Development of a SOPC for PMSM Drives

Ying-Shieh Kung*, Member IEEE, Pin-Ging Huang
Department of Electrical Engineering
Southern Taiwan University of Technology
Yung-Kang City, Tainan Hsien, 710, Taiwan
kung@mail.statn.edu.tw, tony91@msl1.u1.com.tw

Chien-Wu Chen
Energy & Resources Laboratories
Industrial Technology Research Institute
Chutung, Hsinchu, 310, Taiwan
jeff_chen@itri.org.tw

Abstract–a system-on-a-programmable-chip (SOPC) using an Altera FPGA (Stratix EP1S10) and a Nios embedded processor to develop drive system for permanent magnet synchronous motor (PMSM) is presented in this paper. In this SOPC, it has two modules to construct a fully digital drive controller. One module performs the function of the space vector loop control and motion trajectory control of PMSM drives. The other module performs the function of the current loop control for PMSM drives, which includes vector control, SVPWM generation, coordinate transformation, PI controller and the pulse detection of the quadrature encoder. The former is implemented by software using Nios embedded processor due to the need of complicated control algorithm and low sampling frequency control (motion trajectory control & position control: less than 1kHz). The latter is implemented by hardware due to the need of high sampling frequency control (current loop: 16kHz, PWM circuit: 4-8MHz) but simple computation. To confirm the effectiveness of the proposed SOPC, an experimental system with a PMSM, a FPGA experimental board, an inverter and a rectifier has been set up and some experimental results have been validated with the theoretical ones.

I. INTRODUCTION

Owing to the advantages of the superior power density and high efficiency, permanent magnet synchronous motors (PMSM) have widely used in many automation control fields as an actuators. But due to the keen competition among manufacturers, how to design and implement a high performance, compact size and low cost servo motor drives become an important issue.

In recent years, for the progress of VLSI technology, the field programmable gate array (FPGA) has brought more attention due to the advantages of their programmable hard-wired feature, fast time-to-market, shorter design cycle, embedding processor, low power consumption and higher density for the implementation of the digital system [1-2]. Therefore, many practical applications in ac motor control have been studied [3-7]. But in those researches, the FPGA is almost used to realize the hardware part of the overall system.

Nowadays, SOC (System on a Chip) and IP (Intellectual Property) designs can be developed and downloaded into FPGA to work with an embedded processor [8] to construct a SOPC (System on a Programmable Chip) environment. Due to this advantage, we design a fully digital motion control IC for PMSM drives under this SOPC environment, and it is shown in Fig.1. In Fig.1, the current control scheme can be realized by hardware in programmable logic devices (PLD) and the motion control algorithm can be realized by software using Nios processor. Therefore, all of the function needed for building up a PMSM drives can be integrated and realized in this single SOPC chip. At last, we prove that the development of PMSM drives is more compact, programmable, flexible, robust and easy implement under this SOPC environment.

II. DESIGN METHOD OF A SOPC FOR PMSM

The architecture of the proposed motion control SOPC for PMSM drives is shown in Fig. 2, in which the current loop control is implemented by hardware but the position/speed loop control and the motion trajectory calculation is implemented by software using Nios embedding processor.

A. Current vector control

The current vector control of PMSM is described in the dotted line of Fig.2, which includes PI controller, coordinate transformation of Clarke, Clarke-1, Park, Park-1, SVPWM, capture function of current value and encoder pulses etc. After using vector control (current i_d to 0 in Fig.2), it will make the nonlinear and coupling characteristics of PMSM become decouple, and control a PMSM is easy as to control a DC motor. Thus, the torque magnitude control of PMSM is only need to control the current in the direction of q-axis. The SVPWM scheme in Fig.2 is too complicated to describe here, and reader can refer to [4]. Therefore, considering the mechanical load term, the electrical and mechanical equations of PMSM can be written as the following equations,

\[
T_e = K_i i_q \quad (1)
\]

\[
J_m \frac{d}{dt} \omega + B_m \omega = T_e - T_L \quad (2)
\]

where \(T_e, T_L, K_i, J_m, B_m\) are motor torque, load torque, torque constant, inertial value and damping ratio, respectively.

B. Fuzzy controller for position control loop

The position control adopted by fuzzy controller in Fig.2 is implemented by software using Nios embedded processor. The fuzzy controller includes fuzzification, fuzzy rules, inference mechanism and defuzzification [9]. In Fig.2, the tracking error and its error change \(e, \Delta e\) are defined by

\[
e(n) = \theta_d(n) - \theta_n(n) \quad (3)
\]

\[
\Delta e(n) = e(n) - e(n-1) \quad (4)
\]

and \(K_e, K_{\Delta e}\) are the input and output variable of fuzzy controller, respectively. The design procedure of the fuzzy controller is as follows:

1. **Fuzzification**
   - Define the input and output variables of the fuzzy controller.
   - Map the crisp values into fuzzy sets.

2. **Rule Base**
   - Establish a set of if-then rules.

3. **Inference Engine**
   - Apply the rules to the input variables.

4. **Defuzzification**
   - Convert the fuzzy output into a crisp value.

**Fig. 1.** SOPC-based PMSM drives system

- **Fig. 2.** SOPC for PMSM drives

- **Fig. 3.** SOPC-based control system

- **Fig. 4.** SOPC for PMSM drives

0-7803-8346-X/04/$20.00 ©2004 IEEE
Motion Control SOPC

Construction of fuzzy system with membership function.

where $x_i$ is input value, $\xi_e(\bullet)$ is output value, $\xi_i^m$ and $w^m_i$ are mean value and width of the triangular function, respectively.

- Define the fuzzy rule as $[A_1, A_2, E]$, they are symmetrical triangular membership function:

$$
\xi_e(x_i, \xi_i^m, w^m_i) = \begin{cases} 
0 & x_i \leq \xi_i^m - w^m_i / 2 \\
\frac{x_i - \xi_i^m + w^m_i / 2}{w^m_i / 2} & \xi_i^m - w^m_i / 2 < x_i < \xi_i^m \\
\frac{\xi_i^m - w^m_i / 2}{w^m_i / 2} & \xi_i^m < x_i < \xi_i^m + w^m_i / 2 \\
1 & x_i \geq \xi_i^m + w^m_i / 2
\end{cases}
$$

- Derive $M$ fuzzy control rules from the dynamic response characteristics [10], as follows

$$
\text{If } e \text{ is } A_1^m \text{ and } \Delta e \text{ is } A_2^m \text{ THEN } u_i \text{ is } E^m, \text{ m=1...M} \tag{6}
$$

- Construct the fuzzy system with $u_i(\xi, \Delta \theta)$ from those $M$ rules using the singleton fuzzifier, product-inference rule, and central average defuzzifier method. Therefore, (6) is replaced by the following expression:

$$
u_i(\xi, \Delta \theta) = \frac{\sum_{m=1}^{M} c_m \xi_i^m}{\sum_{m=1}^{M} \xi_i^m} \tag{7}
$$

where $c_1, c_2, ..., c_m$ denote the center of the output membership function.

C. Point-to-point motion control

In Fig. 2, the motion trajectory of motor adopts point-to-point motion control scheme with constant acceleration/ deceleration trapezoid velocity profile. In this scheme, the designed parameters are the total rotating angle, $\Delta \theta$ (rad or pulses), maximum angular velocity, $W$ (rad/s or pulses/s), acceleration/deceleration time, $T_{acc}$, and the sampling interval $t_d$. Therefore, from this velocity profile, the instantaneous value at each sampling, $\Delta \theta$, can be calculated and be taken as the position command. The computation procedure is as follows:

Step 1: Computation of the total running time

$$
T_1 = \Delta \theta / W \tag{8}
$$

Because the total running time has to larger than acceleration time $T_{acc}$, therefore $T = \text{MAX}(T_1, T_{acc}) \tag{9}$

Step 2: Modification of the total running time by times of the sampling interval.

$$
N' = \text{INT}(T_{acc} / t_d) \text{ and } T' = N' + T_{acc} \tag{10}
$$

Where $N'$ is the interpolation number and $[\ ]$ denotes Gauss function. Therefore, $T_{acc} = N' \ast t_d$ and $T' = N' \ast t_d \tag{11}$

Step 3: Modification of the maximum velocity

$$
W' = \Delta \theta / T' \tag{12}
$$

Step 4: Calculation of acceleration/deceleration value

$$
A = W' / T_{acc} \tag{13}
$$

Step 5: Calculation of the mid-position

(a) Acceleration region:

$$
\dot{\theta}_i = \dot{\theta}_{i0} + \frac{1}{2} A \ast t^2 \tag{14}
$$

where $t = n \ast t_d$, $0 < n \leq N'$, and the $\dot{\theta}_{i0}$ is the initial position.

(b) Constant speed region:

$$
\dot{\theta}_i = \dot{\theta}_i + W' \ast t \tag{15}
$$

where $t = n \ast t_d$, $0 < n \leq N1$, $\dot{\theta}_i$ is the final position value at acceleration region and $N1 = \text{INT}(N' - 2 \ast N')$.

(c) Deceleration region:

$$
\dot{\theta}_i = \dot{\theta}_i + (W' \ast t - \frac{1}{2} A \ast t^2) \tag{16}
$$

where $t = n \ast t_d$, $0 < n \leq N'$ and $\dot{\theta}_i$ is the final position value at constant speed region.

III. EXPERIMENTS AND RESULTS

The overall experimental system is depicted in Fig. 1, and it includes a FPGA (Stratix EP1S10), a voltage source IGBT inverter and a PMSM. The power of the PMSM is 300W and the rating speed is 2000rpm. An incremental optical encoder (1000 ppr) attached to PMSM has been installed as the rotor’s position sensor. The inverter has 6 sets
of IGBT type power transistors. The collector-emitter voltage of the IGBT is rating 600V, the gate-emitter voltage is rating ±20V, and the collector current in DC is rating 25A and in short time (1ms) is 50A. The photo-IC, Toshiba TLP250, is used for gate driving circuit of IGBT. Input signals of the inverter are PWM signals from FPGA device. The FPG chip adopts Altera Stratix EP1S10F780C6, which has 10,570 LEs, maximum 426 user I/O pins, 6 DSP blocks, total 920,448 RAM bits, and a Nios embedded processor which has a 16-bit or 32-bit configurable CPU core, 1 to 20Kbytes available on chip and maximum 4GBytes off-chip memory, can be downloaded into FPGA to work with IP design circuit to construct an SOPC environment. A custom software development kit (SDK) consists of a compiled library of software routines for the SOPC design, a Make-file for rebuilding the library, and C header files containing structures for each peripheral.

In implementation, the PWM switching frequency of inverter, dead-band and the control sampling frequency of position loop are designed with 16kHz, 1μs, and 1kHz, respectively. In proposed motion control SOPC, the current controller is implemented by PLD hardware and the motion controller is implemented by software using Nios processor. The former, digital circuits of current control IP, is depicted in Fig. 3 which includes circuits of PI controller, coordinate transformation of Clarke, Park, inverse Park, inverse Clarke and circuits of SVPWM, QEP and ADC conversion control, etc. The digital circuit of PI controller, designed by 3 adders, 2 multipliers, 2 D-type flip-flops and 3 max-min value limiters, are shown in Fig. 4. Due to the limit space of paper, except the circuit of PI controller, the description of other digital circuits are omitted here, reader can refer to [3-4]. The latter, that the position, speed and point-to-point motion controller are all implemented by software using Nios embedded processor and it will be downloaded and integrated into FPGA to work with current control IP to form a motion control SOPC for PMSM drive system. The flow charts of interrupt service routine (ISR) for motion control are plotted in Figs. 5-6. The overall circuits included Nios processor and current control IP shown in Fig.2, use 67% utility of Stratix EP1S10F780C6.

To verify the effectiveness of the proposed motion control SOPC for PMSM drives, the speed step response is first be test. When the speed command is 2 Hz square wave signal with magnitude 1200rpm and 1500 rpm, the rotor speed response with 0.025s rising time, zero steady state value but a 5% overshoot is measured and plotted in Fig. 7(a) and the response of current control effort \( \tau_r \) is also shown in Fig. 7(b). Secondly, we continue to test the performance of the position response. When the position command is 1 Hz square wave signal with magnitude 1000 pulses (π/2 radius), the measured rotor position response with 0.05s rising time, no overshoot and zero steady state value and the response of control effort, \( \tau_r \) with 1.8A peak value are shown in Fig. 8(a)~8(b), respectively. All of speed step response and position step response in Fig. 7~8 reveal a fast and good command tracking response. After making sure the performance of our proposed scheme in step position response, we further evaluate the tracking ability of a long distance motion control. In Fig. 2, the total displacement command \( \Delta \theta \) is transmitted from PC through Ethernet interface, and then the instantaneous position command \( \theta' \) at each sampling interval is calculated following the aforementioned design procedure of point-to-point motion control. To test the tracking performance of a point-to-point motion control, the total motion command \( \Delta \theta \) is designed with starting from 0 pulse move to 30000 pulses with step 10000 pulses increment, the maximum angular velocity \( \omega \) is designed with 750 rpm, the acceleration/deceleration time is 50000 pulses/s², and the sampling interval of motion control loop is 200Hz. The trajectory tracking results corresponding with the aforementioned input command is shown in Fig. 9(a) ~ 9(b). It can be seen that no matter what in position or the velocity trajectory tracking, the measured rotor motion can gives a perfect tracking with target in Fig. 9. Therefore, the experimental results in Figs. 7~9 demonstrate the proposed SOPC of motion control for PMSM drives is effectiveness and correctness.

![Current control IP developed in Quart II environment](image-url)
IV. CONCLUSIONS

A SOPC for PMSM drives using a FPGA and a Nios embedded processor has been presented in this paper. This SOPC performs the function of a fully digital and high performance control for PMSM drives system, therefore the current vector control scheme, SVMWM generation, coordinate transformation, QEP detection, fuzzy position control strategy and motion trajectory control are all realized and integrated in this single SOPC. Some experimental results are validated a good performance while using this SOPC in PMSM drives. Finally, those techniques proposed in this paper can be easily expanded and applied in the multi-axis ac servo driving system (NC machine, robotics, etc.) and the 3C consumer products.

REFERENCE


Fig. 4. Digital circuit of PI controller

Fig. 5. The flow chart of control ISR

Fig. 6. The flow charts of motion trajectory planning and data communication

Fig. 7. (a) Step speed response under 1200-1500 rpm command (b) Response of control effort.

Fig. 8. (a) Step position response under 1000~2000 pulses square wave command (b) Response of control effort.

Fig. 9. (a) Long-distance motion trajectory tracking (b) Velocity tracking corresponding with (a).