# O 146. CONTROL OF PERMANENT MAGNET SYNCHRONOUS MOTOR

Abdulhasan Altaey[1], Ali Al-Omari[2]

[1]*Mustansiriyah University, College of Engineering, Baghdad, Iraq*
[2]*Mustansiriyah University, College of Engineering, Baghdad, Iraq*

*E-mail: aaltaey13@gmail.com, ali_jabbar@yahoo.co.uk*

**ABSTRACT:** The use of PMSM (permanent magnet synchronous motor) in the industry increased strongly because of the advantages such as: high efficiency and torque intensity. The ordinary speed control of these motors use encoders which has the effect of increasing the noise, complexity and price of the system. To solve this problem, sensorless speed control have come to industry. There are a lot of sensorless algorithms but the most efficient methods are MRAS and SMO where EEMF (extended electromotive force) should be determined to specify the position or speed. There are two ways to implement these algorithms, either DSP or FPGA. In sensorless speed control, a model is built for the motor, this motor should work in parallel with the real motor. Real states such as currents of the motor are measured, and estimated states of the model are calculated simultaneously. This process is concurrent, and since the parallel processing is one of the best features of the FPGA, then the FPGA is more preferable than DSP in this application. So if parallel process is exist then FPGA is more preferable otherthan DSP. This work gives a simple review about control strategies of PMSM.

*Keywords: FPGA, MRAS, PMSM, SVPWM*

## 1. INTRODUCTION

Today, the Interior Permanent Magnet Synchronous Motor (IPMSM) has become more and more valuable in many industrial applications because of many features such as high-power density, high torque to inertia ratio, high efficiency. In space vector control, the addition of three phase motor stator currents will create a current vector with certain magnitude and angle. By orienting this stator current vector MMF to be at 90° with respect to rotor flux, a maximum amount of torque for a given amount of stator current can be obtained. Hence it is necessary to know the rotor position information to be able to control the speed and torque of this motor. Measurement devices such as resolvers or encoders can be used to obtain rotor angle information but these devices have high prices, increase the wiring, complexity and add noise to the system. To solve these problems, sensorless algorithms can be applied. There are a lot of sensorless algorithms, the most famous are SMO, MRAS, EKF, Fuzzy which can be applied in medium and high-speed range. These algorithms cannot be applied for low speed range because they are depend on the EMF or EEMF which are have very small or zero values in zero and low speed range and hence have a small signal to noise ratio. As a result, only one algorithm exists for low speed range which depend on the saliency of the rotor that is injecting high frequency carrier method. To implement these sensorless algorithms, there are two methods. Either using DSP or FPGA, which one is the best depends on many factors: In sensorless speed control, a model is built for the motor, this motor should work in parallel with the real motor. Real states such as currents of the motor are measured, and estimated states of the model are calculated simultaneously. This process is concurrent, and since the parallel processing is one of the best features of the FPGA, then the FPGA is more preferable than DSP in this application. Another criterion is that in DSPs are instruction based while FPGAs are clock based so to implement any simple mathematic process in DSP three or four initiating instruction have to be processed, as a result an additional time will be lost because of these initiating instructions. All of these reasons make FPGAs are more suitable.

## 2. FPGA IN PMSM CONTROL

In (Tomoki and Asahi, 2013) a deadbeat control (Takao et al, 1990; Alireza, 2015) is used with sensor speed control of IPMSM using FPGA, but inspecting Figure 1 shows that the $I_q$ requires about 50 μs to follow $I_{qref}$. In electric vehicle applications, the switching frequency is limited to not more than 10 kHz (switching period is 100 μs).

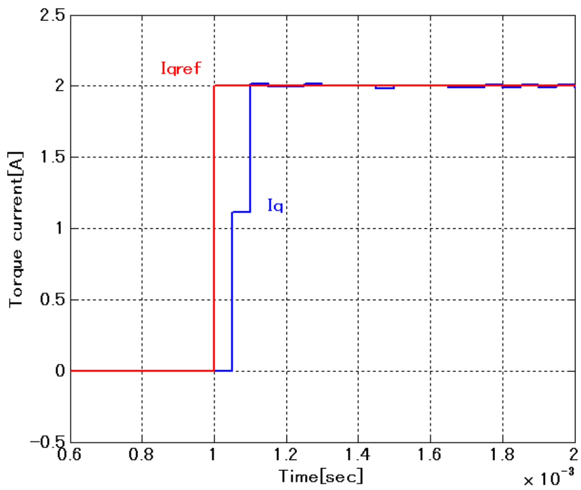


**Figure 2.** Simulation result for step response (Tomoki and Asahi, 2013)

An interesting module is proposed in (Ying-Shieh et al, 2005), in this work, the module consists of a fuzzy controller, a reference model for the PMSM and an adjusting mechanism (to minimize the error between the rotor speed and the output of the reference model). The adaptive fuzzy control in speed control loop is realized by software method under the Nios embedded processor and a current vector control is implemented using PLD. The dead time for the inverter is 1 µs, so the control loop should be finished in this time and the author didn't show the how much time the control loop has taken to track the real speed, however Figure 2 shows that the system tracks the real speed.
One of the basic mathematical tasks is to find the inverse Clarke i.e. converting α-β voltages and currents to a-b-c voltages and currents as shown in (1-3):

$$i_a = i_\alpha \quad (1)$$
$$i_b = -\frac{1}{2}i_\alpha + \frac{\sqrt{3}}{2}i_\beta \quad (2)$$
$$i_c = -\frac{1}{2}i_\alpha - \frac{\sqrt{3}}{2}i_\beta \quad (3)$$

(Ying-Shieh et al, 2005) used the following way to find $\frac{\sqrt{3}}{2}$:

$$\frac{\sqrt{3}}{2} = \frac{1}{2} + \frac{1}{4} + \frac{1}{16} + \frac{1}{32} + \frac{1}{64} + \frac{1}{256} + \frac{1}{512} + \frac{1}{2048} \quad (4)$$

Equation (4) is representing in FPGA as shown in Figure 3.
It is easier to represent $\frac{\sqrt{3}}{2}$ by equation (5):

$$\frac{\sqrt{3}}{2} = \left(\frac{1}{2} - \frac{1}{16} - \frac{1}{256} + \frac{1}{2048}\right)(2) \quad (5)$$

Equation (5) can be representated in VHDL code as shown below:

$i_{\beta 1}$ <='0' & $i_\beta$ (17 downto 1);
$i_{\beta 2}$ <="0000" & $i_\beta$ (17 downto 4);
$i_{\beta 3}$ <="00000000" &$i_\beta$ (17 downto 8);
$i_{\beta 4}$<="00000000000"& $i_\beta$ (17 downto 11);
$\frac{\sqrt{3}}{4}i_\beta$<=$i_{\beta 1}$-$i_{\beta 2}$ -$i_{\beta 3}$-$i_{\beta 4}$;
$\frac{\sqrt{3}}{2}i_\beta$<=$\frac{\sqrt{3}}{4}i_\beta$ (16 downto 0) & '0';
The last code if implemented will decrease the number of elements in Figure 3.

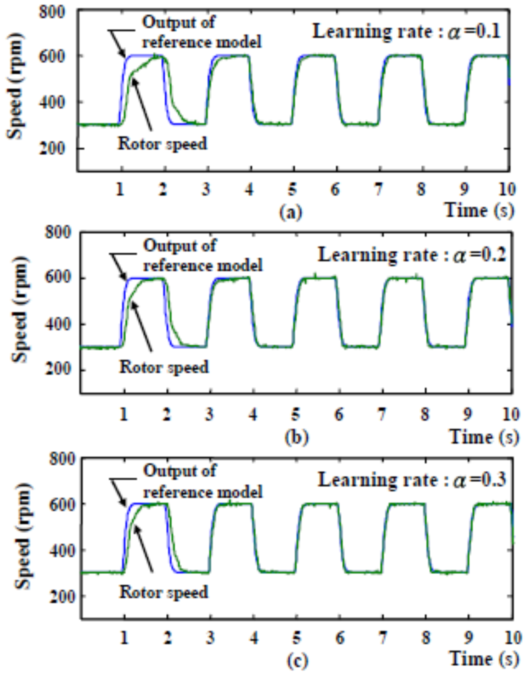


**Figure 2.** Tracking results between the output of reference model and the actual rotor speed under 1.0 Nm load torque but different learning rate (a) α=0.1 (b) α=0.2 (c) α=0.3 (Ying-Shieh et al, 2005).

For the low speed region, the back EMF of the motor is nearly zero and has a lot of noise, the algorithm used in this case is injecting a high frequency voltage in stator winding. The rotor saliency will affect the magnitude of the high frequency stator current and from which the rotor position can be determined. The work in (Maragliano et al, 2010) have designed a novel FPGA based algorithm aimed for the low speed range by injecting a high frequency signal in stator winding and shows a good results as shown in Figure (4-5) where the estimated speed track the measured speed in spite of that the reference speed is square wave which subject the system to hard conditions.

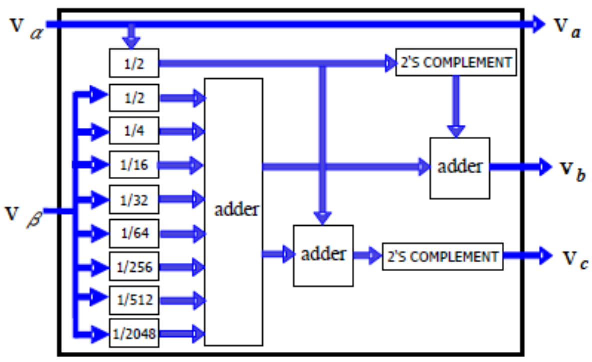


**Figure 3.** Inverse Clark Formulation (Ying-Shieh et al, 2005)

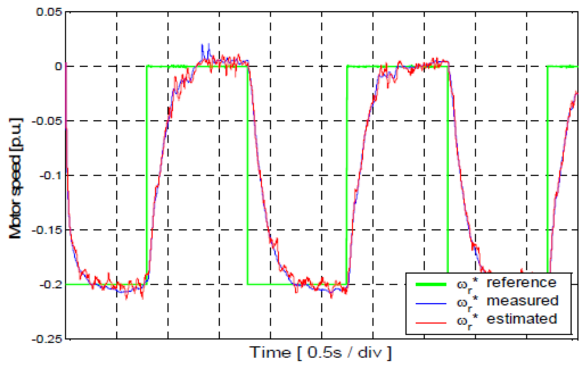


**Figure 4.** Reference, measured and estimated speed (Maragliano et al, 2010)

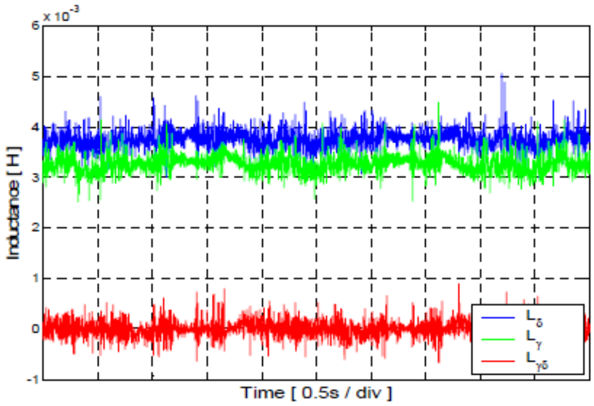


**Figure 5.** Estimated inductances (Maragliano et al, 2010)

Extended Kalman Filter (EKF), is one of the effective algorithms in speed estimation of PMSM but the problem of this algorithm is the high order state equation which lead to high calculation time. The work in (Nguyen et al, 2012) propose use two reduced order EKF work in parallel on FPGA platform as shown in Figure 6 and shows good tracking result as shown in Figure 7.

Two types of direct torque controller (DTC) for PMSM shown in Figure 8 have been controlled in (Yoshiharu et al, 2006). Both of theme used single chip FPGA, One of the controllers was constructed by programming a soft-core CPU and hardware logic circuits written in VHDL, while the other was constructed of only hardware. This work proved that the controller constructed of only hardware logic circuits was able to shorten the control period and it was able to suppress the low torque ripple. So constructing all control function by only hardware will decrease the processing time but at the expense on increasing of logic element which is not a problem in large scale IC. So in this work processing time is decreased from 50µs to 20.34µs. On the other hand, the rate of total logic elements increased from 35% to 53%.

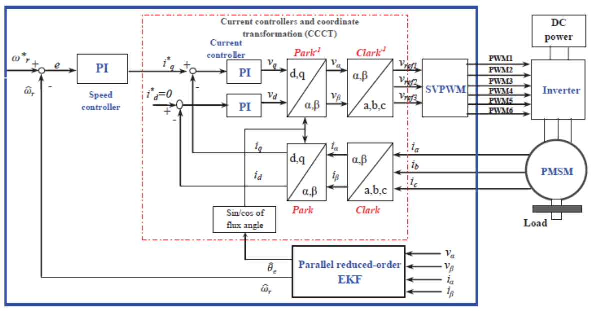


**Figure 6.** The proposed speed control system in (Nguyen et al, 2012)

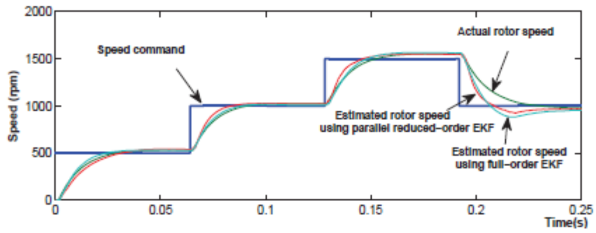


**Figure 7.** Speed response by using the full-order EKF and parallel reduced order EKF under speed varying from 0→500→1000→1500→1000 rpm. [(Nguyen et al, 2012)

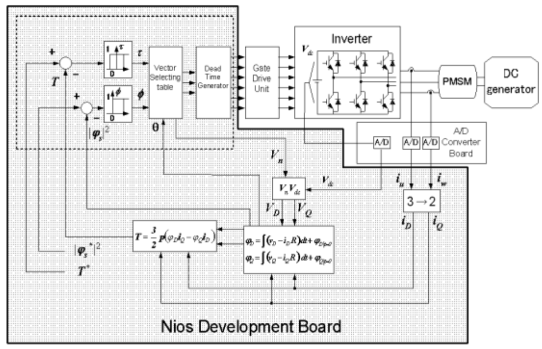


**Figure 8.** Block diagram of DTC for PMSM

## REFERENCES

Alireza, A., Davood, K., Mohammad, M. and Mojtaba, E., 2015 "Permanent Synchronous Motor Predictive Deadbeat Current Control-Robustness Investigation", The 6th International Power Electronics

Maragliano, G. , Marchesoni, M. and Vaccaro, L., 2010 "FPGA Implementation of a Sensorless PMSM Drive Control Algorithm Based on Algebraic Method", 978-1-4244-6392-3/10/$26.00 ©2010 IEEE.

Nguyen, Q., Nguyen, H., Hunter, G. and Ha, Q., 2012 "FPGA-based Sensorless PMSM Drive Using Parallel Reduced-Order Extended Kalman Filter", Control, Automation and Information Sciences (ICCAIS), 2012 International Conference on.

Takao, K., Takeshı, M. and Yushın, Y., 1990, " Dead Beat Control of Three Phase PWM Inverter", lEEE Transactlons On Power Electronics, Vol. 5. No. I , January 1990.

Tomoki, Y. and Asahi K., 2013, "Precise Torque Control for Interior Permanent Magnet Synchronous Motor using FPGA based Hardware Controller", 2013 IEEE GCC Conference and exhibition, November 17-20, Doha, Qatar.

Ying-Shieh, K., Chia-Sheng, C., Kiing-Ing, W. and Ming-Hung, T., 2005, "Development of a FPGA-based Control IC for PMSM Drive with Adaptive Fuzzy Control", 0-7803-9252-3/05/$20.00 ©2005 IEEE.

Yoshiharu, U., Nobukazu, H. and Kuniomi, O., 2006, "Comparison of FPGA-based Direct Torque Controllers for Permanent Magnet Synchronous Motors", Journal of Power Electronics, Vol. 6, No. 2, April 2006.