Simulink/ModelSim Co-Simulation of Sensorless PMSM Speed Controller

[1]Ying-Shieh Kung and [2]Nguyen Vu Quynh
[1,2]Department of Electrical Engineering
Southern Taiwan University, Tainan, Taiwan
[2]Lac Hong University, Vietnam
[1]kung@mail.stut.edu.tw, [2]vuquynh@lhu.edu.vn

[3]Chung-Chun Huang and [4]Liang-Chiao Huang
[3,4]Green Energy and Environment Research Laboratories
Industrial Technology Research Institute
Hsinchu, Taiwan
[3]CCHuang@itri.org.tw, [4]liang.qiao@itri.org.tw

***Abstract*—Based on Simulink/Modelsim co-simulation technology, the design of a sensorless control IP (Intellectual Property) for PMSM (Permanent Magnet Synchronous Motor) drive is presented in this paper. Firstly, a mathematical model for PMSM is derived and the vector control is adopted. Secondly, a rotor flux position is estimated by using a sliding mode observer (SMO). These estimated values are feed-backed to the current loop for vector control and to the speed loop for speed control. Thirdly, the Very-High-Speed IC Hardware Description Language (VHDL) is adopted to describe the behavior of the sensorless speed control IP which includes the circuits of space vector pulse width modulation (SVPWM), coordinate transformation, SMO, fuzzy controller, etc. Fourthly, the simulation work is performed by MATLAB/Simulink and ModelSim co-simulation mode, provided by Electronic Design Automation (EDA) Simulator Link. The PMSM, inverter and speed command are performed in Simulink and the sensorless speed control IP of PMSM drive is executed in ModelSim. Finally, the co-simulation results validate the effectiveness of the sensorless PMSM speed control system.**

Keywords-PMSM; Simulink/Modelsim co-simulation; Sliding mode observer; Sensorless speed control; Fuzzy controller; VHDL.

I. INTRODUCTION

PMSM has been increasingly used in many automation control fields as actuators, due to its advantages of superior power density, high-performance motion control with fast speed and better accuracy. However, conventional motor control needs a speed sensor or an optical encoder to measure the rotor speed and feedback it to the controller for ensuring the precision speed control. Such sensor presents some disadvantages such as drive cost, machine size, reliability and noise immunity. In recent year, a sensorless control without position and speed sensors for PMSM drive become a popular research topic in literature [1-7]. Those sensorless control strategies have sliding mode observer, Kalmam filter, neural network, etc. However, the back EMF and the sliding mode observer are suitable to be implemented by the fix-pointed processor and have been implemented by a digital signal processor (DSP) in most studies [4-5]. Unfortunately, DSP suffers from a long period of development and exhausts many resources of the CPU [8]. FPGA can provide another alternative solution in this issue. Especially, FPGA with programmable hard-wired feature, fast computation ability, shorter design cycle, embedding processor, low power consumption and higher density is better for the implementation of the digital system [9-10] than DSP.

Recently, a co-simulation work by Electronic Design Automation (EDA) Simulator Link has been gradually applied to verify the effectiveness of the Verilog and VHDL code in the motor drive system [11-14]. The EDA Simulator Link [15] provides a co-simulation interface between MALTAB or Simulink and HDL simulators-ModelSim [16]. Using it you can verify a VHDL, Verilog, or mixed-language implementation against your Simulink model or MATLAB algorithm [15]. Therefore, EDA Simulator Link lets you use MATLAB code and Simulink models as a test bench that generates stimulus for an HDL simulation and analyzes the simulation's response [15]. In this paper, a co-simulation by EDA Simulator Link is applied to sensorless speed control for PMSM drive and shown in Fig.1. The PMSM, inverter and speed command are performed in Simulink and the sensorless speed controller described by VHDL code is executed in ModelSim. Finally, some simulations results validate the effectiveness of the sensorless speed control system of PMSM drive.

II. SYSTEM DESCRIPTION OF PMSM DRIVE AND SENSORLESS SPEED CONTROLLER DESIGN

The sensorless speed control block diagram for PMSM drive is shown in Fig. 1. The modelling of PMSM, the SMO-based flux position estimation and the fuzzy controller are introduced as follows:

A. Mathematical Model of PMSM

The typical mathematical model of a PMSM is described, in two-axis *d-q* synchronous rotating reference frame, as follows

$$\frac{di_d}{dt} = -\frac{r_s}{L_d} i_d + \omega_e \frac{L_q}{L_d} i_q + \frac{1}{L_d} \mathrm{v}_d \tag{1}$$

$$\frac{di_q}{dt} = -\omega_e \frac{L_d}{L_q} i_d - \frac{r_s}{L_q} i_q - \omega_e \frac{K_\mathrm{E}}{L_q} + \frac{1}{L_q} \mathrm{v}_q \tag{2}$$

where v_d, v_q are the *d* and *q* axis voltages; i_d, i_q, are the *d* and *q* axis currents, r_s is the phase winding resistance; L_d, L_q are the *d* and *q* axis inductance; ω_e is the rotating speed of magnet flux; K_E is the permanent magnet flux linkage.

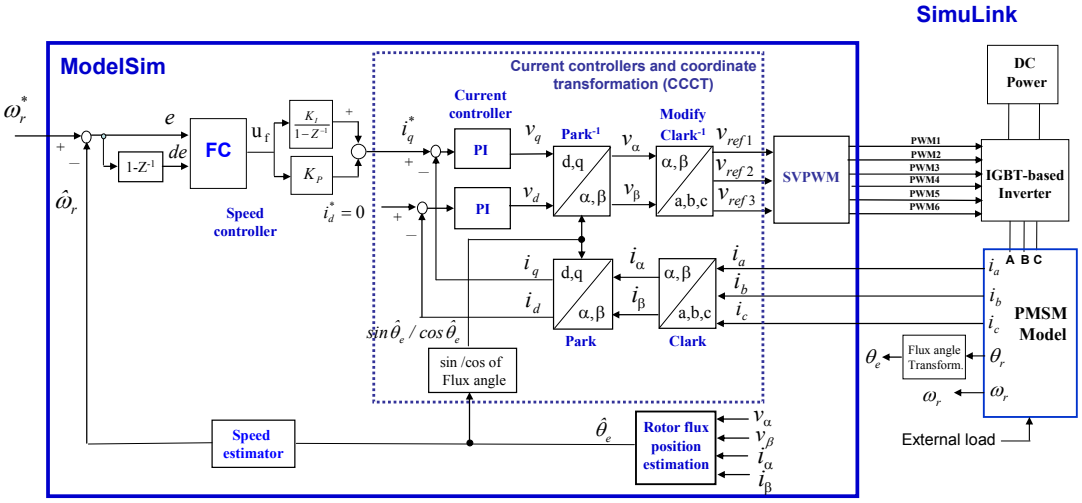

Fig.1 The sensorless speed control block diagram for PMSM drive

The current loop control of PMSM drive in Fig.1 is based on a vector control approach. That is, if the i_d is controlled to 0 in Fig.1, the PMSM will be decoupled and controlling a PMSM like to control a DC motor. Therefore, after decoupling, the torque of PMSM can be written as the following equation,

$$T_e = \frac{3P}{4} K_E i_q \triangleq K_t i_q \tag{3}$$

Considering the mechanical load, the overall dynamic equation of PMSM drive system is obtained by

$$J_m \frac{d}{dt}\omega_r + B_m \omega_r = T_e - T_L \tag{4}$$

where T_e is the motor torque, P is pole pairs, K_t is torque constant, J_m is the inertial value, B_m is damping ratio, T_L is the external torque, ω_r is rotor speed.

B. Design of the rotor flux position estimation

Rotor flux position estimation in Fig.1 constructed by a sliding mode observer (SMO), a bang-bang controller, a low-pass filter and a position computation is shown in Fig. 2. The detailed formulation is described as follows.

Firstly, the circuit equation of PMSM on the *d-q* rotating coordinate in (1) is re-formulated as

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} r_s + sL & -\omega_e L \\ \omega_e L & r_s + sL \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_e K_E \end{bmatrix} \tag{5}$$

Where $L \triangleq L_d = L_q$. Transforming (5) of the circuit equation of PMSM on the $\alpha - \beta$ fixed coordinate can be derived by the following equation

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} r_s + sL & 0 \\ 0 & r_s + sL \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \omega_e K_E \begin{bmatrix} -\sin\theta_e \\ \cos\theta_e \end{bmatrix} \tag{6}$$

where $[v_\alpha \; v_\beta]^T$ is voltage on fixed coordinate; $[i_\alpha \; i_\beta]^T$ is current on fixed coordinate; L is the inductance of the d-axis or q axis, respectively; θ_e is angular position at magnet flux; s is differential operator. In addition, in (6), let's define the EMF as

$$e = \begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix} \triangleq \omega_e K_E \begin{bmatrix} -\sin\theta_e \\ \cos\theta_e \end{bmatrix} \tag{7}$$

The EMF includes the position information from the flux.

Secondly, to easily observe the EMF, (6) is rewritten as the state space form by current variable,

$$\frac{d}{dt}\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = A \cdot \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \frac{1}{L}\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} - \frac{1}{L}\begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix} \tag{8}$$

where $A = \begin{bmatrix} -r_s/L & 0 \\ 0 & -r_s/L \end{bmatrix}$ (9)

Thirdly, a sliding mode observer is designed by

$$\frac{d}{dt}\begin{bmatrix} \hat{i}_\alpha \\ \hat{i}_\beta \end{bmatrix} = A \cdot \begin{bmatrix} \hat{i}_\alpha \\ \hat{i}_\beta \end{bmatrix} + \frac{1}{L}\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} - \frac{1}{L}\begin{bmatrix} z_\alpha \\ z_\beta \end{bmatrix} \tag{10}$$

where the $[\hat{i}_\alpha \; \hat{i}_\beta]^T$ is the estimated current on fixed coordinate and the Z is defined in (10) which is the output gain of the bang-bang controller in Fig.2.

$$Z = \begin{bmatrix} z_\alpha \\ z_\beta \end{bmatrix} \triangleq k * sign\left(\begin{bmatrix} \hat{i}_\alpha - i_\alpha \\ \hat{i}_\beta - i_\beta \end{bmatrix}\right) \tag{11}$$

where the current error is defined by $e_{cur} \triangleq [\tilde{i}_\alpha \quad \tilde{i}_\beta]^T = [\hat{i}_\alpha - i_\alpha \quad \hat{i}_\beta - i_\beta]^T$. Further, if we choose the k be large enough, the inequality in (12) can be reached

$$e_{cur}^T \dot{e}_{cur} < 0 \tag{12}$$

and the SMO can enter into sliding mode condition. Therefore, it generates the results of $e_{cur} = \dot{e}_{cur} = 0$. Substituting the result into (8) and (10), the Z in (10) will approach to EMF in (6),

$$\begin{bmatrix} z_\alpha \\ z_\beta \end{bmatrix} = \begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix} = \omega_e K_E \begin{bmatrix} -\sin\theta_e \\ \cos\theta_e \end{bmatrix} \tag{13}$$

Fourthly, to alleviate the high frequency switching in bang-bang control, a low-pass filter is applied in Fig.2,

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix}\hat{e}_\alpha\\ \hat{e}_\beta\end{bmatrix} = -\omega_0\begin{bmatrix}\hat{e}_\alpha\\ \hat{e}_\beta\end{bmatrix} + \omega_0\begin{bmatrix}z_\alpha\\ z_\beta\end{bmatrix} \tag{14}$$

where $\omega_0 = 2\pi f_0$. Finally, the rotor position $\hat{\theta}_e$ can be computed by

$$\hat{\theta}_e = \tan^{-1}(-\frac{\hat{e}_\alpha}{\hat{e}_\beta}) \tag{15}$$

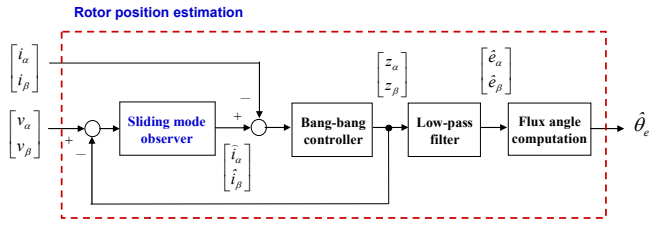

Fig.2 Rotor flux position estimation based on SMO

In implementation, the above formulations in the continuous system have to transfer to the discrete system. Besides, we use $[\hat{e}_\alpha\ \hat{e}_\beta]^T$ instead of $[z_\alpha\ z_\beta]^T$ as the feedback value in SMO; therefore, the difference equation of the modified sliding mode observer in (10) is

$$\begin{bmatrix}\hat{i}_\alpha(n+1)\\ \hat{i}_\beta(n+1)\end{bmatrix} = \begin{bmatrix}\phi & 0\\ 0 & \phi\end{bmatrix}\begin{bmatrix}\hat{i}_\alpha(n)\\ \hat{i}_\beta(n)\end{bmatrix} + \psi\begin{bmatrix}v_\alpha(n)\\ v_\beta(n)\end{bmatrix} - \psi\begin{bmatrix}\hat{e}_\alpha(n)\\ \hat{e}_\beta(n)\end{bmatrix} \tag{16}$$

where $\phi \triangleq e^{-\frac{r_s}{L}T_s}$, $\psi \triangleq \frac{1}{r_s}(1-e^{-\frac{r_s}{L}T_s})$ and T_s is the sampling time. In addition, from (14), the difference equation of the EMF estimation can also be expressed by

$$\begin{bmatrix}\hat{e}_\alpha(n+1)\\ \hat{e}_\beta(n+1)\end{bmatrix} = \begin{bmatrix}\hat{e}_\alpha(n)\\ \hat{e}_\beta(n)\end{bmatrix} + 2\pi f_0\begin{bmatrix}z_\alpha(n)-\hat{e}_\alpha(n)\\ z_\beta(n)-\hat{e}_\beta(n)\end{bmatrix} \tag{17}$$

Once the EEMF is estimated, the estimated rotor position $\hat{\theta}_e$ can be directly computed by

$$\hat{\theta}_e(n) = \tan^{-1}(-\frac{\hat{e}_\alpha(n)}{\hat{e}_\beta(n)}) \tag{18}$$

Finally, a summary for estimating the rotor position is shown by the following design procedures:

Step1: Estimate the estimated current by SMO in (16).

Step2: Calculate the current error by $\tilde{i}_\alpha(n) = \hat{i}_\alpha(n) - i_\alpha(n)$ and

$\tilde{i}_\beta(n) = \hat{i}_\beta(n) - i_\beta(n)$

Step3: Obtain the Z gain of the current observer in (11)

Step4: Estimate the EMF in (17).

Step5: Obtain the estimated rotor position in (18)

C. Fuzzy controller (FC)

The fuzzy controller in this study uses singleton fuzzifier, triangular membership function, product-inference rule and central average defuzzifier method. In Fig. 1, the tracking error e and the error change de are defined by

$$e(n) = \omega_r^*(n) - \omega_r(n) \tag{19}$$

$$de(n) = e(n) - e(n-1) \tag{20}$$

and u_f represents the output of the fuzzy controller. The ω_r^* is the speed command. The design procedure of the fuzzy controller is as follows:

(a) Take e, de and u_f as the input and output variable of fuzzy controller and define their linguist values E and dE in Fig.3 by $\{A_0, A_1, A_2, A_3, A_4, A_5, A_6\}$ and $\{B_0, B_1, B_2, B_3, B_4, B_5, B_6\}$, respectively. Each linguist value of E and dE are based on the symmetrical triangular membership function. The symmetrical triangular membership function are determined uniquely by three real numbers $\xi_1 \le \xi_2 \le \xi_3$, if one fixes $f(\xi_1) = f(\xi_3) = 0$ and $f(\xi_2) = 1$.

(b) Compute the membership degree of e and de. Figure 3 shows that the only two linguistic values are excited and gave a non-zero membership in any input value, and the membership degree $\mu_{A_i(e)}$ can be derived, in which the error e is located between e_i and e_{i+1}, two linguist values of A_i and A_{i+1} are excited, and the membership degree is obtained by

$$\mu_{A_i}(e) = \frac{e_{i+1} - e}{2} \quad \text{and} \quad \mu_{A_{i+1}}(e) = 1 - \mu_{A_i}(e) \tag{21}$$

where $e_{i+1} \triangleq -6 + 2*(i+1)$. Similar results can be obtained in computing the membership degree $\mu_{B_j(de)}$.

(c) Select the initial fuzzy control rules, such as,

$$IF\ \ e\ is\ A_i\ \ and\ \ \Delta e\ is\ B_j\ \ \ THEN\ \ u_f\ \ is\ \ c_{j,i} \tag{22}$$

where i and j = 0~6, A_i and B_j are fuzzy number, and $c_{j,i}$ is real number.

(d) Construct the fuzzy system $u_f(e,de)$ by using the singleton fuzzifier, product-inference rule, and central average defuzzifier method. For example, if the error e is located between e_i and e_{i+1}, and the error change de is located between de_j and de_{j+1}, only four linguistic values A_i, A_{i+1}, B_j, B_{j+1} and corresponding consequent values $c_{j,i}$, $c_{j+1,i}$, $c_{j,i+1}$, $c_{j+1,i+1}$ can be excited, and the (22) can be replaced by the following expression:

$$u_f(e,de) = \frac{\sum_{n=i}^{i+1}\sum_{m=j}^{j+1} c_{m,n}[\mu_{A_n}(e) * \mu_{B_m}(de)]}{\sum_{n=i}^{i+1}\sum_{m=j}^{j+1} \mu_{A_n}(e) * \mu_{B_m}(de)} \triangleq \sum_{n=i}^{i+1}\sum_{m=j}^{j+1} c_{m,n} * d_{n,m} \tag{23}$$

where $d_{n,m} \triangleq \mu_{A_n}(e) * \mu_{B_m}(de)$. And those $c_{m,n}$ denote the value of the singleton fuzzier.

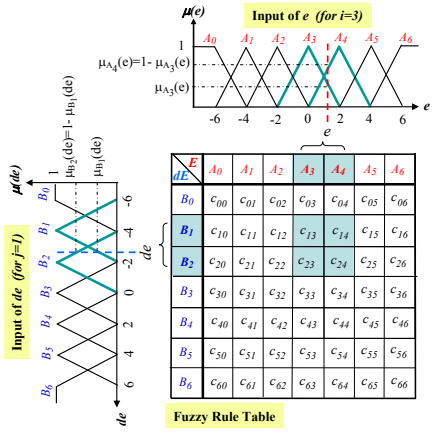

Fig. 3. The designed fuzzy controller

III. SIMULINK/MODELSIM CO-SIMULATION OF SENSORLESS SPEED CONTROLL FOR PMSM DRIVE

In Fig.1, it shows the sensorless speed control block diagram for PMSM drive and its Simulink/ModelSim co-simulation architecture is presented in Fig.4. The PMSM, IGBT-based inverter and speed command are performed in Simulink, and the sensorless speed controller described by VHDL code is executed in ModelSim with three works., The work-1 to work-3 of ModelSim in Fig.4 respectively performs the function of speed estimation and speed loop fuzzy controller, the function of current controller and coordinate transformation (CCCT) and SVPWM, and the function of SMO-based rotor flux position estimation. All works in ModelSim are described by VHDL. The sampling frequency of current and speed control is designed with 16 kHz and 2kHz, respectively. The clocks of 50MHz and 12.5MHz will supply all works of ModelSim.

A finite state machine (FSM) is employed to model the work-1 and work-3 of ModelSim, and shown in Fig.5 and Fig.6, respectively. In Fig.5, the data type adopts 16-bit length with Q15 format and 2's complement operation. The multiplier and adder apply Altera LPM (Library Parameterized Modules) standard. It manipulates 22 steps machine to carry out the overall computations. The steps s_0~s_1 execute the speed estimation, s_2~s_3 perform the computation of speed error and error change; steps s_4~s_7 execute the function of the fuzzification; s_8 describe the look-up table and s_9~s_{17} defuzzification; and steps s_{18}~s_{21} execute the computation of PI controller and command output. The *SD* is the section determination of *e* and *de* and the *RS,1* represents the right shift function with one bit. The operation of each step in Fig.5 is 80ns (12.5MHz) in FPGA; therefore total 22 steps only need 1.76μs operation times. Further, In Fig.6, The data type adopts 12-bit length with Q11 format and 2's complement operation. The multiplier, adder and divider apply Altera LPM standard component but the arctan function uses our developed component. It manipulates 36 steps machine to carry out the overall computations. The steps s_0~s_8 execute the estimation of current value; steps s_9~s_{10} compute the current error; s_{11} is the bang-bang control; s_{12}~s_{15} describe the computation of EMF and s_{16}~s_{35} perform the computation of the rotor position. The operation of each step in Fig.6 is 80ns (12.5MHz) in FPGA; therefore total 36 steps only need 2.88μs operation times. In Fig.4 the circuit design of CCCT and SVPWM in work-2 of ModelSim refers to [9]. The FPGA (Altera) resource usages of work-1 to work-3 of ModelSim in Fig.4 are 2,043 LEs (Logic Elements) and 0RAM bits, 2,085 LEs and 24,576 RAM bits; 1,151LEs and 49,152 RAM bits, respectively.

IV. SIMULATION RESULTS

The co-simulation architecture for sensorless PMSM speed control system is shown in Fig.4. The SimPowerSystem blockset in the Simulink executes the PMSM and the inverter. The EDA simulator link for ModelSim executes the co-simulation using Verilog HDL code running in ModelSim program. The designed PMSM parameters used in simulation of Fig.4 are that pole pairs is 4, stator phase resistance is 1.3Ω, stator inductanc is 6.3mH, inertia is J=0.000108 kg*m^2 and friction factor is F=0.0013 N*m*s.

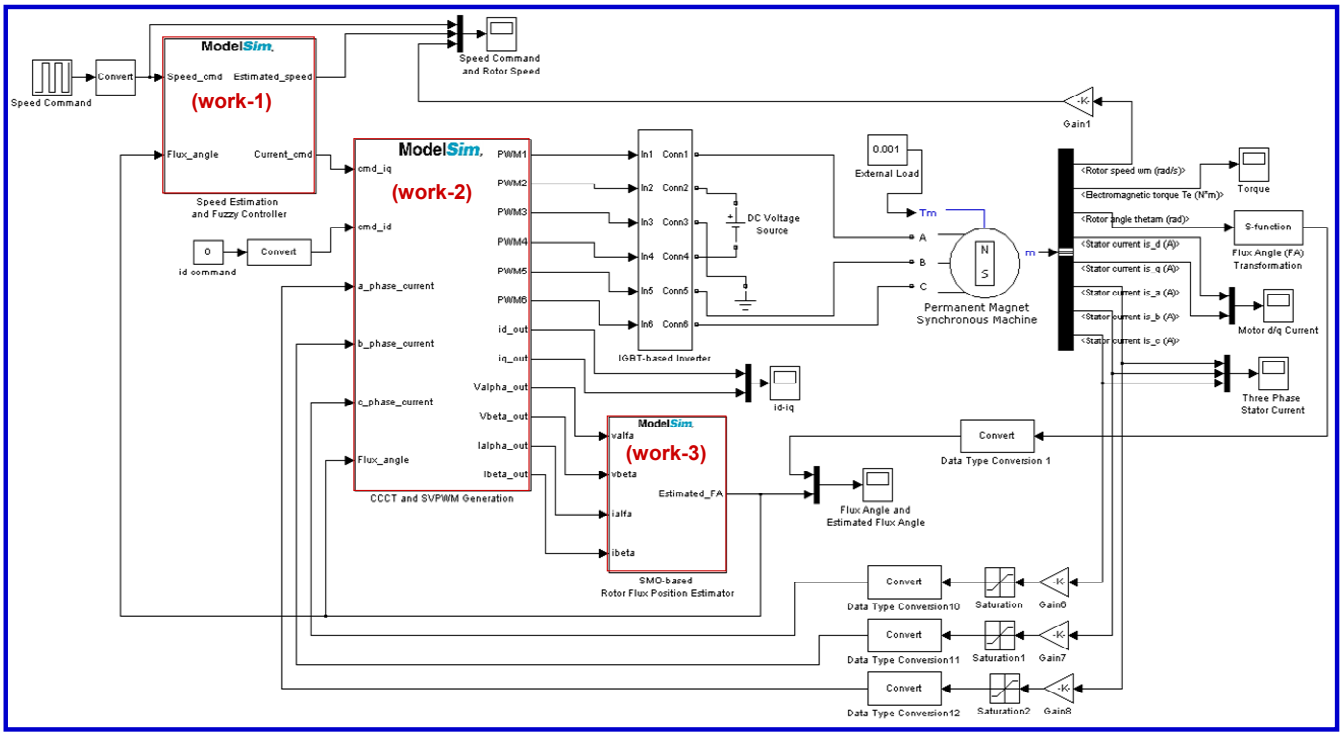

Fig.4 The Simulink/ModelSim co-simulation architecture for sensorless speed control of PMSM drive

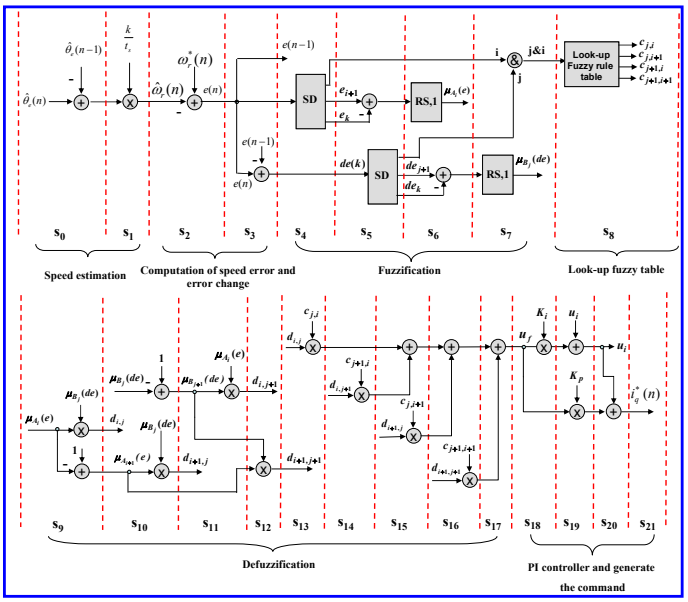

Fig.5 State diagram of an FSM for describing the speed estimation and FC

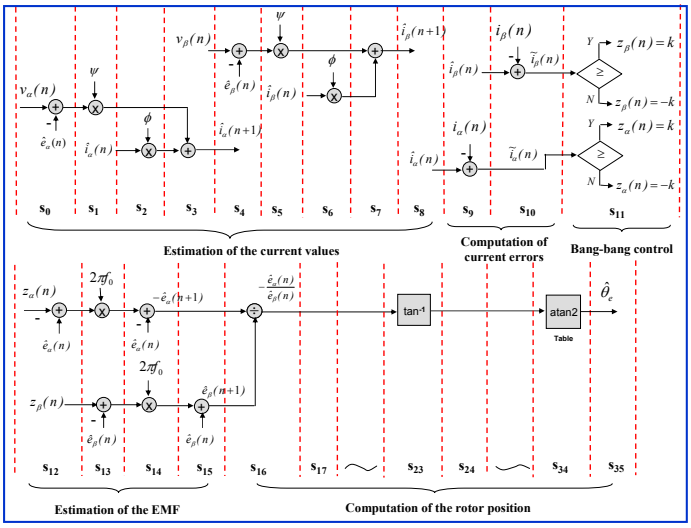

Fig.6 State diagram of an FSM for describing the SMO-based rotor position estimation algorithm

In the simulation of the rotor flux position estimation, sensor speed control is considered and the running speeds of PMSM with 250rpm, 500rpm, 1000rpm and 2000 rpm are tested. The simulation results for the actual rotor flux position θ_e, the estimated rotor flux position $\hat{\theta}_e$ are shown in Fig. 7. It presents that the estimated rotor flux position can follows the actual rotor flux position with some delay time. The delay time is about 600μs while PMSM running speed at 2000 rpm, but about 200μs at 250 rpm. After confirming the effectiveness of the rotor flux position estimation in the sensor speed control, we continue the simulation work in sensorless control architecture. The estimated rotor flux position will be feed-backed to the current loop for vector control and to the speed loop for speed control. The simulation work of the step speed response is tested. The motor speeds command is designed with step varying from 0rpm→500rpm→1000rpm→1500rpm→1000rpm, and the results for actual rotor speed, estimated rotor speed response and current response are shown in Fig.8. The rising time and steady-state value are about 210ms and near 0mm, which presents a good speed following response without overshoot occurred. However, the simulations shown in Figs. 7~8 demonstrate the effectiveness and correctness of the sensorless speed control IP for PMSM.

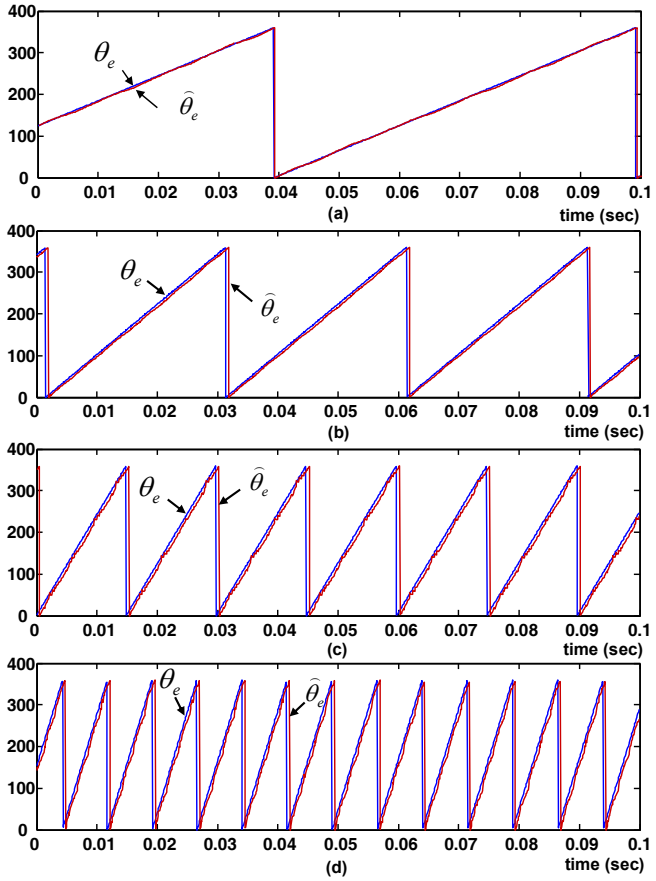

Fig. 7 Actual rotor flux angle (θ_e) and estimated rotor flux angle ($\hat{\theta}_e$) under PMSM speed running at (a)250rmp (b)500rpm (c)1000rpm (d)2000rpm

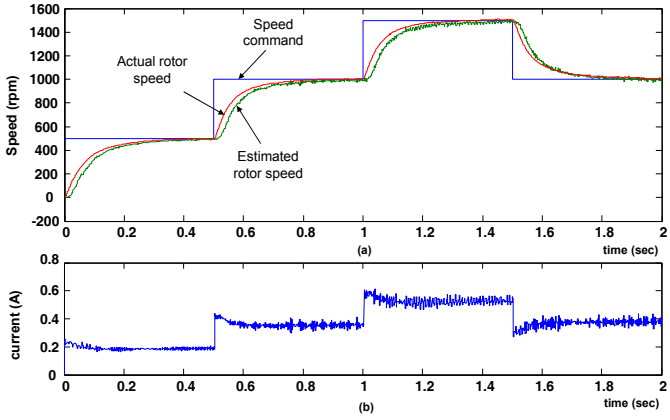

Fig. 8 (a) Step speed command, actual rotor speed and estimated rotor speed response (b) current response

V. CONCLUSIONS

This study has been presented a sensorless speed control IP for PMSM drive and successfully demonstrated its performance through co-simulation by using Simulink and ModelSim. After confirming the effective of VHDL code of sensorless speed control IP, we will realize this code in the experimental FPGA-based PMSM drive system for further verifying its function in the future work.

ACKNOWLEDGMENT

The financial support provided by Bureau of Energy is gratefully acknowledged.

REFERENCE

[1] V.C. Ilioudis and N.I. Margaris, "PMSM Sensorless Speed Estimation Based on Sliding Mode Observers," *in Proceedings of Power Electronics Specialists Conference (PESC)*, pp.2838~2843, 2008.

[2] W. Lu and Y. Hu and W. Huang and J. Chu and X. Du and J. Yang, "Sensorless Control Of Permanent Magnet Synchronous Machine Based on A Novel Sliding Mode Observer," *in Proceedings of Power Electronics and Applications Conference*, pp.1~4, 2009.

[3] M. Ezzat and J.d. Leon and N. Gonzalez and A. Glumineau, "Sensorless Speed Control of Permanent Magnet Synchronous Motor by using Sliding Mode Observer," *in Proceedings of 2010 11th International Workshop on Variable Structure Systems*, pp.227~232, June 26 - 28, 2010.

[4] S. Chi and Z. Zhang and L. Xu, "Sliding-Mode Sensorless Control of Direct-Drive PM Synchronous Motors for Washing Machine Applications," *IEEE Trans. on Indus. Applica.*, vol. 45, no. 2, pp.582~590,Mar./Apr. 2009.

[5] D. Jiang and Z. Zhao and F. Wang, "A Sliding Mode Observer for PMSM Speed and Rotor Position Considering Saliency," *in Proceedings of IEEE Power Electronics Specialists Conference* (PESC), pp.809~814, 2008.

[6] P. Borsje, and T.F Chan,. and Y.K. Wong, and S.L. Ho, "A Comparative Study of Kalman Filtering for Sensorless Control of a Permanent-Magnet Synchronous Motor Drive," *in Proceedings of IEEE International Conference on Electric Machines and Drives,* pp.815~822, 2005.

[7] H.A.F. Mohamed and S. S. Yang and M. Moghavvemi , "Sensorless Fuzzy SMC for a Permanent Magnet Synchronous Motor ," *in Proceedings of ICROS-SICE International Joint Conference* 2009, pp.619~623.

[8] Z.Zhou, T. Li, T. Takahahi and E. Ho, "FPGA realization of a high-performance servo controller for PMSM," *in Proceeding of the 9th IEEE Application Power Electronics conference and Exposition,* 2004, vol.3, pp. 1604-1609.

[9] Y.S. Kung and M.H. Tsai, "FPGA-based speed control IC for PMSM drive with adaptive fuzzy control," *IEEE Trans. on Power Electronics*, vol. 22, no. 6, pp. 2476-2486, Nov. 2007.

[10] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems – a review" *IEEE Trans. Ind. Electron.*, vol. 54, no.4, pp.1824-1842, Aug. 2007.

[11] M. F. Castoldi, G. R. C. Dias, M. L. Aguiar and V. O. Roda, "Chopper-Controlled PMDC motor drive using VHDL code," *in Proceedings of the 5th Southern Conference on Programmable Logic*, pp. 209~212, 2009.

[12] M. F. Castoldi and M. L. Aguiar, "Simulation of DTC strategy in VHDL code for induction motor control," *in Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE)*, pp.2248-2253, 2006.

[13] J. L´azaro, A. Astarloa, J. Arias, U. Bidarte and A. Zuloaga, "Simulink/Modelsim simulable VHDL PID core for industrial SoPC multiaxis controllers," *in Proceedings of the IEEE Industrial Electronics 32nd Annual Conference (IECON)*, pp.3007-3011, 2006.

[14] Y. Li , J. Huo, X. Li, J. Wen, Y. Wang and B. Shan, "An open-loop sin microstepping driver based on FPGA and the Co-simulation of Modelsim and Simulink," *in Proceedings of the International Conference on Computer, Mechatronics*, Control and Electronic Engineering (CMCE), pp. 223-227, 2010.

[15] The Mathworks, Matlab/Simulink Users Guide, *Application Program Interface Guide*, 2004

[16] Modeltech, ModelSim Reference Manual, 2004.