Design of PMSM Speed Control system based on simulink Model

Wengliang Liu, Pengfei Xu
Department of Electrical Engineering, Tianjin University Of Science & Technology, Tianjin 300222, China
a596673152@163.com

Abstract. The application of motor speed control system in industrial production, automatic control, aerospace and other fields is more and more extensive, but more and more complex speed control system makes the programming of digital controller become more and more difficult. This paper presents a simulation model of permanent magnet synchronous motor (PMSM) speed control system based on magnetic field oriented control (FOC) based on MATLAB/Simulink software, through which the executable code engineering files of DSP are generated automatically. In addition, the algorithm model built by Simulink is only suitable for the pure software environment, and the model can not be directly applied to the actual digital control. Therefore, this paper first builds the FOC control model based on pure software simulation, then converts it into a pure discrete control model. Finally, by configuring some necessary parameters, the model automatically generates DSP executable C language code. This method accelerates the development speed of digital controller code in the speed regulation system, reduces the difficulty of the system code development, and has great practical application value.

1. Introduction

The early motor speed regulation system was an open-loop control system composed of pure hardware circuit. Although the control was reliable and easy to operate, but because of the soft mechanical characteristics, it could not correct the error and the poor anti-interference ability and so on. It limits the rapid development of speed control system. With the development of power electronics technology and micro-controller, digital control mode has gradually replaced the early pure hardware control mode because of its advantages of high control precision, small volume and low cost. The core of digital controller is micro-controller. DSP is a kind of micro-controller with excellent performance. It can realize precise control of PWM, high precision sampling, high speed and stable communication and so on.\[1\]

At present, domestic motor speed regulation technology has been quite mature, widely used in motor control, grid-connected power generation, industrial production and other occasions. However, with the continuous improvement of the function and performance requirements of the speed regulation system, the complexity of the algorithm and code of the speed regulation system also increases, and it takes a lot of manpower and time to write and maintain the control code. The development cycle of new products has been greatly increased, thus limiting the rapid development and popularity of digital controllers.\[2\] In this paper, MATLAB/Simulink is used for simulation modeling, which can greatly improve the development efficiency of the control system, and play a great role in the verification of various control strategies. MATLAB is a highly integrated development...
software platform, its function is very perfect, design, simulation, debugging, implementation are integrated together. The Simulink simulation toolbox in MATLAB encapsulates all kinds of tools into component library humanly, and can realize complex operation function only by simply dragging and connecting.\[3\]

In recent years, in order to directly apply the Simulink model to the actual product development, Mathworks has introduced many hardware support packages, which can support the C language code and Verilog HDL code needed by ARM, DSP, FPGA and other micro-controllers. The process of modeling and code generation is modular design.\[4\] In addition, when using Simulink for automatic code generation, the following issues need to be noted:

a) The solution method in Simulink can be continuous or discrete, but in DSP, the solution can only be a pure discrete algorithm.\[5\]

b) Simulink supports more comprehensive data types, while DSP only applies fixed-point data and floating-point data. (the range of data types varies from processor to processor, for example, the DSP, of the TMS320F2812 model supports only fixed-point data types, while the DSP of the TMS320F28335 model supports fixed-point and floating-point data types.\[6\])

c) The simulation time in Simulink is generally finite, but the DSP is running without time limit.

d) The step size in Simulink can be fixed step size or variable step size, while step size in DSP is fixed.

This paper mainly aims at the FOC speed control of permanent magnet synchronous motor (PMSM), and designs the PMSM based on Simulink model and automatically generates the code. First, the software simulation environment of FOC control is built to verify the correctness and effectiveness of the control method. Secondly, the continuous time of the model is modified to a discrete simulation model, and the DSP executable engineering file is generated automatically. Finally, the validity of the code is verified by connecting the experimental motor.

2. Simulation of FOC Software based on SIMULINK

2.1. $i_d = 0$ Vector control mode

The vector control strategy can be divided into: $i_d = 0$ control, $\cos \varphi = 1$ control, maximum torque / current control, constant flux chain control and so on according to the different operating environment, speed range and performance requirements of the motor. Because $i_d = 0$ control strategy has the advantages of good torque performance and simple control, $i_d = 0$ control method is adopted in this paper. In the d-q coordinate system, the angle between the Q axis and the stator current vector is always zero, and the electromagnetic torque $T_e$ is proportional to the Q-axis current vector $i_q$. Thus, the control problem of three-phase AC motor is transformed into the control of DC motor, which greatly reduces the difficulty of control. Figure 1 is a space vector diagram under $i_d = 0$ control mode.

Since it is the control mode of $i_d = 0$, the flux chains of the d and Q axes are as follows:

$$\begin{cases}
\psi_d = \psi_f \\
\psi_q = L_d i_q
\end{cases}$$  \hspace{1cm} (1)

Fig.1 Space vector diagram of $i_d = 0$ control mode
Voltage equation:
\[
\begin{align*}
    u_d &= -\omega \psi_q = -\omega L_{q} i_s \\
    u_q &= R_d i_q + \omega \psi_d = R_d i_s + \omega \psi_f \\
\end{align*}
\] (2)

Electromagnetic torque equation:
\[
    T_e = \frac{3}{2} n_p (\psi_{d} i_q - \psi_{q} i_d) = \frac{3}{2} n_p \psi_f i_s
\] (3)

At this point, the value of angle $\delta$ can be obtained in conjunction with figure 1:
\[
    \delta = \arctan\left(\frac{\psi_{d}}{\psi_f}\right) = \arctan\left(\frac{\psi_{q}}{\psi_{d}}\right)
\] (4)

The power factor angle $\varphi$ is:
\[
    \varphi = \arctan\left(\frac{\omega L_{d} i_s}{\omega \psi_f + R_d i_s}\right) = \arctan\left(-\frac{u_d}{u_q}\right)
\] (5)

If the stator winding $R_s$ is ignored, that is $\delta = \varphi$, the power factor of the motor can be obtained as follows:
\[
    \cos \varphi = \frac{\psi_f}{\sqrt{\psi_f^2 + L_d^2 i_s^2}}
\] (6)

2.2. Software simulation of $i_d = 0$ control mode

The control system of motor adopts double closed loop control structure, which is speed loop and current loop control respectively. The speed loop plays an important role in the whole regulation process. It can make the actual speed of the motor fast, follow the input given stably, realize the speed regulation without static check, and greatly reduce the steady-state error of the system. At the same time, because the speed loop is a closed-loop system, it has good immunity to the sudden change of load. The current loop mainly serves as a protective circuit to limit the current over-current. At the same time, the current loop can restrain the disturbance of the grid voltage. When starting and braking, the motor can work under the maximum allowable current, so that the system can enter steady state in the shortest time. According to the double closed loop control structure and $i_d = 0$ control mode, the system simulation model is set up as shown in figure 2.

Fig. 2 Simulation diagram of $i_d = 0$ vector control speed regulation system

The system consists of a speed PI regulator, a current PI regulator, a Park and Clark transform, a matrix converter, a motor model, a current detection device and a speed detection device. Both the speed PI regulator and the current PI regulator are regulators with finite amplitude.

Using Matlab/Simulink to simulate, the simulation time is 1s and the solver is ode23tb. The system starts without load, the initial speed is given as 1000rpm, the load torque of 5Nm is added at 0.2s, the load torque of 5Nm is added at 0.4s, and the initial rotational speed is raised to 1200rpm at 0.6s. The initial rotational speed is reduced to 600rpm at 0.8s.
Figure 3 shows the waveform of the d axis, Q axis and A phase current in the motor. They are displayed in an oscilloscope, which is advantageous to the comparison of the data.

The average value of d axis current component is approximately 0. A phase current when starting and adjusting speed, and it is also approximate to continuous sine wave. The current of Q axis changes with the change of load and rotation speed: at 0.2s and 0.4s, the current of Q axis increases correspondingly because of the increase of load. At 0.6 s, the Q axis current increases by 0.05 s due to the increase of rotational speed. However, due to the current limitation, the current is quickly returned to steady operation. The rotational speed drops suddenly at 0.8 s, and the Q axis current decreases first, but quickly returns to steady state. The system can respond quickly and run stably.

Figure 4 is the speed waveform, figure 5 is the speed waveform at 0.2 s and 0.4 s, and figure 6 is the electromagnetic torque waveform. It can be seen that the actual speed can closely follow the given, increase the load, the speed of the basic no change, so the system has a good anti-disturbance ability. In addition, the system can provide the maximum electromagnetic torque when the motor starts and the speed rises, which ensures that the motor can have the maximum acceleration and does not appear the phenomenon of over-current. When the speed is stable, the load torque and the electromagnetic torque can balance each other to ensure the steady speed of the motor.

Through the software simulation of Simulink, it can be seen that the FOC control can accurately follow the given speed, resist the disturbance of the load, and have excellent dynamic and static performance.

3. FOC discrete algorithm Model based on SIMULINK

When building a FOC algorithm model, you first have to comply with the four requirements mentioned above. In order to make the sampled data feedback value real-time, the ADC interrupt trigger module is used to ensure data sampling, algorithm execution and PWM signal generation synchronization. The collection of rotational speed is carried out independently and continuously, and the collected rotational speed information is continuously compared with the input given.

3.1. Speed acquisition module

The algorithm model of speed acquisition module is shown in figure 7. The QEP module receives and counts the pulse of the photoelectric encoder, and can convert the pulse number to the rotational speed value by formula (7).
\[
    n = \frac{60(c_k - c_{k-1})}{0.01 \times 2048 \times 4} \tag{7}
\]

In the formula, \(c_k\) and \(c_{k-1}\) are the number of pulses in the adjacent time period.

3.2. Interrupt trigger module
The model of the interrupt trigger module is shown in figures 8 and 9. The interrupt trigger module includes voltage and current sampling module, Clark and Park transform module, Park inverse transform, magnetic field orientation module, rotational speed regulation module, current regulation module and SVPWM modulation module.

3.2.1. Voltage and current sampling module
In this paper, the FOC algorithm is used to control the feedback of three analog variables: motor current \(I_a\) and \(I_b\), DC bus voltage \(U_{dc}\). Therefore, the channels of the ADC sampling module correspond to the A0, A1, A3. The collected data need to be adjusted by (8) ~ (10) in order to be added to the following algorithm. The algorithm model is shown in figure 10.

\[
    I_a = \frac{A_0 + IUR - 2150}{IUS} \tag{8}
\]

\[
    I_b = \frac{A_1 + IVR - 2150}{IVS} \tag{9}
\]

\[
    U_{dc} = \frac{A_3}{4096} \times 350 \tag{10}
\]

The IUR and IVR in the above formula are the correction terms of the current signal, which is used to calibrate the current signal. The IUS and IVS are obtained by adjusting the experiment.
3.2.2. Clark and Park transform module

The control of the three-phase motor is to transform the ABC three-phase coordinate system into the rotating d-q coordinate system, which greatly simplifies the control difficulty of the motor. The Clark transformation matrix transforms the ABC three-phase coordinate system into a stationary α-β coordinate system, as shown in the formula (11) and the figure 11, and then the Park transformation transforms the stationary α-β coordinate system into a rotating d-q coordinate system. As shown in formula (12) and figure 12.

Clark transformation:

\[
\begin{bmatrix}
I_d \\
I_q
\end{bmatrix} =
\begin{bmatrix}
\frac{\sqrt{2}}{2} & 0 \\
\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2}
\end{bmatrix}
\begin{bmatrix}
I_a \\
I_b
\end{bmatrix}
\]

Park transformation:

\[
\begin{bmatrix}
i_d \\
i_q
\end{bmatrix} =
\begin{bmatrix}
\cos \theta & \sin \theta \\
-\sin \theta & \cos \theta
\end{bmatrix}
\begin{bmatrix}
i_\alpha \\
i_\beta
\end{bmatrix}
\]

3.2.3. Speed and current regulation module

It is worth noting that in the rotational speed adjustment module, the PI regulation formula in figure 2 is:

\[y = (k_p + \frac{k_i}{s})e\]  \hspace{1cm} (13)

Where \(k_p\) is proportional magnification coefficient and \(k_i\) is integral coefficient. This is a continuous-time PI regulator, so it should be discretized by bilinear transformation\(^8\) to

\[s = \frac{T}{2} \frac{z - 1}{z + 1}\]  \hspace{1cm} (14)

The algorithm model of the speed regulator is shown in figure 13.

Fig. 13 algorithm model of speed regulator

The current regulator is the same, and only after bilinear transformation can the algorithm model be added.
3.2.4. SVPWM modulation module
The SVPWM modulation of the algorithm model is the same as that of the pure software part of SVPWM modulation. Unlike the software simulation, the final output waveform does not compare with the triangle wave output PWM waveform, but directly into the DSP EPWM module, because the DSP EVA event manager comes with the comparison module. Can be compared with the input waveform, output the desired PWM waveform. In addition, the CPU frequency of F2812 chip used in this paper is 150MHz, PWM frequency is 2.5kHz, so the clock cycle number is \((150MHz/2.5kHz) / 2=30000\). The hardware switch is different from the ideal switch in the simulation environment. It needs to set the dead-zone time: the dead-zone coefficient is set to 8 and the dead-zone time is set to 15 clock cycles.

4. Code Generation of FOC algorithm

4.1. Configuration of Model parameters
After the FOC algorithm model has been built, the parameters are configured in Simulink to generate code automatically, and the algorithm type is set as Fixed-step, algorithm mode, and the fixed step size of discrete, is set to auto. In the Hardware Implementation option, configure the DSP model, using TI's TMS320F2812 model DSP. In the Code Generation option, the System target file column is set to idelink_ert.tlc, Language set to C. In the Coder Target option, set Build_and_execute (download and execute), Board to C2000 Custom, Processor set to F2812.

4.2. Code automatic report generation
After the above settings, the algorithm model can be compiled. After the completion of the code compilation, Matlab will transfer the code to the CCS, after which the program will automatically download to DSP and run, or you can adjust the code in CCS. The code generation report is shown in figure 14. In the code generation report, you can see the code generation of each sub-module through the link, you can easily see the specific register configuration parameters, model parameters and model corresponding code.

![Code generation report for the model](image)

Fig. 14 Code generation report for the model

4.3. Experimental result
The user interface used in the experiment is LABVIEW 8.0 software, which can deliver the given initial speed to DSP in real time, and does not need DSP to stop running or reburn the code.

![No-load start waveform at given speed 1000rpm](image)

Fig. 15 No-load start waveform at given speed 1000rpm
The code generated above is burned into DSP, and the motor is started under the condition of no load, with a given speed of 1000rpm. The experimental results observed at the LABVIEW control interface are shown in figure 15. The red waveform is the reference waveform and the white waveform is the actual speed waveform.

The waveform when the speed suddenly rises from 600rpm to 1000rpm is shown in figure 16.

![Figure 16 Waveform of a given rotational speed when it suddenly rises from 600rpm to 1000rpm](image)

From figure 16, it can be seen that the motor response curve under the control of FOC algorithm model is good, the actual speed can closely follow the input reference speed, and the motor speed control is better. The experimental results are basically similar to those of the software simulation, and the waveform can agree with each other, which proves the validity and accuracy of generating code automatically by using the Simulink model.

5. Conclusion
This paper presents a design method of automatic code generation based on MATLAB/Simulink model. At first, the FOC pure software simulation based on Simulink is carried out, and the speed transformation curve of the motor is simulated by using the control method of \( i_d = 0 \). Secondly, the pure software simulation is discretized, and the discrete FOC algorithm model is built. After configuring the parameters of the model, the executable code of DSP is generated automatically. After the operation of the test bench, the waveform of rotational speed transformation is similar to that of pure software simulation, which proves the validity of the model-based design method. Using this method, the development cycle of new product can be greatly reduced, and the complicated code writing and maintenance can be eliminated, so it has higher practical application value.

References
[1] Guo, Jing. Research and implementation of SVPWM Inverter based on DSP[D]. University Of Science and Technology Of Taiyuan. 2014.
[2] Zhu, Yalong. Design of permanent Magnet synchronous Motor Control Software based on Matlab Code Generation[D]. Jilin University. 2014.
[3] Zhou, Dan; Zeng, Yuhang. C Code Generation of DC Motor Speed Control system based on Matlab/Simulink[J]. Application of single Chip Microcomputer and embedded system. 2015(10).
[4] Wang, Qiaoming; Li, Zhongjian; Jiang, Dayu. Research on DSP automatic Code Generation Technology of Matlab platform[J]. Modern electronic technology. 2012 (7).
[5] Zhang, Weifeng; Zhang, Yanhui; Wang, Ruichun; Pan, Xiaonin. “A model-based DSP control platform for rapid prototype of SVPWM.” International Conference on Signal Processing Proceedings. ICSIP, p 2523-2526, 2010.
[6] Sun, Haiyang; Teng, Jianfu. Improved indirect discretization Design of Fractional Calculator[J]. Computer Engineering and Application. 2012(28).