Simulink/ModelSim Co-Simulation of Sensorless PMSM Speed Controller

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

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

$$\frac{di_d}{dt} = -r_di_d + \omega_s L_qi_q + \frac{1}{L_d}v_d$$  \hspace{1cm} (1)

$$\frac{di_q}{dt} = -\omega_s L_d i_d - r_qi_q - \omega_s K_F + \frac{1}{L_q}v_q$$  \hspace{1cm} (2)

where $v_d$, $v_q$ are the $d$ and $q$ axis voltages; $i_d$, $i_q$ are the $d$ and $q$ axis currents, $r_i$ is the phase winding resistance; $L_d$, $L_q$ are the $d$ and $q$ axis inductance; $\omega_s$ is the rotating speed of magnet flux; $K_F$ is the permanent magnet flux linkage.
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,

$$T_e = \frac{3P}{4} K_e i_d A K_i n$$

(3)

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

$$J_{m} \frac{d}{dt} \varphi + B_m \varphi = T_e - T_l$$

(4)

where $T_e$ is the motor torque, $P$ is pole pairs, $K_i$ is torque constant, $J_m$ is the inertial value, $B_m$ is damping ratio, $T_l$ is the external torque, $\varphi$ 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 the following equation

$$\begin{bmatrix}
    v_d \\
    v_q
\end{bmatrix} = \begin{bmatrix}
    r_d + sL & -\omega_d L \\
    \omega_d L & r_q + sL
\end{bmatrix} \begin{bmatrix}
    i_d \\
    i_q
\end{bmatrix} + \begin{bmatrix}
    0 \\
    \omega_d K_i
\end{bmatrix}$$

(5)

where $L_{d} = L_{q}$. Transforming (5) of the circuit equation of PMSM on the $\alpha - \beta$ fixed coordinate can be derived by the following equation

$$\begin{bmatrix}
    v_\alpha \\
    v_\beta
\end{bmatrix} = \begin{bmatrix}
    r_\alpha + sL & -\omega_\alpha L \\
    \omega_\alpha L & r_\beta + sL
\end{bmatrix} \begin{bmatrix}
    i_\alpha \\
    i_\beta
\end{bmatrix} + \begin{bmatrix}
    0 \\
    \omega_\alpha K_i
\end{bmatrix} - \sin \theta_r$$

(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; $\theta_r$ is angular position at magnet flux; $s$ is differential operator. In addition, in (6), let’s define the EMF as

$$e = \begin{bmatrix}
    e_\alpha \\
    e_\beta
\end{bmatrix} A \omega_\alpha K_i \begin{bmatrix}
    -\sin \theta_r \\
    \cos \theta_r
\end{bmatrix}$$

(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,

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

(8)

where

$$A = \begin{bmatrix}
    -r_\alpha / L & 0 \\
    0 & -r_\beta / L
\end{bmatrix}$$

(9)

Thirdly, a sliding mode observer is designed by

$$\frac{d}{dt} \begin{bmatrix}
    i_\alpha \\
    i_\beta
\end{bmatrix} = \begin{bmatrix}
    i_\alpha \\
    i_\beta
\end{bmatrix} + \frac{1}{L} \begin{bmatrix}
    v_\alpha \\
    v_\beta
\end{bmatrix} - \frac{1}{L} \begin{bmatrix}
    e_\alpha + k \cdot \text{sign}(i_\alpha - \hat{i}_\alpha) \\
    e_\beta + k \cdot \text{sign}(i_\beta - \hat{i}_\beta)
\end{bmatrix}$$

(10)

where the $[i_\alpha, i_\beta]^T$ 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.

$$Z = \begin{bmatrix}
    z_\alpha \\
    z_\beta
\end{bmatrix} = \begin{bmatrix}
    z_\alpha \\
    z_\beta
\end{bmatrix} A k \cdot \text{sign}(i_\alpha - \hat{i}_\alpha)$$

(11)

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

$$e_{\text{error}}^T e_{\text{error}} < 0$$

(12)

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

$$\begin{bmatrix}
    z_\alpha \\
    z_\beta
\end{bmatrix} = \begin{bmatrix}
    e_\alpha \\
    e_\beta
\end{bmatrix} = \omega_\alpha K_i \begin{bmatrix}
    -\sin \theta_r \\
    \cos \theta_r
\end{bmatrix}$$

(13)

Fourthly, to alleviate the high frequency switching in bang-bang control, a low-pass filter is applied in Fig.2,
\[
\frac{d}{dt} [\hat{\varepsilon}_\theta] = -\omega_b [\hat{\varepsilon}_\alpha + \omega_b [\hat{\varepsilon}_\beta]]
\]

(14)

where \(\omega_b = 2\pi f_o\). Finally, the rotor position \(\hat{\theta}_r\) can be computed by

\[
\hat{\theta}_r = \tan^{-1}(-\frac{\hat{\varepsilon}_\beta}{\hat{\varepsilon}_\alpha})
\]

(15)

Rotor position estimation

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

\[
\begin{bmatrix}
\hat{\varepsilon}_\alpha(n+1) \\
\hat{\varepsilon}_\beta(n+1)
\end{bmatrix} = \begin{bmatrix}
\phi & 0 \\
0 & \phi
\end{bmatrix} \begin{bmatrix}
\hat{\varepsilon}_\alpha(n) \\
\hat{\varepsilon}_\beta(n)
\end{bmatrix} + \begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix} \begin{bmatrix}
\varepsilon(n) - \hat{\varepsilon}_\alpha(n) \\
\zeta(n) - \hat{\varepsilon}_\beta(n)
\end{bmatrix}
\]

(16)

where \(\phi = e^{-\frac{T_s}{2}}, \psi = e^{-\frac{T_s}{2}}\) and \(T_s\) is the sampling time. In addition, from (14), the difference equation of the EMF estimation can also be expressed by

\[
\begin{bmatrix}
\hat{\varepsilon}_\alpha(n+1) \\
\hat{\varepsilon}_\beta(n+1)
\end{bmatrix} = \begin{bmatrix}
\phi & 0 \\
0 & \phi
\end{bmatrix} \begin{bmatrix}
\hat{\varepsilon}_\alpha(n) \\
\hat{\varepsilon}_\beta(n)
\end{bmatrix} + 2\pi f_o \begin{bmatrix}
\zeta(n) - \hat{\varepsilon}_\alpha(n) \\
\zeta(n) - \hat{\varepsilon}_\beta(n)
\end{bmatrix}
\]

(17)

Once the EEMF is estimated, the estimated rotor position \(\hat{\theta}_r\) can be directly computed by

\[
\hat{\theta}_r(n) = \tan^{-1}(-\frac{\hat{\varepsilon}_\beta(n)}{\hat{\varepsilon}_\alpha(n)})
\]

(18)

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

Step 1: Estimate the estimated current by SMO in (16).
Step 2: Calculate the current error by \(\bar{j}_a(n) = \hat{j}_a(n) - j_a(n)\) and \(\bar{j}_r(n) = \hat{j}_r(n) - j_r(n)\)
Step 3: Obtain the Z gain of the current observer in (11)
Step 4: Estimate the EMF in (17).
Step 5: Obtain the estimated rotor position in (18)

\[\text{C. Fuzzy controller (FC)}\]

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

\[
e(n) = \omega^* - \omega(n)
\]

(19)

\[
de(n) = e(n) - e(n-1)
\]

(20)

and \(u_t\) represents the output of the fuzzy controller. The \(\omega^*\) is the speed command. The design procedure of the fuzzy controller is as follows:

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

(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 \(\mu_{A_j}(e)\) can be derived in which the error is located between \(e_1\) and \(e_{i+1}\), two linguistic values of \(A_j\) and \(A_{j+1}\) are excited, and the membership degree is obtained by

\[
\mu_{A_j}(e) = \frac{e_{i+1} - e}{2} \quad \text{and} \quad \mu_{A_{j+1}}(e) = 1 - \mu_{A_j}(e)
\]

(21)

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

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

\[
\text{IF } e \text{ is } A_j \text{ and } de \text{ is } B_j \text{ THEN } u_i \text{ is } c_{ji}
\]

(22)

where \(i = 0 - 6, A_i\) and \(B_j\) are fuzzy number, and \(c_{ji}\) is real number.

(d) Construct the fuzzy system \(u_i(e,de)\) 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 \(c_{ji}, c_{j+1,i}, c_{j+1,i+1}\) can be excited, and the (22) can be replaced by the following expression:

\[
u_f(e,de) = \frac{\varepsilon}{\mu_{A_j}(e) \cdot \mu_{B_j}(de)} \sum_{i=0}^{i+1} (i) \sum_{j, \mu_{B_j}(de)} (\varepsilon)^{\mu_{A_j}(e)} \cdot \mu_{A_i}(de) \text{ } d_{m,n}
\]

(23)

where \(d_{m,n} = \Delta \cdot \mu_{A_i}(de) \cdot \mu_{B_i}(de)\). And those \(c_{m,n}\) denote the value of the singleton fuzzifier.
III. SIMULINK/MODELSIM CO-SIMULATION OF SENSORLESS SPEED CONTROL 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 2 kHz, respectively. The clocks of 50 MHz and 12.5 MHz 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 \( s_0 \) to \( s_1 \) execute the speed estimation, \( s_2 \) to \( s_3 \) perform the computation of speed error and error change; steps \( s_4 \) to \( s_7 \) execute the function of the fuzzification; \( s_8 \) describe the look-up table and \( s_9 \) to \( s_{17} \) defuzzification; and steps \( s_{18} \) to \( s_{21} \) execute the computation of PI controller and command output. The \( SD \) is the section determination of \( e \) and \( de \) and the \( RS/L \) represents the right shift function with one bit. The operation of each step in Fig.5 is 80 ns (12.5 MHz) in FPGA; therefore total 22 steps only need 1.76 \( \mu \)s 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 \( s_{27} \) to \( s_{30} \) execute the estimation of current value; steps \( s_{31} \) to \( s_{30} \) compute the current error; \( s_{31} \) is the bang-bang control; \( s_{32} \) to \( s_{33} \) describe the computation of EMF and \( s_{34} \) to \( s_{35} \) perform the computation of the rotor position. The operation of each step in Fig.6 is 80 ns (12.5 MHz) in FPGA; therefore total 36 steps only need 2.88 \( \mu \)s 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 0 RAM bits, 2,085 LEs and 24,576 RAM bits; 1,151 LEs 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 1.3 \( \Omega \), stator inductance is 6.3 mH, inertia is \( J=0.000108 \) kg*m² and friction factor is \( F=0.0013 \) N*m*s.
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 $\theta_e$, the estimated rotor flux position $\hat{\theta}$ 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 600$\mu$s while PMSM running speed at 2000 rpm, but about 200$\mu$s 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 $\rightarrow$ 500rpm $\rightarrow$ 1000rpm $\rightarrow$ 1500rpm $\rightarrow$ 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.
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