Design of FPGA-Based Speed Control of Permanent Magnet Synchronous Motor

Prasert Pinprathomrat¹ Napat Watjanatepin¹ Nitipat Eawsakul¹
Manoon Puangpool² Somchai Chatratana³

¹Rajamangala Institute of Technology nitipat_e@hotmail.com
²King Mongkut's Institute of Technology North Bangkok mpp@kmitnb.ac.th
³National Science and Technology Development Agency somchaich@nstda.or.th

ABSTRACT

The implementation of speed control of Permanent Magnet Synchronous Motor (PMSM) with Field Programmable Gate Array (FPGA) is proposed. Cascade control with inner loop as a current control and an outer loop as a speed control is chosen for simplicity and fast response. FPGA is a single chip (single processing unit), which will perform the following tasks: receive and convert control signal, create a reference current signal, control current and create switch signal and act as speed controller in a PI form. The 80 kHz sampling frequency and 25 bit of floating point data are defined in this implementation. The experimental results show that the performance of FPGA based speed control is comparable with the hardware based speed control, with the advantage of control algorithm flexibility.

Keywords: FPGA-Based, Speed Control, Permanent Magnet Synchronous Motor, PMSM

1. INTRODUCTION

The advantages of microprocessor-based control systems are mainly related to the programmable nature of microprocessors and their computing and communication capabilities. The ability to modify program strategy and configuration at software level provide high speed operation and adaptation flexibility. Owing to their computing capabilities, complex advanced control algorithms can be implemented. The communication capabilities of microprocessors make it possible to control the electric drives in a networked configuration where the coordination must be done by a central computer, or in an automated system containing several intelligent drives. The disadvantages and limitations of microprocessor based digital control systems are due mostly to the inherent properties of discrete systems that result from sampling nature, quantization, and truncation errors. These errors can seriously affect the load disturbance rejection limits of the controller. The computation delay limits the system bandwidth and too large delay can affect the stability of the system. The real-time performance must be taken into consideration while implementing the concurrent functions with time sharing methodologies.

Recent developments in Very-Large-Scale-Integration (VLSI) technology has radically affected the design process based on microprocessor. Nowadays, it is possible to build complex analog and digital integrated circuits from high performance Application-Specific-Integrated Circuit (ASIC) or Field Programmable Gate Array (FPGA). These technologies allow a fully digital hardware implementation in order to reach high-speed processor and optimal control performances. Unfortunately, an ASIC design is very complex and very expensive. Therefore it needs to be produced in large numbers to be profitable. The market of variable-speed drives for PMSM is not big enough to justify an ASIC manufacturing. This paper proposes an optimal and reused digital hardware integration for the design of FPGA-Based speed control of PMSM.

This approach is considered suitable for FPGA rapid prototyping. The function blocks, configured by the FPGA, can be treated as the concurrent and programmable hardware modules, gaining both efficiency and flexibility in system design. The program, designed by an end-user, will form logic gates within FPGA into a digital loop. FPGA works as a hardware format that allows the user to define the work procedure. FPGA provides more flexibility than microprocessor [1]-[3].

2. BACKGROUND THEORY OF PMSM

2.1 Model and Control Structure of the PMSM [4]

The structure of a PMSM-based drive system is shown in Fig. 1. The control inputs take values from the discrete set \(-u_0, +u_0\) instead of on-off signals from the discrete set \{0, 1\}. Let the six on-off signals be \(s_n = [s_{n1}, s_{n2}, s_{n3}, s_{n4}, s_{n5}, s_{n6}]^T\) with \(s_{n4} = 1 - s_{n4}, s_{n5} = 1 - s_{n5}, s_{n6} = 1 - s_{n6}\) and the current control inputs for sliding mode control design be \(U_{g_m} = [u_i, u_2, u_3]^T\), then the following relation holds:

\[
U_{g_m} = u_0 G_a s_n \text{ where } G_a = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}
\]
In general, the dynamic model of a PMSM can be established using physical laws as given by (2). Losses occur in a motor are negligible.

\[
di \frac{du}{dt} = \frac{u}{L} - \frac{R}{L} i - \frac{e}{L} \\
\frac{di}{dt} = \frac{u}{L} - \frac{R}{L} i - \frac{e}{L} \\
\frac{di}{dt} = \frac{u}{L} - \frac{R}{L} i - \frac{e}{L} \\
\frac{di}{dt} = \frac{u}{L} - \frac{R}{L} i - \frac{e}{L} \\
\frac{di}{dt} = \frac{u}{L} - \frac{R}{L} i - \frac{e}{L} \tag{2}
\]

where \( R_s \) is the phase winding resistance and \( L_s \) is the phase winding inductance; \( i_s, i_b, i_c \) are the phase currents and \( u_s, u_b, u_c \) the phase voltages. Furthermore, \( e_s, e_b, e_c \) are the phase induced EMF components of the following form (3)

\[
e_s = \frac{dV_s}{dt} = -\lambda_s \omega_s \sin \theta_s \\
e_b = \frac{dV_b}{dt} = -\lambda_b \omega_s \sin(\theta_s + 2\pi/3) \\
e_c = \frac{dV_c}{dt} = -\lambda_c \omega_s \sin(\theta_s - 2\pi/3) \tag{3}
\]

where \( \omega_s = d\theta_s/dt \) is the electrical angular speed of the motor rotor.

The relationship between the phase voltages \( u_s, u_b, u_c \) and the discontinuous controls \( u_s, u_b, u_c \) is given by (4) where matrix \( A_{abc}^{123} \) is defined as (5)

\[
\begin{bmatrix} u_s & u_b & u_c \end{bmatrix} = A_{abc}^{123} \begin{bmatrix} i_1 & i_2 & i_3 \end{bmatrix} \tag{4}
\]

\[
A_{abc}^{123} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \tag{5}
\]

Note that matrix \( A_{abc}^{123} \) is a singular matrix which implies that the phase voltages \( u_s, u_b, u_c \) are not independent. As will be shown later, the sum of \( u_s, u_b, u_c \) is equal to zero due to the physical configuration given in Fig. 1.

For speed control of a PMSM, a cascaded control structure is usually preferred, with an inner current control loop and an outer speed control loop as shown in Fig. 2.

2.2 Sliding Mode Current Control

The goal of the current control is to design a current controller to track the desired currents \( \mathbf{I}^* \) which are normally provided by an outer-loop speed controller. In the context of sliding mode current control, the controller has to determine the discontinuous controls \( u_s, u_b, u_c \) as well as the on-off signals of the switch. The on-off signals may also be called switching patterns and are defined as \( s_{abc} = i_{abc} - i_{abc} \).

The switching patterns are the control signals feeding to the gates of the power converters, e.g. a voltage source inverter as shown in Fig. 1. Switching patterns for each phase are

\[
s_1 = l_s - i_s \\
s_2 = l_s - i_s \\
s_3 = l_s - i_s \tag{6}
\]

and the discontinuous control \( \mathbf{u} = u, \text{sign}(s) \) are defined in the following form

\[
u_i = u, \text{sign}(s_i) \\
u_i = u, \text{sign}(s_i) \tag{7}
\]

where \( u \) denotes the supplied link voltage. To enforce the sliding mode, control gain \( u \) should be selected such that \( ss^T < 0 \). Now check this condition by evaluating \( ss^T \) and select \( u \) as shown in (8) to (12).

\[
s_1s_1' = s_1 \left( \frac{di}{dt} + \frac{e_s}{L_s} + \frac{R_s}{L_s} i_s - \frac{u_s}{L_s} \right) \\
s_2s_2' = s_2 \left( \frac{di}{dt} + \frac{e_b}{L_b} + \frac{R_b}{L_b} i_b - \frac{u_b}{L_b} \right) \\
s_3s_3' = s_3 \left( \frac{di}{dt} + \frac{e_c}{L_c} + \frac{R_c}{L_c} i_c - \frac{u_c}{L_c} \right) \tag{8}
\]

Replace \( u_{abc} \) with (4) and (7), thus \( s_{abc}u_{abc} \) in (9) can be expressed as (10). The results are 8 statuses of the
switching possibilities. However, switch symbol will not
be +1 or –1 at the same time. Therefore the possibility of
six-state switch will be as equation (10).

\[
(x_{12}\|u_{\text{abc}})_{\text{max}} = \frac{2}{3} u_0|\vec{s}_{125}|
\]

(10)

Then sliding mode condition can be enforced.

\[
u_0 > \frac{3}{2} \left[ \frac{\text{d}F_{\text{abc}}}{\text{d}t} + e_{\text{abc}} + R_i i_{\text{abc}} \right]
\]

(11)

With \( F_{\text{abc}} = L_s (\text{d}i_{\text{abc}}/\text{d}t) + e_{\text{abc}} + R_i i_{\text{abc}} \)

\[
u_0 > \frac{3}{2} \max \left( |F'_e|, |F'_o|, |F'_r| \right)
\]

(12)

2.3 Speed control [4][5]

For the design of speed controller in an outer loop, the
current control loop may be treated as an ideal current
source, i.e. it can track a given reference current
immediately and accurately. This assumption can be true
only for systems in which the electrical time constant is
much smaller than the mechanical time constant, or for
systems where the dynamic response of the speed control
is not critical. Any control design techniques, linear or
nonlinear, may be used for speed control loop. In many
industrial systems PI (proportional plus integral)
controllers are used with or without feed-forward
compensation, depending on the nature of the controlled
system and the performance requirements. This type of
controller is simple, but it may be sensitive to
disturbances in mechanical subsystem. Since load torque
\( (\tau_l) \) is unknown, it will be set as a disturbance in the
system.

The s-domain transfer function of analog PI
controller \( G_p(s) \) obtained from the mathematical
modeling can be given in the form of (13), where \( K_p \) and
\( K_i \) are the proportional and integral gains respectively.

\[
G_p(s) = K_p + \frac{K_i}{S}
\]

(13)

A transformation of \( s \) from the s-domain into \( f(z) \) in
the z-domain using Bilinear Transformation ( this
technique, also called the Tustin or trapezoidal
approximation) gives the relationship (14), where \( T \) is the
sampling period.

\[
s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}
\]

(14)

The z-domain transfer function of discrete PI
controller \( G_p(z) \) obtained from the mathematical
modeling can be given in the form of (15)

\[
G_p(z) = K_p + \frac{K_i T}{2} \frac{1 + z^{-1}}{1 - z^{-1}}
\]

(15)

3. IMPLEMENTATION [6-8]

In FPGA module, APEX DSP Development Board
(starter Version) of Altera Co., Ltd will be used as a
controller. The Quartus II Limited Edition development
software provides a comprehensive environment for
system-on-a-programmable-chip (SOPC) design. It
controls PMSM. The motor parameters used to verify the
design principle are \( P = 8 \) Pole, \( J_s = 0.002 \text{ kg m}^2 \), \( B_s = 0.0002 \text{ Nm s rad}^{-1} \), \( k_i = 0.1665 \text{ Nm A}^{-1} \), \( R_i = 0.1 \Omega \), \( L = 3 \text{ mH} \)
The supplied link voltage is \( u_0 = 40 \text{ V} \). The data were
being processed with the size of 25 bit of floating point
as suggested by Pavle Belanović,[9], working at sampling
frequency 80 kHz.

![Fig. 3: Block diagram of PMSM control with FPGA](image)

The block diagram of a speed control of a permanent
magnet synchronous motor is shown in Fig.3. The FPGA
is employed to perform 4 main functions as follows:

3.1. Receive and convert speed and current signals

3.1.1. The block diagram of FPGA and associated
circuits are shown in Fig. 4. The analog speed reference
and the analog current signals are fed to IC MAX 310 for
multiplexing and then converted to digital signal by an
analog to digital (A/D) converter. The FPGA will
perform the following tasks: 1) Control the multiplexing
and de-multiplexing of the input signals. 2) Create \( i_b \)
signal from the currents flow in the three-phase motor
windings which are star-connected without neutral. The
expression is \( i_b = -(i_c + i_a) \). 3) Convert 10 bit fixed point
speed data into 25 bit floating point for calculation
purpose in next step. These functions require about 3,000
logic gates.

![Fig. 4: Receiving reference speed signal and motor
current with FPGA](image)
3.1.2. Fig. 5. shows the block diagram of FPGA for actual speed calculation. Three digital signals from encoder are pulse A, pulse B and pulse Z signal. A phase angle difference between pulse A and pulse B is 90 degree. It will set a speed and direction of motor rotation. Pulse Z signal will set a starting point of rotor. This function requires about 1,200 logic gates.

3.2. Create current reference signals

Three current reference signals in form of alternating current format can be generated by FPGA as shown in Fig. 6. The magnitude of reference current signal from PI controller generating from section 3.4 was multiplied by three phase sinusoidal waveform. The sinusoidal waveform was generated by the translation from sin table into sin value. In order to get a synchronizing relationship between current wave and rotor movement, it requires about 12,000 logic gates.

3.3. Control current and create switch signal

The block diagram for switch signal generation and sliding mode current control are shown in Fig. 7. Equation (7) was programmed on FPGA by comparing reference current signal (section 3.2) to measured current signal (section 3.1.1). The outputs of the comparators are switch signals of inverter in Fig. 1. D-flip-flop will set a sampling frequency of switching at 80 kHz. This function requires 8,000 logic gates.

3.4. PI Controller

PI controller block diagram was written in the form of PI controlling equation (15) on FPGA using 25 bit of floating point format for processing (Fig. 8). It requires about 80,000 logic gates.

4. EXPERIMENTAL RESULTS

Fig. 9. shows the current waveforms measured from an experiment with 250 Watts PMSM. The current control was in sliding mode. The $i_a$ (actual current) in CH2 is in good agreement with $i_a^*$ (current reference) at CH1. The scale for current is 2 A/DIV and the time scale is 25 ms/div.

Fig. 10: Current Controlling Signal in Sliding Mode Control.

The performance of speed control is shown in Fig. 10 and Fig. 11. The scale for all speeds is 500 rpm/div (a). speed reference. b). actual speed. ) and for all currents is 2 A/div (c). current reference of DC signal. d). current reference of AC signal. e). actual current. ), with the time scale of 500 ms/div. It can be seen that the control system responded well with the change of controller gain and time constant.
5. CONCLUSION

FPGA has been used for controlling PMSM. FPGA is set in the form of IC single chip at sampling frequency 80 kHz. It was programmed to control the motor current in sliding mode and to control the speed of the motor by processing the floating point number. The user can design the work procedure and define the value of the parameters without changing there structure such as fixed point of microprocessor. Experimental results showed that FPGA based speed control performance in real time is comparable to the hardware based speed control in terms of response time and accuracy, with the advantage of flexibility in control scheme implementation.

6. REFERENCES


Fig. 10: PMSM speed control using $K_p = 0.0002441$ and $K_i = 0.0023846$

Fig. 11: PMSM speed control using $K_p = 0.0004882$ and $K_i = 0.0047683$