{T_s}{J} (T_{e(k)} - f \omega_{r(k)} - T_{m(k)}),$$

$$\theta_{r(k+1)} = \theta_{r(k)} + \omega_{r(k)} T_s,$$

$$T_{e(k+1)} = 1.5 p \lambda i_{q(k+1)},$$

$$\omega_{e(k+1)} = p \omega_{r(k+1)},$$

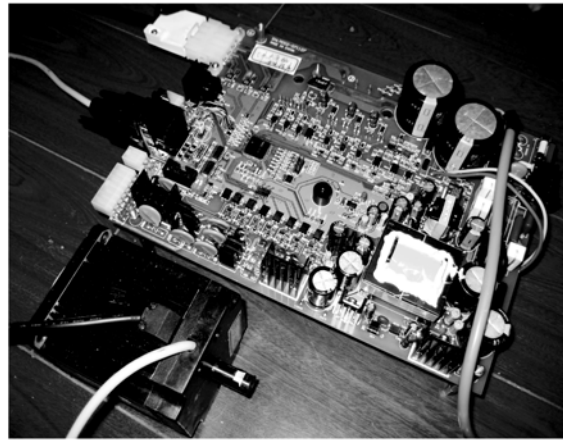
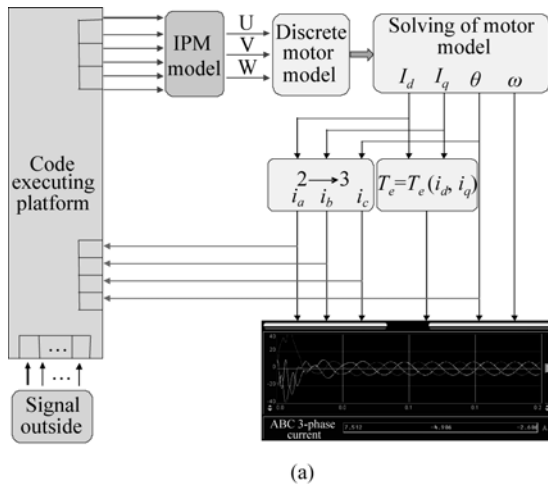
$$\theta_{e(k+1)} = p \theta_{r(k+1)},$$

$$\begin{cases} I_{a(k+1)} = I_{d(k+1)} \cos \theta_{e(k+1)} - I_{q(k+1)} \sin \theta_{e(k+1)}, \\ I_{b(k+1)} = I_{d(k+1)} \cos \left( \theta_{e(k+1)} - \frac{2}{3} \pi \right) \\ \quad - I_{q(k+1)} \sin \left( \theta_{e(k+1)} - \frac{2}{3} \pi \right), \\ I_{c(k+1)} = I_{d(k+1)} \cos \left( \theta_{e(k+1)} + \frac{2}{3} \pi \right) \\ \quad - I_{q(k+1)} \sin \left( \theta_{e(k+1)} + \frac{2}{3} \pi \right). \end{cases} \quad (14)$$

The simulation system has been built up successfully. The closed-loop simulation system and the real servo system are shown in Figure 8.

In the real system, the control code is required to achieve the step response of  $I_d$  ( $I_d = 1$  A). By executing the same control code, the result of the simulation system with the discrete models is shown in Figure 9.

From Figure 9, we can see that the results of real system and simulation system are very similar. This indicates that the simulation system with the discrete models



**Figure 8** Systems built in this paper. (a) The simulation system; (b) the real experimental system.



**Figure 9** The responses of  $I_d$ . (a) The response of  $I_d$  in the real system; (b) the response of  $I_d$  in the simulation system.

can obtain accurate simulation results.

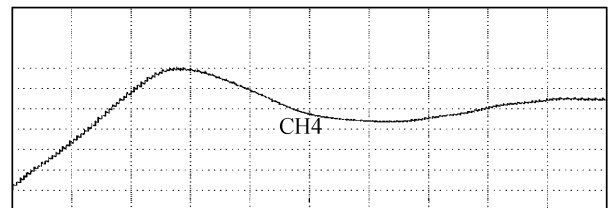
## 4 Parameter identification and optimization based on the simulation system

### 4.1 Parameter identification

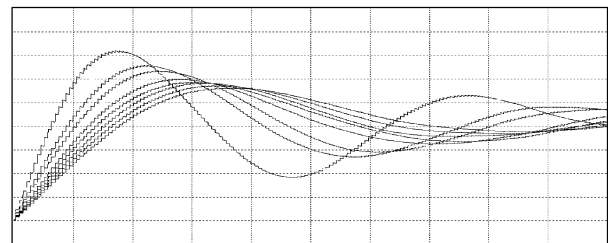
In the real system, to realize the vector control and PID control, inductances of  $d$ -axis and  $q$ -axis ( $L_d$  and  $L_q$ ) are necessary parameters. If the inductances can be obtained accurately, researchers would regulate the control parameters to meet the requirements of response and positioning<sup>[10,11]</sup>. Based on the simulation system, a novel method is designed to implement the parameter identification.

According to the method of vector control,  $I_a = I_d$  if both  $\theta_e$  and  $I_q$  are kept zero. The step response of  $I_d$  is

determined by the parameters ( $K_p$  and  $L_d$ ). If  $K_p$  is constant, the step response would be determined by  $L_d$ . In the simulation system, numbers of step responses of  $I_d$  can be obtained by use of different  $L_d$ . If  $K_p = 500$ , the step response of  $I_a$  in the real system is shown in Figure 10. In the simulation system,  $K_p$  is also set to be 500 and  $L_d$  is changed from 5 to 100 mH. The simulation system produces many curves of step response (Figure 11). From Figure 11, we can select a curve which is similar with the one in Figure 10. If its corresponding  $L_d$  is 25 mH in the simulation system, the value of  $L_d$  in the real system should be 25 mH. Similarly, we can achieve  $L_q$  by the parameter identification based on the simulation system.



**Figure 10** The step response in the real servo system.



**Figure 11** The step responses in the simulation system.

### 4.2 Parameter optimization

In the real system, to obtain the best parameters the en-

gineer has to regulate them time after time [12, 13]. Based on the simulation system, the exhaust algorithm is designed to implement the parameter optimization automatically.

The *IAE* ( $IAE = \int_0^T |e(t)| dt$ ) is usually used to evaluate the PID parameters. For the curve of step response, the hatching area shows the performance of the PID controller (see Figure 12). The simulation system can get  $n \times m$  groups of *IAE* values by the  $n \times m$  groups of  $(K_p, T_i)$  and select the least *IAE* value to get the best parameters.

Based on the exhaust algorithm, by calculating the *IAE* value and selecting the least one from the results, the simulation system can obtain the best parameters (see Figure 13).

For  $K_{p,0}=0, T_{i,0}=0, \Delta K_v = 0 \times 200, \Delta T_v = 0 \times 00200000, n = 64, \text{ and } m = 64$ , the simulation system would generate 4096 curves of step responses together with their *IAE* values (Figure 14).

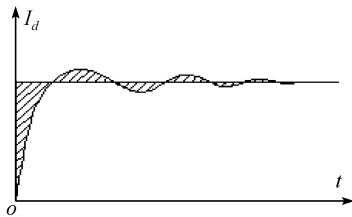


Figure 12 Integral of absolute magnitude of the error (*IAE*).

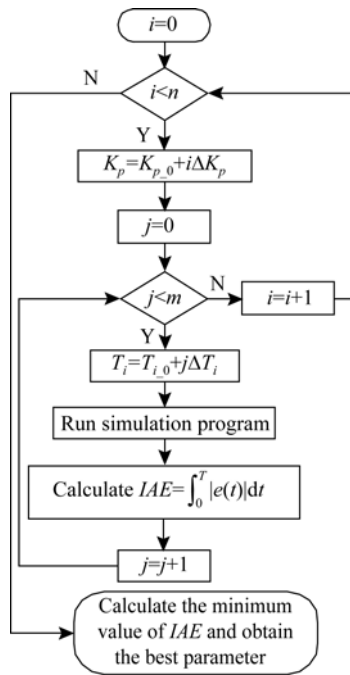


Figure 13 Parameter optimization by using the exhaust algorithm.

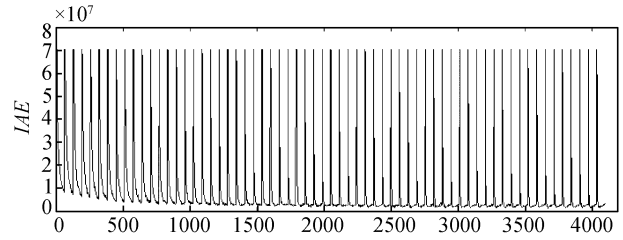


Figure 14 The *IAE* values achieved by the exhaust algorithm.

In Figure 14, the parameter group ( $K_p = 0 \times 2800, T_i = 0 \times 07200000$ ) is the best one and its corresponding *IAE* value is the least. The parameter group was applied to the simulation system and the real servo system. The results of step response are shown in Figure 15.

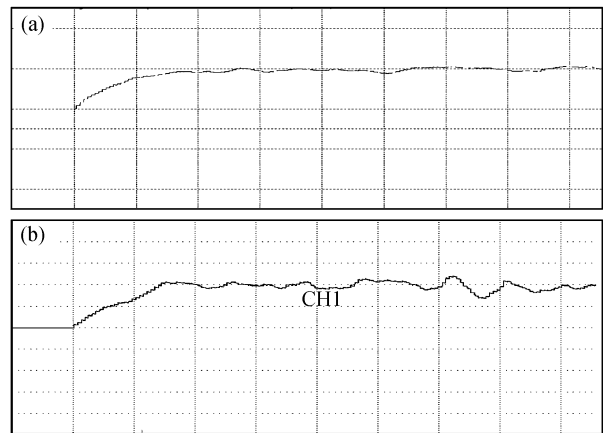
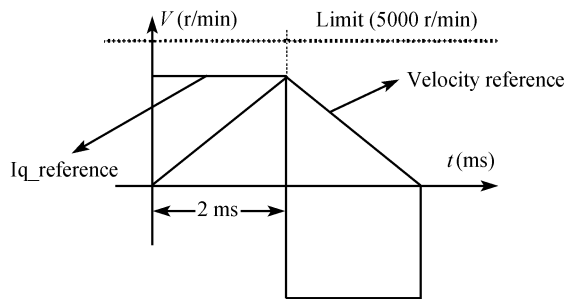


Figure 15 Step response results. (a) Step response result in the simulation system; (b) step response result in the real servo system.

From Figure 15, we can find that the results of the simulation system and the real system are very similar. The effect of the step response looks nice. It is shown that the parameter optimization based on the simulation system can be used to gain the best control parameters.

## 5 Application to high-speed/high-precision positioning system

For a high-speed/high-precision positioning system, the motor is required to complete the fixed angular displacement with a great acceleration. So the system should be excellent in response and positioning. Therefore, the control parameters play an important role in the system. Usually, if the parameters of motor are obtained in advance, the best control parameters can be gotten by regulating. For the system, the velocity reference and current reference of  $q$ -axis ( $I_q$  reference) are shown in Figure 16. The real current of  $q$ -axis is required to

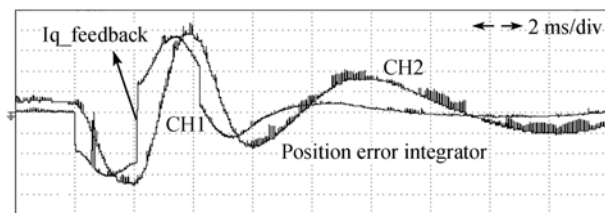


**Figure 16** The velocity and current references in the system.

respond as rapidly as the  $I_q$  reference (see Figure 16).

By using the simulation system, the parameters of the motor were obtained by parameter identification, and the best control parameters were gotten by parameter optimization. With the best control parameters, the system achieved the excellent experimental results (see Figure 17). Compared with the  $I_q$  reference in Figure 16, the  $I_q$  feedback (the feedback of  $I_q$ ) meets the requirement of response commendably.

For the system, if the error integrator of position (see



**Figure 17** The current of  $q$ -axis and the position error in the system.

Figure 17) is cleared as soon as the velocity reference (see Figure 16) is completed, then it is considered to achieve the performance of high-speed/high-precision positioning. From Figures 16 and 17, we can see that the real system needs 5 ms to achieve the positioning and the time of velocity reference is required to be 4 ms. In one minute, the system can carry out the high-speed/high-precision positioning 12000 times.

The experimental results show that the simulation system can not only be applied to the control code, but can also be used to seek for the best control parameters for AC servo systems.

## 6 Conclusions

(1) A closed-loop simulation system was proposed, which is a useful innovation in improving the development efficiency of digital AC servo systems.

(2) The discrete models have been built. The simulation system can get accurate results with the dead-time.

(3) Based on the simulation system, a new method was proposed to carry out the parameter identification. It can solve the difficult problem of parameter identification for AC servo systems.

(4) Based on the simulation system, the exhaust algorithm was designed to implement the parameter optimization. It has been proved that the parameters optimized by the simulation system can achieve favorable experimental results.

- Seok B L. Closed-loop estimation of permanent magnet synchronous motor parameters by PI controller gain tuning. *IEEE T Energy Convers*, 2006, 21(4): 863–870[[doi](#)]
- Min Y, Chen L. A transient energy function for power systems including the induction motor model. *Sci China Ser E-Tech Sci*, 2007, 50(5): 575–584
- Yusivar F, Wakao S. Minimum requirements of motor vector control modeling and simulation utilizing C MEX S-function in matlab/simulink. *IEEE*, 2001, 1(3): 315–321
- Ning J, Kevin B, Philip A, et al. A simulation method for the firing sequences of motor units. *J Electromyogr Kines*, 2007, 17(5): 527–534
- Alfred V, Aho R S, Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools*. Beijing: Chinese People Press, 2002. 112–117
- Petrovic V, Ortega R. Design and implementation of an adaptive controller for torque ripple minimization in PM synchronous motors. *IEEE T PE*, 2000, 15(5): 871–880
- Christopher W, David R. *A Retargetable C Compiler Design and Implementation*. Beijing: Electrical Industry Press, 2005. 294–298
- Grady B, James R, Ivar J. *The Unified Modeling Language User Guide*. Beijing: Mechanical Industry Press, 2006. 90–101
- Jabbar M A, Hla N P, Liu Z J, et al. Modeling and numerical simulation of a brushless permanent magnet DC motor in dynamic conditions by time-stepping technique. *IEEE T Ind Appl*, 2004, 40(3): 763–770[[doi](#)]
- Kim Y S, Choi Y K. Speed-sensorless vector control for permanent-magnet synchronous motors based on instantaneous reactive power in the wide-speed region. *IEE Proc Electr Power*, 2005, 152(5): 1343–1349[[doi](#)]
- Liu H T, Mei J P, Zhao X M, et al. Inverse dynamics and servomotor parameter estimation of a 2-DOF spherical parallel mechanism. *Sci China Ser E-Tech Sci*, 2008, 51(3): 277–287
- Zhang D L, Ai X, Xu C J, et al. DSP-based software AC servo systems with PM synchronous motors. *Electr Mach Sys*, 2001, 2(6): 755–758
- Morimoto S, Sanada M, Takeda Y. Inverter-driven synchronous motor for constant power. *IEEE T Auto Contr*, 1996, 2(1): 18–24