We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Realization of a Control IC for PMLSM Drive Based on FPGA Technology

Ying-Shieh Kung[1] and Chung-Chun Huang[2]
[1]Department of Electrical Engineering, Southern Taiwan University
[2]Green Energy and Environment Research Laboratory,
Industrial Technology Research Institute
Taiwan

1. Introduction

The advantages of superior power density, high-performance motion control with fast speed and better accuracy, are such that PMLSM (Permanent Magnet Linear Synchronous Motor) are being increasingly used in many automation control fields as actuators (McLean, 1988; Gieras & Piech, 2000; Budig, 2000;), including computer-controlled machining tools, X-Y driving devices, robots, semiconductor manufacturing equipment, etc. However, the PMLSM does not use conventional gears or ball screws, so the payload upon the mover greatly affects the positioning performance (Liu et al., 2004). To cope with this problem, many advanced control techniques (Qingding et al., 2002; Lin et al., 2007; Wai & Chu, 2007), such as fuzzy control, neural networks control and robust control have been developed and applied to the position control of the PMLSM drive to obtain high operating performance. However, the execution of a neural network or fuzzy controller requires many computations, so implementing these highly complex control algorithms depend on the PC systems in most studies before (Qingding et al., 2002; Liu et al., 2004). In recent years, the fixed-point DSP (Digital Signal Processor) and the FPGA (Field Programmable Gate Array) provide a possible solution in this issue (Lin et al., 2005; Kung, 2008). Comparing with FPGA, although the intelligent control technique using DSP provides a flexible skill, it suffers from a long period of development and exhausts many resources of the CPU.

Nowadays, the FPGA has brought more attention before. The advantages of the FPGA includes 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 (Cho, et al., 2009; Monmasson & Cirstea, 2007; Naouar et al., 2007; Kung & Tsai, 2007; Jung & Kim, 2007; Huang & Tsai, 2009; Kung, et al., 2009). FPGA provides a compromise between the special-purpose ASIC (application specified integrated circuit) hardware and general-purpose processors (Wei et al., 2005). Recently, Li et al. (2003) utilized an FPGA to implement autonomous fuzzy behavior control on mobile robot. Lin et al., (2005) presented a fuzzy sliding-mode control for a linear induction motor drive based on FPGA. But, due to the fuzzy inference mechanism module adopts parallel processing circuits, it consumes much more FPGA resources; therefore limited fuzzy rules are used in their proposed method.

To solve the aforementioned problem, this work firstly proposed to use an FPGA and an embedded NiosII processor to develop a control IC for PMLSM. The control IC has two modules. One module performs the functions of the PTP motion trajectory for PMLSM. The

other module performs the functions of position/speed/current controllers of PMLSM drive. The former is implemented by software using Nios II embedded processor and the latter included by an AFC circuit, QEP (Quadrature Encoder Pulse) capture circuits, SVPWM (Space Vector Pulse Width Modulation) circuits, ADC interface circuit and current vector control circuit are implemented by hardware in FPGA. Secondly, to reduce the usage of FPGA resource, an FSM (Finite-State Machine) joined by a multiplier, an adder, a lookup table, and some comparators and registers is presented to model the circuits of AFC, SVPWM and current vector control. And VHDL (VHSIC hardware description language) is adopted to describe the circuits of the FSM (Hsu et al., 1996). Therefore, the hardware/software co-design technology can make the controller of PMLSM more compact, flexible, better performance and less cost. The FPGA chip employed herein is an Altera Stratix II EP2S60F672C5 (Altera, 2008) with 48,352 ALUTs, maximum 492 user I/O pins, 36 DSP blocks, 2,544,192 bits of RAM, and a Nios II embedded processor. Finally, an experimental system including an FPGA experimental board, an inverter and a PMLSM, is set up to verify the correctness and effectiveness of the proposed method.

The core idea and some results of this work have been published in the IEEE ICIT2006 conference (Kung, et al., 2006). This work exposes more clearly technical description in control IC design of PMLSM. The organization of this chapter is that the first section is the introduction, the second session describes the mathematical model of PMLSM, current vector control and position AFC design, the third session presents the control IC design, the fourth session shows the experimental system and results, and the final session is the conclusion.
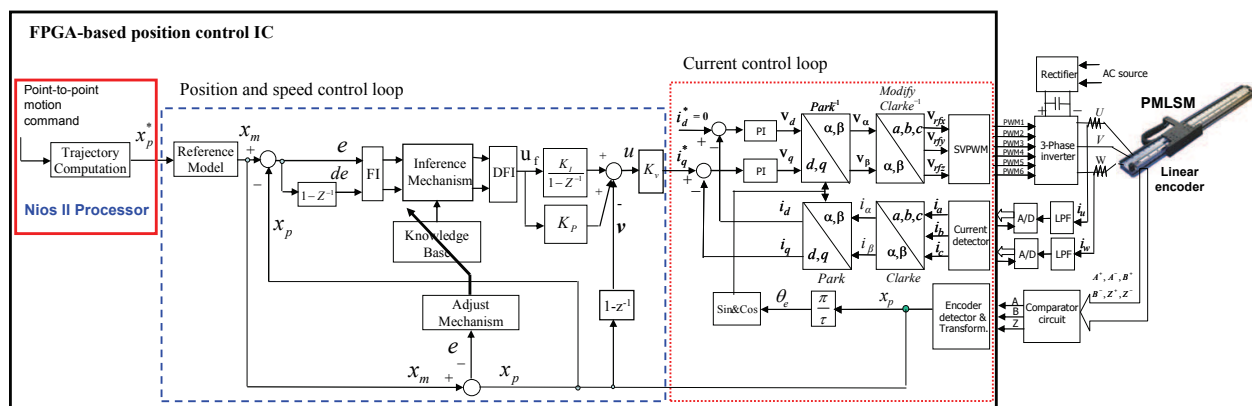

Fig. 1. The architecture of the FPGA-based position control IC for PMLSM drive system

2. System description of PMLSM drive and AFC design

The internal architecture of the proposed FPGA-based controller system for a PMLSM drive is shown in Fig. 1. A PTP motion trajectory, an AFC in the position loop, a P controller in the speed loop and a current vector control scheme for PMLSM are all realized in one FPGA.

2.1 Mathematical model of the PMLSM drive

The dynamic model of a typical PMLSM can be described in the synchronous rotating reference frame, as follows

$$\frac{di_d}{dt} = -\frac{R_s}{L_d}i_d + \frac{\pi}{\tau}\frac{L_q}{L_d}\dot{x}_p i_q + \frac{1}{L_d}\mathrm{v}_d \tag{1}$$

$$\frac{di_q}{dt} = -\frac{\pi}{\tau}\frac{L_d}{L_q}\dot{x}_p i_d - \frac{R_s}{L_q}i_q - \frac{\pi}{\tau}\frac{\lambda_f}{L_q}\dot{x}_p + \frac{1}{L_q}v_q \tag{2}$$

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; $\dot{x}_p$ is the translator speed; λ_f is the permanent magnet flux linkage; τ is the pole pitch. The developed electromagnetic thrust force is given by

$$F_e = \frac{3\pi}{2\tau}((L_d - L_q)i_d + \lambda_f)i_q \tag{3}$$

The current control of a PMLSM drive is based on a vector control approach. That is, if we control i_d to 0 in Fig.1, the PMLSM will be decoupled, so that control a PMLSM will become easy as to control a DC linear motor. After simplification and considering the mechanical load, the model of a PMLSM can be written as the following equations,

$$F_e = \frac{3}{2}\frac{\pi}{\tau}\lambda_f i_q \overset{\Delta}{=} K_t i_q \tag{4}$$

with

$$K_t = \frac{3}{2}\frac{\pi}{\tau}\lambda_f \tag{5}$$

and the mechanical dynamic equation of PMLSM is

$$F_e - F_L = M_m \frac{d^2 x_p}{dt^2} + B_m \frac{dx_p}{dt} \tag{6}$$

where F_e, K_t, M_m, B_m and F_L represent the motor thrust force, the force constant, the total mass of the moving element, the viscous friction coefficient and the external force, respectively.

The current loop of the PMLSM drive in Fig.1 includes PI controller, coordinate transformations of Clark, Modified inverse Clark, Park, inverse Park, SVPWM (Space Vector Pulse Width Muldulation), pulse signal detection of the encoder etc. The coordination transformation of the PMLSM in Fig. 1 can be described in synchronous rotating reference frame. Figure 2 is the coordination system in rotating motor which includes stationary a-b-c frame, stationary α-β frame and synchronously rotating d-q frame. Further, the formulations among three coordination systems are presented as follows.

1.   *Clarke*: stationary a-b-c frame to stationary α-β frame.

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{7}$$
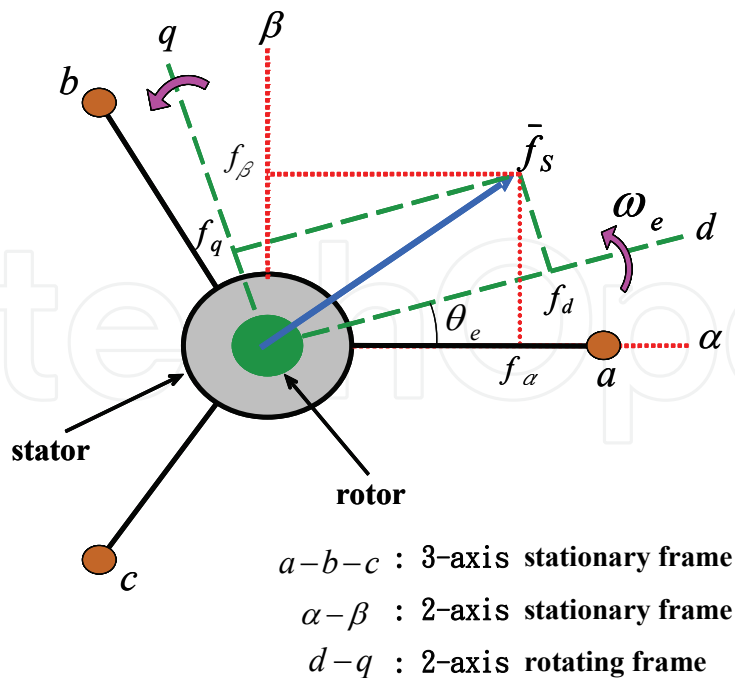
Fig. 2. Transformation between stationary axes and rotating axes

2.   Modified *Clarke*[-1]: stationary α-β frame to stationary *a-b-c* frame.

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{-1}{2} & \frac{\sqrt{3}}{2} \\ \frac{-1}{2} & \frac{-\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} v_\beta \\ v_\alpha \end{bmatrix} \tag{8}$$

3.   *Park*: stationary α-β frame to rotating *d-q* frame.

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos\theta_e & \sin\theta_e \\ -\sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{9}$$

4.   *Park*[-1]: rotating *d-q* frame to stationary α-β frame.

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} \cos\theta_e & -\sin\theta_e \\ \sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} v_d \\ v_q \end{bmatrix} \tag{10}$$

where θ_e is the electrical angle.

In Fig. 1, two digital *PI* controllers are presented in the current loop of PMSM. For the example in *d* frame, the formulation is shown as follows.

$$e_d(k) = i_d^*(k) - i_d(k) \tag{11}$$

$$v_{p\_d}(k) = k_{p\_d}\, e_d(k) \tag{12}$$

$$v_{i\_d}(k) = v_{i\_d}(k-1) + k_{i\_d}\, e_d(k-1) \tag{13}$$

$$v_d(k) = v_{p\_d}(k) + v_{i\_d}(k) \qquad (14)$$

the e_d is the error between current command and measured current. The k_{p_d}, k_{i_d} are P controller gain and I controller gain, respectively. The $v_{p\_d}(k)$, $v_{i\_d}(k)$, $v_d(k)$ are the output of P controller only, I controller only and the PI controller, respectively. Similarity, the formulation of PI controller in q frame is the same.

2.2 Design scheme of Space Vector Pulse Width Modulation (SVPWM)

SVPWM is a special switching scheme of a 3-phase power converter with the six power transistors. The typical structure of 3-phase power converter is shown in Fig. 3. According to the ON/OFF switching of upper transistors in Fig. 3, there have eight possible combinations. The eight vectors are called basic space vectors and they are denoted by U_0, U_{60}, U_{120}, U_{180}, U_{240}, U_{300}, O_{000} and O_{111}, which are shown in Fig. 4. Using the *Clarke* transformation, the project values in α-β axis for six basic space vectors can be obtained and are also shown in Fig. 4. The SVPWM technique is applied to approximate the reference voltage U_{out}, and it combines of the switching pattern with the basic space vectors. Therefore, the motor-voltage vector U_{out} will be located at one of the six sectors (S3, S1, S5, S4, S6, S2) at any given time. Thus, for any PWM period, it can be approximated by the vector sum of two vector components lying on the two adjacent basic vectors, as the following:

$$U_{out} = \frac{T_1}{T}U_X + \frac{T_2}{T}U_{X+60} + \frac{T_0(O_{000} \ or \ O_{111})}{T}, \qquad (15)$$

where $T_0 = T - T_1 - T_2$ and T is half of PWM carrier period. The detailed design scheme is described as following steps:
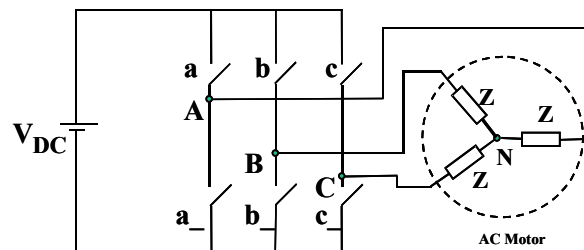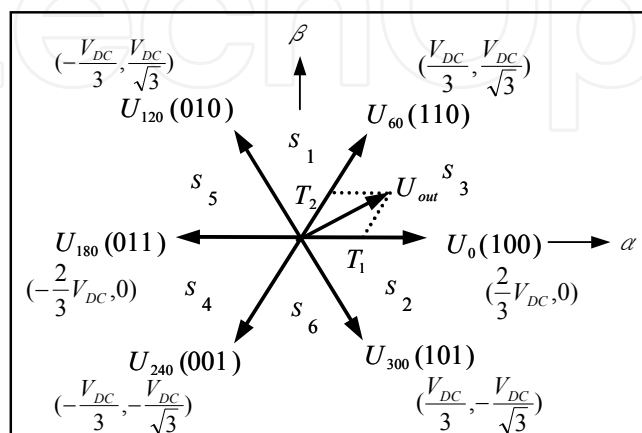

Fig. 3. Typical 3-phase power converter and AC motor

Fig. 4. Basic vector space and switching patterns

1. *Calculation of T_1 and T_2:* Any output voltage can be projected into each adjacent basic vector in SVPWM strategy. For example, the output voltage vector U_{out} in the sector S_3 can be the combination of U_0 and U_{60} shown in Fig. 4. Therefore, the calculation of T_1 and T_2 can be shown as,

$$U_0 = \frac{2}{3}V_{DC}\vec{\alpha} \tag{16}$$

$$U_{60} = \frac{1}{3}V_{DC}\vec{\alpha} + \frac{1}{\sqrt{3}}V_{DC}\vec{\beta} \tag{17}$$

If we substitute (16)~(17) into (15), we obtain

$$U_{out} = \frac{T_1}{T}U_0 + \frac{T_2}{T}U_{60} = \frac{T_1}{T}(\frac{2}{3}V_{DC}\vec{\alpha}) + \frac{T_2}{T}(\frac{V_{DC}}{3}\vec{\alpha} + \frac{V_{DC}}{\sqrt{3}}\vec{\beta})$$
$$\underline{\underline{\Delta}} \; V_\alpha\vec{\alpha} + V_\beta\vec{\beta} \tag{18}$$

and compare the coefficient in (18), thus

$$T_1 = \frac{T}{2V_{DC}}(3V_\alpha - \sqrt{3}V_\beta) \tag{19}$$

$$T_2 = \sqrt{3}\frac{T}{V_{DC}}V_\beta \tag{20}$$

In the similar way, the T_1 and T_2 in other sector can be derived and be rearranged in Table 1, which T_X, T_Y and T_Z are represented as the followings:

$$T_X = \sqrt{3}\frac{T}{V_{DC}}V_\beta \tag{21}$$

$$T_Y = \frac{T}{2V_{DC}}(3V_\alpha + \sqrt{3}V_\beta) \tag{22}$$

$$T_Z = \frac{T}{2V_{DC}}(-3V_\alpha + \sqrt{3}V_\beta) \tag{23}$$

If it is at the saturation condition $T_1 + T_2 > T$, the T_1 and T_2 should be modified as:

$$T_{1SAT} = T_1\frac{T}{T_1 + T_2} \tag{24}$$

$$T_{2SAT} = T_2\frac{T}{T_1 + T_2} \tag{25}$$

2. *Determination of the duty cycles and CMPx:* After the calculation of T_1 and T_2, it has to re-transfer it to the duty cycles and *CMPx* values to generate the PWM output signals for controlling the power transistor switching time in Fig. 3. First, the duty cycles are defined as Ta_{on}, Tb_{on} and Tc_{on} which are calculated as the follows:

$$Ta_{\mathrm{on}} = (T - T_1 - T_2)/2 = T_0/2 \tag{26}$$

$$Tb_{\mathrm{on}} = Ta_{\mathrm{on}} + T_1 \tag{27}$$

$$Tc_{\mathrm{on}} = Tb_{\mathrm{on}} + T_2 \tag{28}$$

Then, the *CMP1~CMP3* values can be obtained in Table 2, depend on the sector number. For example in S3 sector, its output waveforms PWM1~PWM3 are depicted in Fig. 5 with the duty time at U_0 (100), U_{60} (110) and zero vector (O_0 and O_{111}) be T_1, T_2, T_0, respectively.

3. *Determination of the sector:* To determine the sector, we first modified the *Clarke^{-1}* transformation as follows,

$$\begin{bmatrix} V_{rfx} \\ V_{rfy} \\ V_{rfz} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{-1}{2} & \frac{\sqrt{3}}{2} \\ \frac{-1}{2} & \frac{-\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_\beta \\ V_\alpha \end{bmatrix} \tag{29}$$

then, the output waveforms of V_{rfx}, V_{rfy} and V_{rfz} for sinusoid wave inputs (V_α, V_β) can be calculated and shown in Fig. 6. They can determine the sector according to the following rules:

$$\begin{aligned} &\text{If } V_{rfx} > 0 \text{ then a} = 1 \text{ else a} = 0 \\ &\text{If } V_{rfy} > 0 \text{ then b} = 1 \text{ else b} = 0 \\ &\text{If } V_{rfz} > 0 \text{ then c} = 1 \text{ else c} = 0 \\ &\text{Sector } = a + 2b + 4c \end{aligned} \tag{30}$$

From equations (21)~(23) and (29), we have.

$$\begin{bmatrix} T_x \\ T_y \\ T_Z \end{bmatrix} = \frac{\sqrt{3}T}{V_{DC}} \begin{bmatrix} V_{rfx} \\ -V_{rfz} \\ -V_{rfy} \end{bmatrix}. \tag{31}$$

T_x, T_y and T_z can be derived directly from V_{rfx}, V_{rfy} and V_{rfz}.

	S_3	S_1	S_5	S_4	S_6	S_2
T_1	$-T_Z$	T_Z	T_X	$-T_X$	$-T_Y$	T_Y
T_2	T_X	T_Y	$-T_Y$	T_Z	$-T_Z$	$-T_X$

Table 1. T_1 and T_2 in all specific sectors

Sector	S_3	S_1	S_5	S_4	S_6	S_2
CMP1	Ta_{on}	Tb_{on}	Tc_{on}	Tc_{on}	Tb_{on}	Ta_{on}
CMP2	Tb_{on}	Ta_{on}	Ta_{on}	Tb_{on}	Tc_{on}	Tc_{on}
CMP3	Tc_{on}	Tc_{on}	Tb_{on}	Ta_{on}	Ta_{on}	Tb_{on}

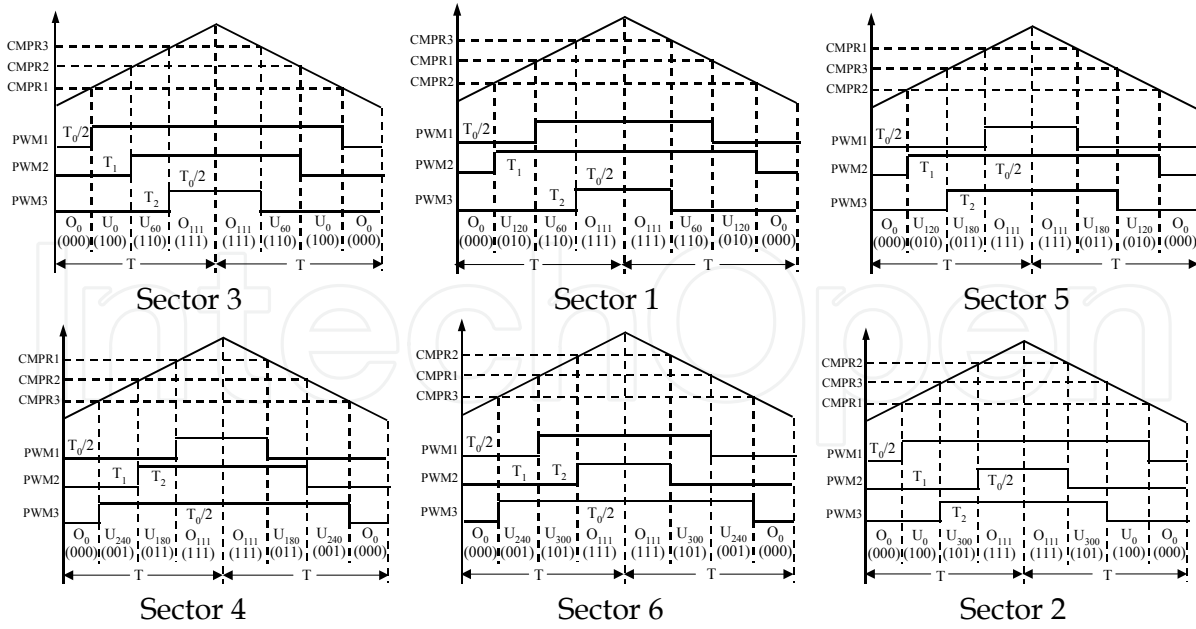Table 2. Assigning duty cycle to *CMPx* in any sector

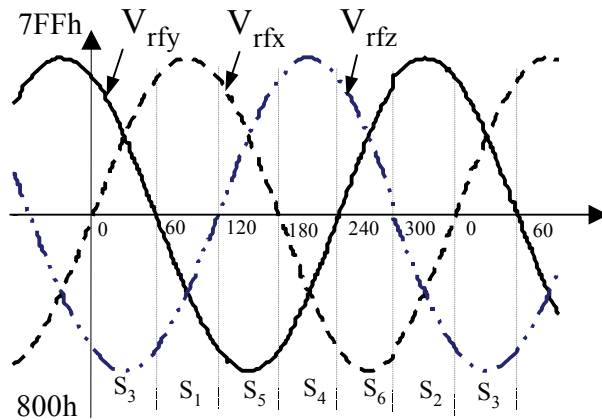Fig. 5. PWM patterns and duty cycle operating at all sectors

Fig. 6. 3-phase sinusoid output waveform

4. *Computation procedures of SVPWM design:* SVPWM design method is summary as following procedures.

Step 1. Determination of the sector according to the rule shown in (30), where V_{rfx}, V_{rfy}, V_{rfz} are the input signals of the SVPWM block circuit in Fig.1.

Step 2. Calculation of T_X, T_Y and T_Z from (31).

Step 3. Determination of T_1 and T_2 from Table 1. If it is at the saturation condition, we can use (24) and (25) to modify the T_1 and T_2.

Step 4. Determination of the duty cycle Ta_{on}, Tb_{on}, Tc_{on} from (26)~(28).

Step 5. Assignment of the duty cycles to *CMP1*, *CMP2* and *CMP3* from Table 2.

2.3 Adaptive fuzzy controller (AFC) in position control loop

The blue dash rectangular area in Fig. 1 presents the architecture of an AFC for the PMLSM drive. It consists of a fuzzy controller, a reference model and a parameter adjusting mechanism. Detailed description of these is as follows.

1. *Fuzzy controller (FC):*

In Fig.1, the tracking error and the change of the error, *e*, *de* are defined as

$$e(k) = x_m(k) - x_p(k) \tag{32}$$

$$de(k) = e(k) - e(k-1) \tag{33}$$

and *e*, *de* and u_f are input and output variables of FC, respectively. The fuzzifier gains K_e, K_{de} and defuzzifier gain K_u are used in the normalization and denormalization condition. The design procedure of the FC is as follows:

a. Take the *e* and *de* as the input variables of the FC, and define their linguist variables as *E* and *dE*. The linguist value of *E* and *dE* are {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* is based on the symmetrical triangular membership function which is shown in Fig.7. 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 $f(\xi_2) = 1$. With respect to the universe of discourse of [-6.6], the numbers for these linguistic values are selected as follows:

$$A_0 = B_0 : \{-6,-6,-4\}, A_1 = B_1 : \{-6,-4,-2\}, A_2 = B_2 : \{-4,-2,0\}, A_3 =$$
$$= B_3 : \{-2,0,2\}, A_4 = B_4 : \{0,2,4\}, A_5 = B_5 : \{2,4,6\}, A_6 = B_6 : \{4,6,6\} \tag{34}$$

b. Compute the membership degree of the *e* and *de*. Figure 7 shows that the only two linguistic values are excited (resulting in a non-zero membership) in any input value, and the membership degree is obtained by

$$\mu_{A_i}(e) = \frac{e_{i+1} - e}{2} \quad \text{and} \quad \mu_{A_{i+1}}(e) = 1 - \mu_{A_i}(e) \tag{35}$$

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

c. Select the initial fuzzy control rules, such as,

$$\text{IF } e \text{ is } A_i \text{ and } \Delta e \text{ is } B_j \text{ THEN } u_f \text{ is } c_{j,i} \tag{36}$$

where *i and j* = 0~6, A_i and B_j are fuzzy number, and $c_{j,i}$ is a real number. The graph of the fuzzy rule table and the fuzzification are shown in Fig. 7.

d. Construct the output of the fuzzy system $u_f(e,de)$ by using the singleton fuzzifier, product-inference rule, and central average defuzzifier method. Although there are total 49 fuzzy rules in Fig. 7 will be inferred, actually only 4 fuzzy rules can be effectively excited to generate a non-zero output. Therefore, if an error *e* is located between e_i and e_{i+1}, and an 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 output of the fuzzy system can be inferred by the following expression:

$$u_f(e,de) = \frac{\sum_{n=i}^{i+1}\sum_{m=j}^{j+1} c_{m,n}[\mu_{A_n}(e) * \mu_{B_m}(de)]}{\sum_{n=i}^{i+1}\sum_{m=j}^{j+1} \mu_{A_n}(e) * \mu_{B_m}(de)} \triangleq \sum_{n=i}^{i+1}\sum_{m=j}^{j+1} c_{m,n} * d_{n,m} \tag{37}$$

where $d_{n,m} \triangleq \mu_{A_n}(e) * \mu_{B_m}(de)$. And those $c_{i,j}$ are adjustable parameters. In addition, by using

(35), it is straightforward to obtain $\sum_{n=i}^{i+1}\sum_{m=j}^{j+1} d_{n,m} = 1$ in (37).

2.   *Reference model (RM)*:

Second order system is usually as the RM in the adaptive control system. Therefore, the transfer function of the RM in Fig.1 can be expressed as

$$\frac{x_m(s)}{x_p^*(s)} = \frac{\omega_n^2}{s^2 + 2\varsigma\omega_n s + \omega_n^2} \tag{38}$$

where ω_n is natural frequency and ς is damping ratio. Furthermore, because the characteristics of no overshoot, fast response and zero steady-state error are the important factors in the design of a PMLSM servo system; therefore, it can be considered as the selective criterion of ω_n and ς. The design methodology is described as follows: Firstly, the (38) matches the requirement of a zero steady-state error condition. Secondly, if we choose $\varsigma \geq 1$, it can guarantee no overshoot condition. Especially, the critical damp value $\varsigma = 1$ has a fastest step response. Hence, the relation between the rising time t_r and the natural frequency ω_n for a step input response in (38) can be derived and shown as follows.

$$(1 + \omega_n t_r)e^{-\omega_n t_r} = 0.1 \tag{39}$$

Once the t_r is chosen, the natural frequency ω_n can be obtained. Furthermore, applying the bilinear transformation, (38) can be transformed to a discrete model by

$$\frac{x_m(z^{-1})}{x_p^*(z^{-1})} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{40}$$

and the difference equation is written as.

$$x_m(k) = -b_1 x_m(k-1) - b_2 x_m(k-2) + a_0 x_p^*(k) + a_1 x_p^*(k-1) + a_2 x_p^*(k-2) \tag{41}$$

3.   *Parameter adjusting mechanism*:

The gradient descent method is used to derive the AFC control law in Fig. 1. The objective of the parameters adjustment in FC is to minimize the square error between the mover position and the output of the RM. The instantaneous cost function is defined by

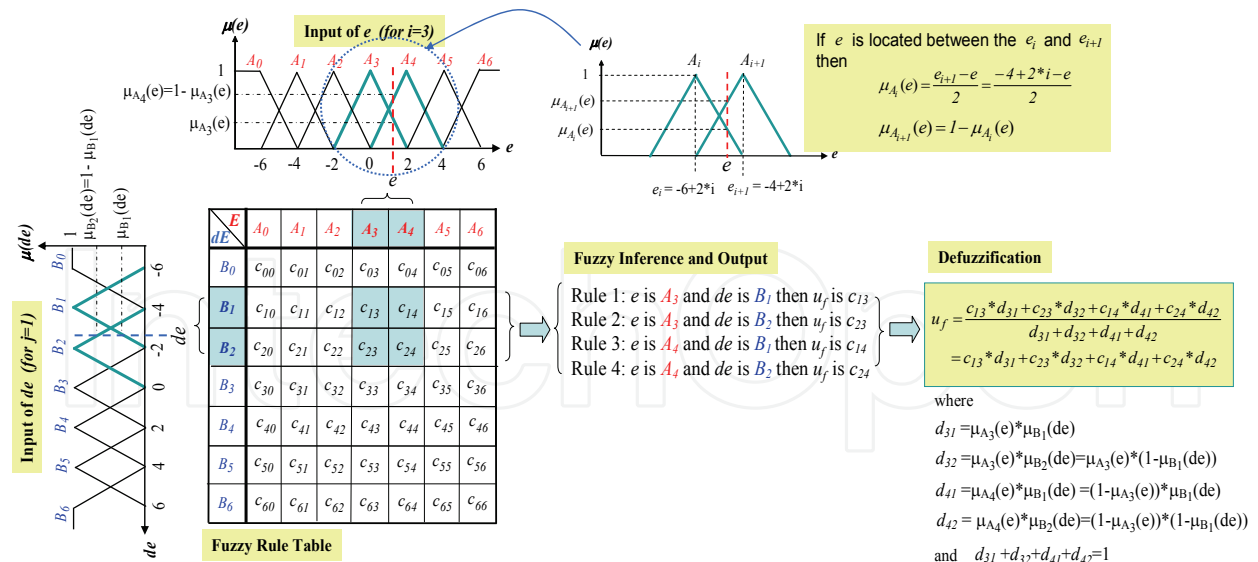$$J(k+1) = \frac{1}{2}e_m(k+1)^2 = \frac{1}{2}\left[x_m(k+1) - x_p(k+1)\right]^2 \tag{42}$$

Fig. 7. The symmetrical triangular membership function of e and de, fuzzy rule table, fuzzy inference and fuzzification

and the four defuzzifier parameters of $c_{j,i}$, $c_{j+1,i}$, $c_{j,i+1}$, $c_{j+1,i+1}$ are adjusted according to

$$\Delta c_{m,n}(k+1) \propto -\frac{\partial J(k+1)}{\partial c_{m,n}(k)} = -\alpha \frac{\partial J(k+1)}{\partial c_{m,n}(k)} \tag{43}$$

with $m = j, j+1$, $n = i, i+1$ and where α represents learning rate. However, following the similar derivation with (Kung & Tsai, 2007), the $\Delta c_{m,n}$ can be obtained as

$$\Delta c_{m,n}(k) = \alpha(K_p + K_i)\Psi K_v e(k)d_{n,m} \approx \alpha(K_p + K_i)Sign(\Psi)K_v e(k)d_{n,m} \tag{44}$$

with $m = j, j+1$ and $n = i, i+1$. Because the motor parameter Ψ is not easily to determined, so the $sign(\Psi)$ is used in (44). The sign (.) represents the sign operator.

4. *Point-to-point (PTP) Motion Trajectory*

The PTP trajectory is considered to evaluate the motion performance in PMLSM. For smoothing the move of the PMLSM at the start and stop condition, the motion trajectory is designed with the trapezoidal velocity profile and its formulation is shown as follows.

$$s(t) = \begin{cases} \frac{1}{2}At^2 + s_0 & 0 \le t \le t_a \\ v_m(t-t_a) + s(t_a) & t_a \le t \le t_d \\ -\frac{1}{2}A(t-t_d)^2 + v_m(t-t_d) + s(t_d) & t_d \le t \le t_s \end{cases} \tag{45}$$

Where $0<t<t_a$ is at the acceleration region, $t_a<t<t_d$ is at the constant velocity region, and $t_d<t<t_s$ is at the deceleration region. The S represents the position command; A is the acceleration/deceleration value; s_0 is the initial position; v_m is the maximum velocity; t_a, t_d and t_s represents the end time of the acceleration region, the start time of the deceleration region and the end time of the trapezoidal motion, respectively.

3. The design of a motion control IC for PMLSM drive

Figure 8 illustrates the internal architecture of the proposed FPGA implementation of a PTP motion trajectory, an AFC and a current vector controller for PMLSM drive system. The internal circuit comprises a Nios II embedded processor IP (Intelligent Properties) and a position control IP. The Nios II processor is depicted to perform the function of the PTP motion trajectory, generate the position command, collect the response data and communicate with external device. All programs in Nios II processor are coded in the C language. The position control IP includes mainly a circuit of the position AFC and speed P controller, a circuit for current controllers and coordinate transformation (CCCT), a QEP circuit, a SVPWM circuit and an ADC interface circuit. The sampling frequency of the position control loop is designed with 2kHz. The frequency divider generates 50 Mhz (Clk), 25 Mhz (Clk-step), 12 kHz (Clk-cur), and 2 kHz (Clk-sp) clock to supply all circuits in Fig. 8. The internal circuit of CCCT performs the function of two PI controllers, table look-up for *sin/cos* function and the coordinate transformation for Clark, Park, inverse Park, modified inverse Clarke. The CCCT circuit designed by FSM is shown in Fig. 9, which uses one adder, one multiplier, an one-bit left shifter, a look-up-table and manipulates 24 steps machine to carry out the overall computation. The data type is 12-bit length with Q11 format and 2's complement operation. In Fig. 9, steps s0~s1 is for the look-up *sin/cos* table; steps s2~s5 and s5~s8 are for the transformation of Clarke and Park, respectively; steps s9~s14 is for the computation of d-axis and q-axis PI controller; and steps s15~s19 and s20~s23 represent the transformation of the inverse Park and the modified inverse Clarke, respectively. The
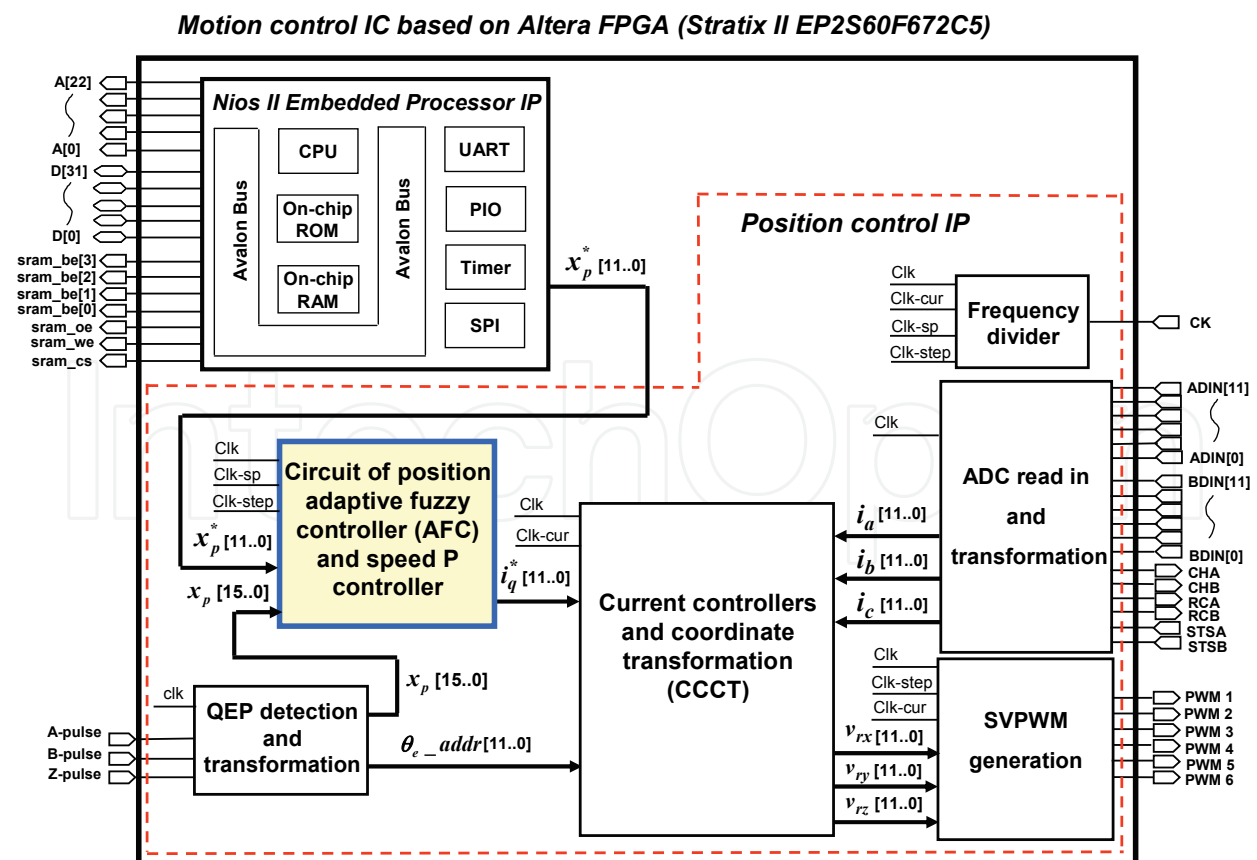

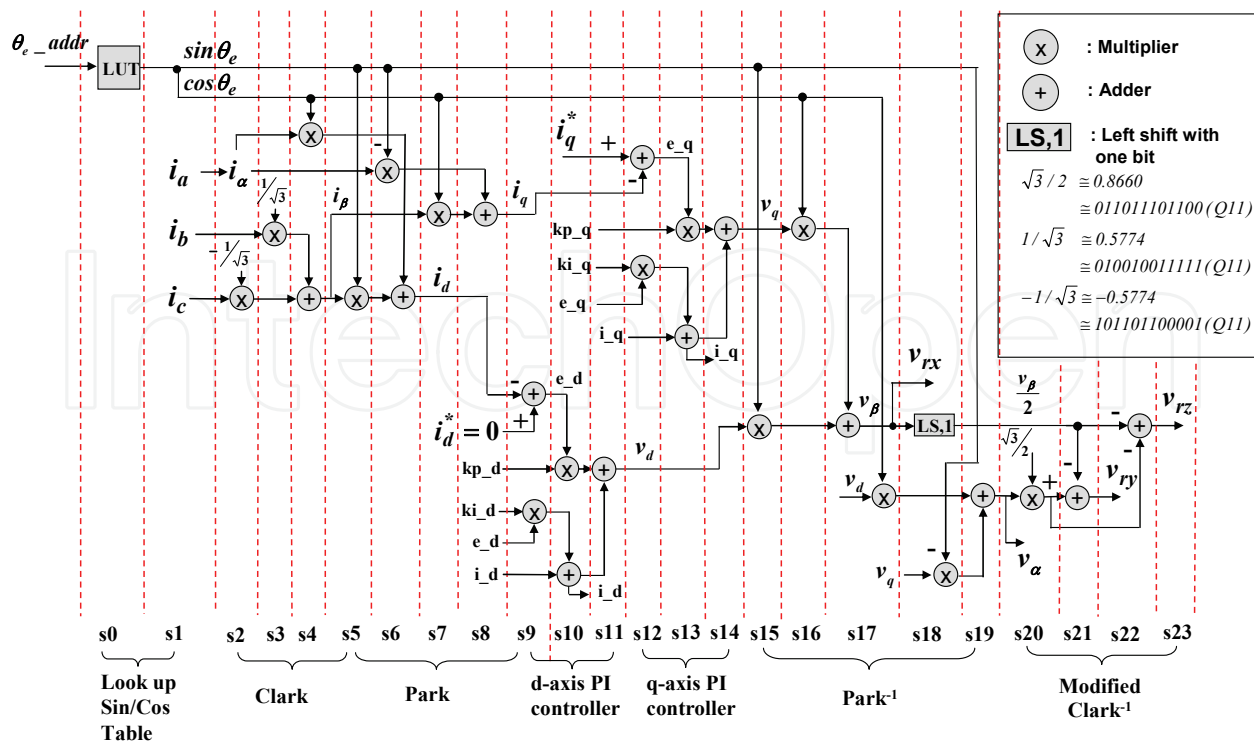Fig. 8. Internal architecture of a motion control IC for PMLSM in FPGA

Fig. 9. Designed CCCT circuit in Fig. 8

operation of each step in FPGA can be completed within 40ns (25 MHz clock); therefore total 24 steps need 0.96 μs operation time. Although the FSM method needs more operation time than the parallel processing method in executing CCCT circuit, it doesn't loss any control performance in overall system because the 0.96 μs operation time is much less than the designed sampling interval, 62.5 μs (16 kHz) of current control loop in Fig. 1. To prevent numerical overflow and alleviate windup phenomenon, the output values of I controller and PI controller are both limited within a specific range.

An FSM is also employed to model the AFC of the position loop and P controller of the speed loop in PMLSM and shown in Fig. 10, which uses one adder, one multiplier, a look-up table, comparators, registers, etc. and manipulates 35 steps machine to carry out the overall computation. With exception of the data type in reference model are 24-bits, others data type are designed with 12-bits length, 2's complement and Q11 format. Although the algorithm of AFC is highly complexity, the FSM can give a very adequate modeling and easily be described by VHDL. Furthermore, steps $s_0 \sim s_6$ execute the computation of reference model output; steps $s_6 \sim s_9$ are for the computation of mover velocity, position error and error change; steps $s_9 \sim s_{12}$ execute the function of the fuzzification; s_{13} describe the look-up table and $s_{14} \sim s_{22}$ defuzzification; and steps $s_{23} \sim s_{34}$ execute the computation of velocity and current command output, and the tuning of fuzzy rule parameters. 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.10 can be completed within 40ns (25 MHz clock) in FPGA; therefore total 35 steps need a 1.4μs operation time. It doesn't loss any control performance for the overall system because the operation time with 1.4μs is much less than the sampling interval, 500 μs (2 kHz), of the position control loop in Fig.1. In Fig. 8, the SVPWM circuit and QEP circuit are presented in Fig. 11(a) and Fig. 11(b). The SVPWM circuit in Fig.11(a) herein is designed to be 16 kHz frequency and 1μs dead-band, respectively. The algorithm
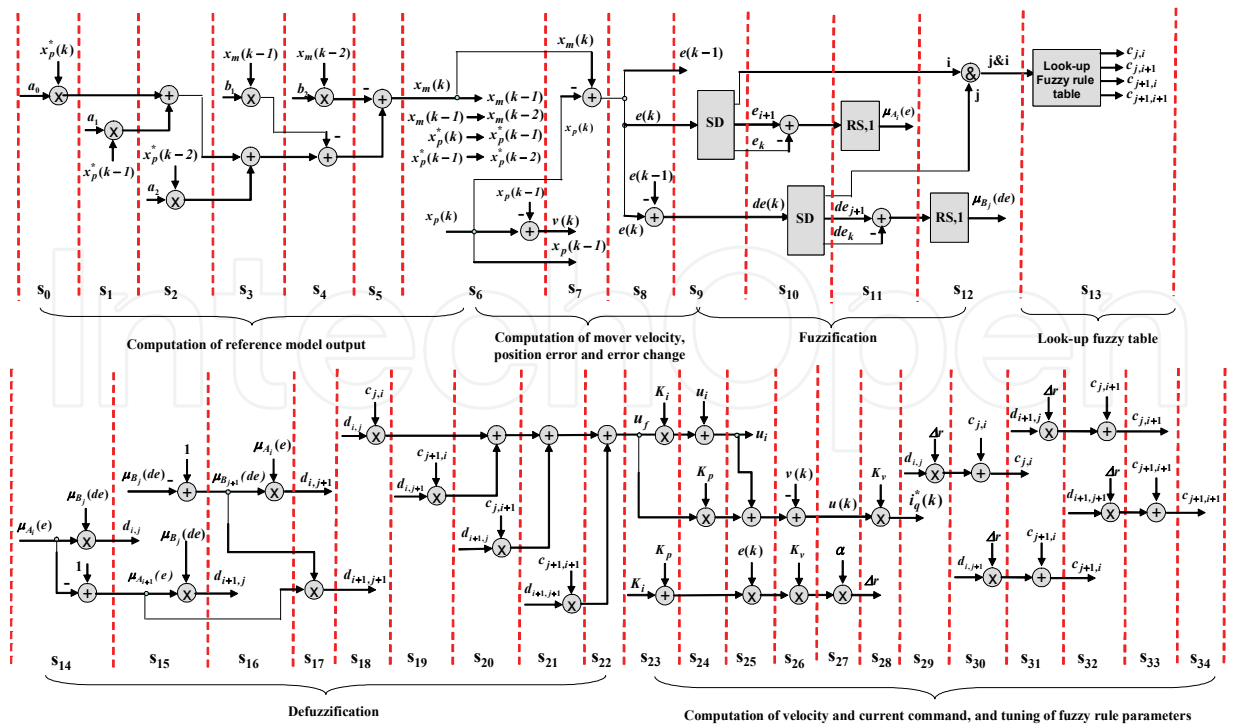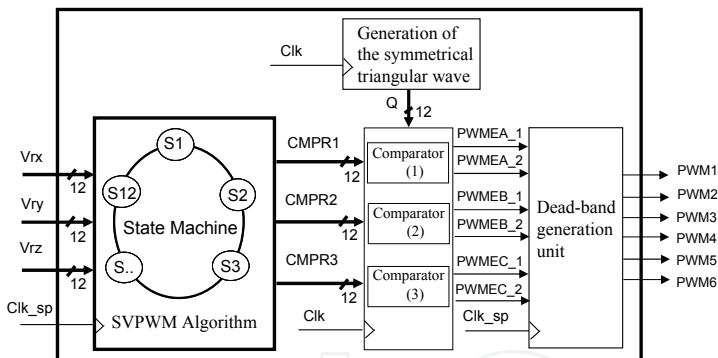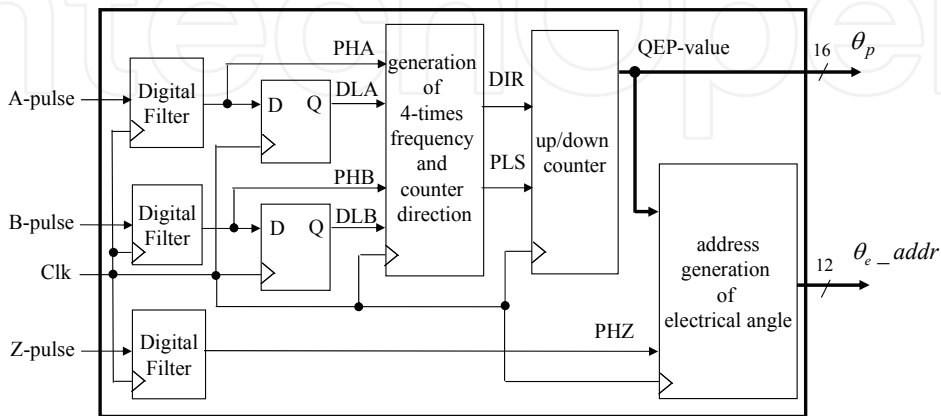
Fig. 10. State diagram of an FSM for describing the AFC in position loop and P controller in speed loop

(a)

(b)

Fig. 11. (a) Circuit of SVPWM generation (b) circuit of QEP detection and transformation

of SVPWM refers to Section 2.2. The circuit of the *QEP* module is shown in Fig.11(b), which consists of two digital filters, a decoder and an up-down counter. The filter is used for reducing the noise effect of the input signals *PA* and *PB*. The pulse count signal *PLS* and the rotating direction signal *DIR* are obtained using the filtered signals through the decoder circuit. The *PLS* signal is a four times frequency pulses of the input signals *PA* or *PB*. The *Qep* value can be obtained using *PLS* and *DIR* signals through a directional up-down counter. The overall resource usage of the AFC circuit needs 8,055 ALUTs, the Nios II embedded processor IP needs 8,275 ALUTs and 46,848 RAM bits and the position control IP needs 12,269 ALUTs and 297,984 RAM bits in FPGA. Therefore, the motion control IC uses 42.4% ALUTs resource and 13.6% RAM resource of Stratix II EP2S60.

4. Experimental results

The overall experimental system depicted in Fig.1 includes an FPGA (Stratix II EP2S60F672C5), a voltage source IGBT inverter and a PMLSM. The PMLSM was manufactured by the BALDOR electric company; and it is a single-axis stage with a cog-free linear motor and a stroke length with 600mm. The parameters of the motor are: R_s = 27 Ω , $L_d = L_q$ = 23.3 mH, K_t = 79.9N/A. The input voltage, continuous current, peak current (10% duty) and continuous power of the PMLSM are 220V, 1.6A, 4.8A and 54W, respectively. The maximum speed and acceleration are 4m/s and 4 g but depend on external load. The moving mass is 2.5Kg, the maximum payload is 22.5Kg and the maximum thrust force is 73N under continuous operating conditions. A linear encoder with a resolution of 5μm is mounted on the PMLSM as the position sensor, and the pole pitch is 30.5mm (about 6100 pulses). The inverter has three sets of IGBT power transistors. The collector-emitter voltage of the IGBT is rated 600V; the gate-emitter voltage is rated ±20V, and the DC collector current is rated 25A and in short time (1ms) is 50A. The photo-IC, Toshiba TLP250, is used in the gate driving circuit of IGBT. Input signals of the inverter are PWM signals from the FPGA device.

For validating the effectiveness of the current vector control in Fig. 1, the input current command, $(i_d^*, i_q^*) = (0A, 1A)$ is set, and the measured currents of i_d , i_q are shown in Fig. 12(a), the corresponding currents in *a-b-c* axes and in $\alpha - \beta$ axes are shown in Fig. 12(b) and Fig. 12(c), respectively. In Fig. 12, as a result of the EMI effect in the motor driver board, the ripple in measured current has a little high. Nevertheless, the experiment result is still presented that the measured current could tracks the current command. Furthermore, it not merely validates the function of the current vector control, but also could make the PMLSM decouple.

Have confirmed the effectiveness of the current loop vector control in the PMLSM drive in loop, the realization of position controller based on the FPGA in Fig.1 is further evaluated. The control sampling frequency of the current, speed and position loops are designed as 16kHz, 2kHz and 2kHz, respectively. In the proposed motion control IC, the current controller, the speed controller and the adaptive fuzzy position controller are all realized by hardware in FPGA, and the PTP motion trajectory algorithm is implemented by software using the Nios II embedded processor. The speed controller adopts a P controller with gain K_v=1.4.The AFC is used in the position loop, the membership function and the initial fuzzy rule table are designed as Fig.13(a), and the PI gains are chosen by K_p=2.5, K_i=0.08. The transfer function of the reference model is selected by a second order system with the

natural frequency of 40 rad/s and damping ratio of 1. The step response is first tested to evaluate the performance of the proposed controller. Figure 14 shows the position step responses of the mover under a payload of 0 Kg and 11 Kg using the FC (learning rate=0) when the position command is a 4/3Hz square wave signal with a 10mm amplitude. The parameters $c_{i,j}$ of the fuzzy rule table are adequately selected in Fig. 13(a) at the 0kg payload condition, and the step response shows a good dynamic response with a rising time of 0.1s, no overshoot and a near-zero steady state in Fig. 14(a). However, when a 11 kg load is
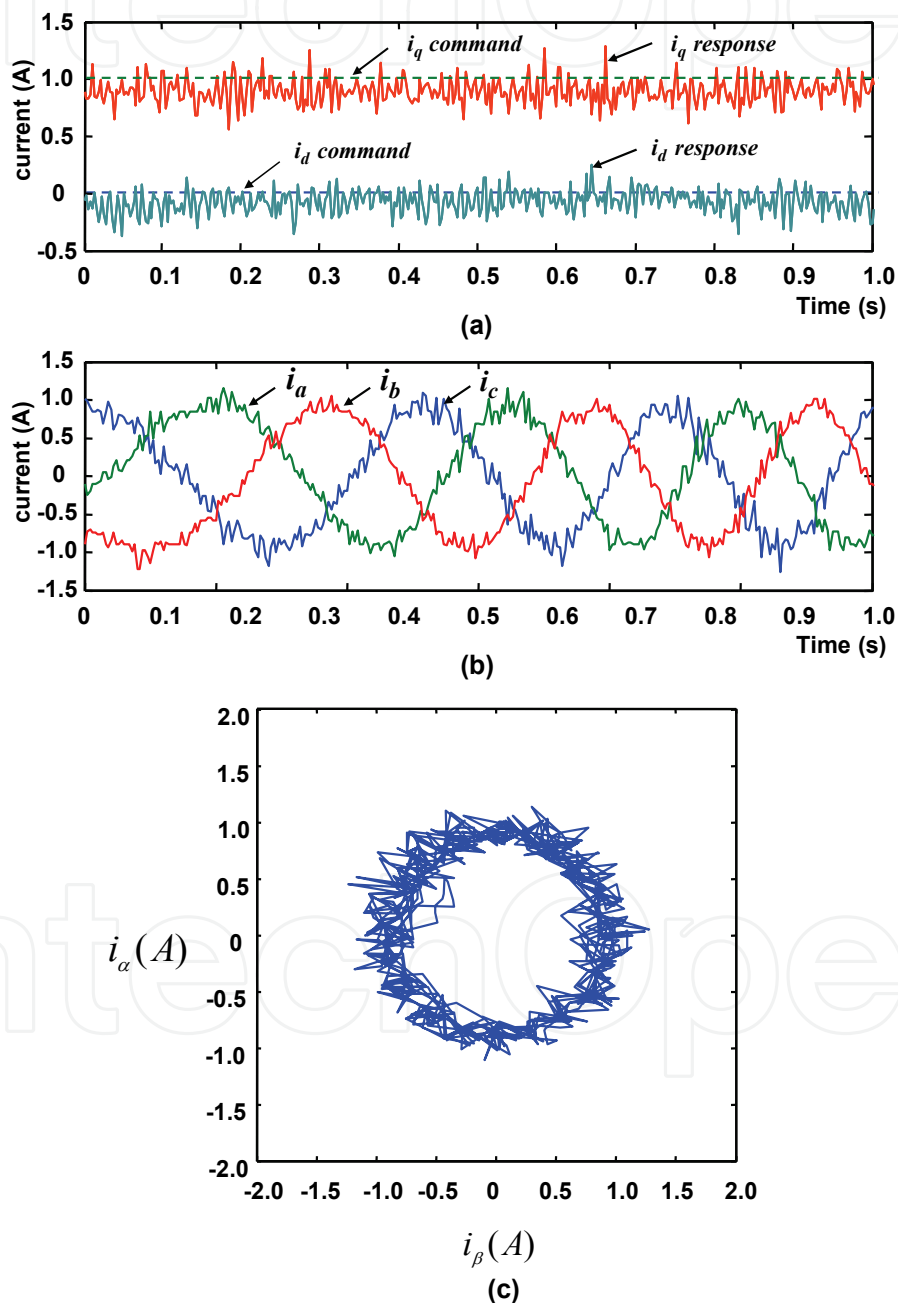
Fig. 12. (a) i_d^*, i_q^* and i_d, i_q response in current control loop (b) Three phase waveforms (c) i_α, i_β response in current control loop
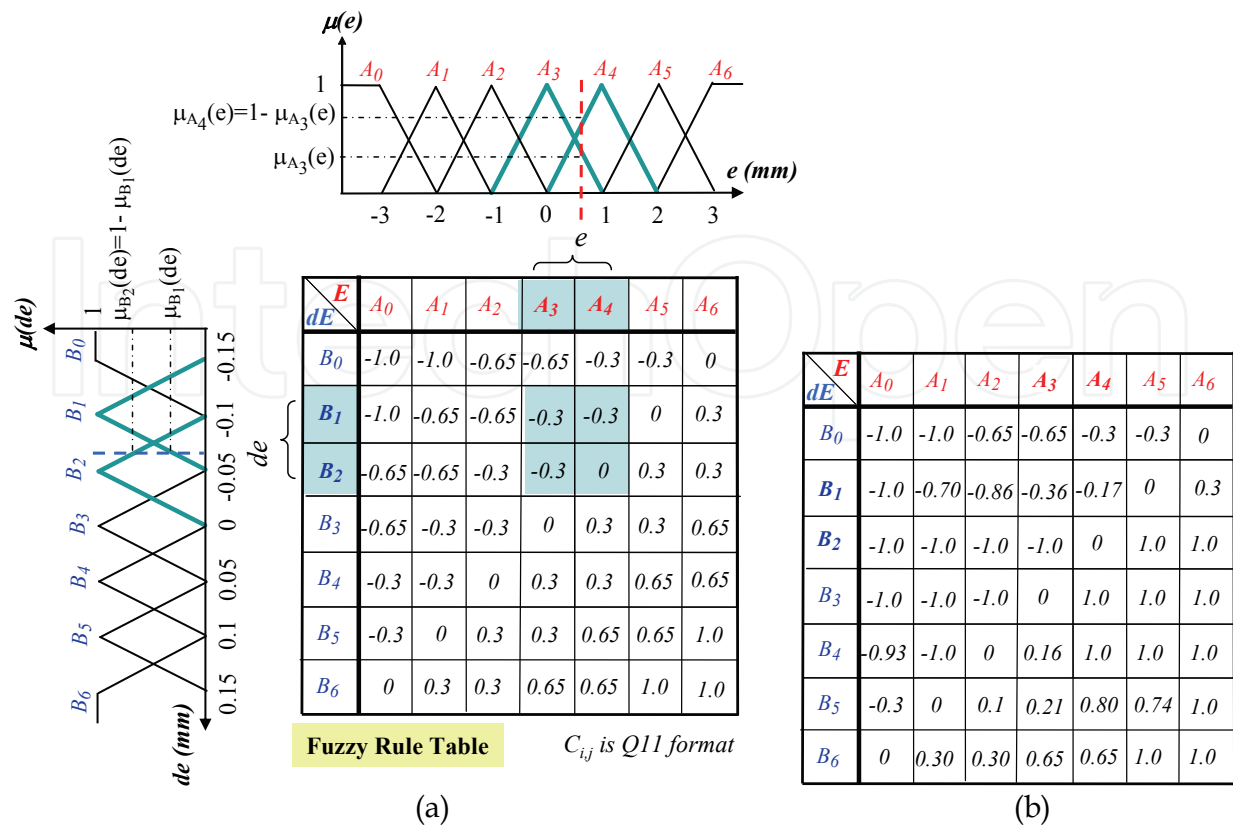
Fuzzy Rule Table — $C_{i,j}$ is Q11 format

(a)

dE\\E	A_0	A_1	A_2	A_3	A_4	A_5	A_6
B_0	-1.0	-1.0	-0.65	-0.65	-0.3	-0.3	0
B_1	-1.0	-0.65	-0.65	-0.3	-0.3	0	0.3
B_2	-0.65	-0.65	-0.3	-0.3	0	0.3	0.3
B_3	-0.65	-0.3	-0.3	0	0.3	0.3	0.65
B_4	-0.3	-0.3	0	0.3	0.3	0.65	0.65
B_5	-0.3	0	0.3	0.3	0.65	0.65	1.0
B_6	0	0.3	0.3	0.65	0.65	1.0	1.0

(b)

dE\\E	A_0	A_1	A_2	A_3	A_4	A_5	A_6
B_0	-1.0	-1.0	-0.65	-0.65	-0.3	-0.3	0
B_1	-1.0	-0.70	-0.86	-0.36	-0.17	0	0.3
B_2	-1.0	-1.0	-1.0	-1.0	0	1.0	1.0
B_3	-1.0	-1.0	-1.0	0	1.0	1.0	1.0
B_4	-0.93	-1.0	0	0.16	1.0	1.0	1.0
B_5	-0.3	0	0.1	0.21	0.80	0.74	1.0
B_6	0	0.30	0.30	0.65	0.65	1.0	1.0

Fig. 13. Fuzzy membership function and (a) the initial fuzzy rule table in experiment (b) the tuned fuzzy rule table at 4th second

added upon the mover and the same fuzzy control rule table and controller parameters are used, the position dynamic response worsens and exhibits a 14.2% overshoot in Fig. 14(b). It reveals that the dynamic performance of the PMLSM is affected by the payload on the mover. Accordingly, an AFC is adopted in Fig.1 to solve this problem. When the proposed AFC is used with learning rate being 0.1, the tracking results are highly improved and presented in Fig. 15. Initially, the mover of the PMLSM tracks the output of the reference model with oscillation. After five square wave commands, the $c_{i,j}$ parameters are tuned to adequate values which is shown in Fig. 13(b), and the mover can closely follow the output of the reference model. Simultaneously, in Fig. 15(c), the peak of the instantaneous cost function is gradually decayed to a constant value. Secondly, the frequency response is considered to evaluate the performance of the proposed controller. A tested input signal of a sinusoid wave with 10mm amplitude and the frequency variation from initial 2 Hz to final 6 Hz is provided. In this design, the frequency tracking response and the tracking error of the PMLSM without and with adaptation under 11kg payload are shown in Fig. 16 and Fig.17. Figure 17 reveals that the position tracking error by using the AFC (learning rate=0.1) is only 0.35~0.7 times of that obtained by using the FC in Fig.16. However, Fig.16 reveals the phase lag phenomenon using the FC is more serious than using the AFC in Fig.17. Finally, to test the tracking performance of a PTP motion trajectory, a repeated go-and-return displacement motion command with a trapezoidal velocity profile under 11kg payload is provided, and the overall displacement, the maximum velocity and the acceleration/ deceleration are designed to be 400 mm, 1m/s and 3.6m/s², respectively. The tracking results concerning the displacement trajectory and its velocity profile corresponding to the aforementioned input

commands using the FC and the AFC are shown in Fig. 18 and Fig. 19, respectively. The experimental results reveal that both in position and velocity response are well tracked, but the position tracking error by using the AFC is less than the one by using the FC. Therefore, the experimental results in Figs. 12 to 19 demonstrate that the proposed FPGA-based AFC and PTP motion trajectory for the PMLSM drive is effective and robust.

5. Conclusion

A motion control IC for a PMLSM drive based on the FPGA technology is successfully demonstrated in this chapter. The conclusions herein are summarized as follows. Firstly, an FSM joined by one multiplier, one adder, one LUT, some comparators and registers has been employed to model the overall AFC algorithm for the PMLSM, such that it not only is easily implemented by VHDL but also can reduce the FPGA resources usage. Secondly, the functionalities required to build a fully digital motion controller of the PMLSM, such as a PTP motion trajectory, an AFC in the position loop, a P controller in the speed loop and a current vector controller are all realized within one FPGA. Thirdly, the PTP motion trajectory scheme is implemented in software by using Nios II embedded processor and the AFC and the current vector controller algorithm are implemented in hardware by FPGA. The software/hardware co-design technology with the parallel processing can make the system performance increased. Finally, the experimental results by step response, frequency response and PTP motion tracking have been revealed well performance in the proposed FPGA-based PMLSM motion control system.
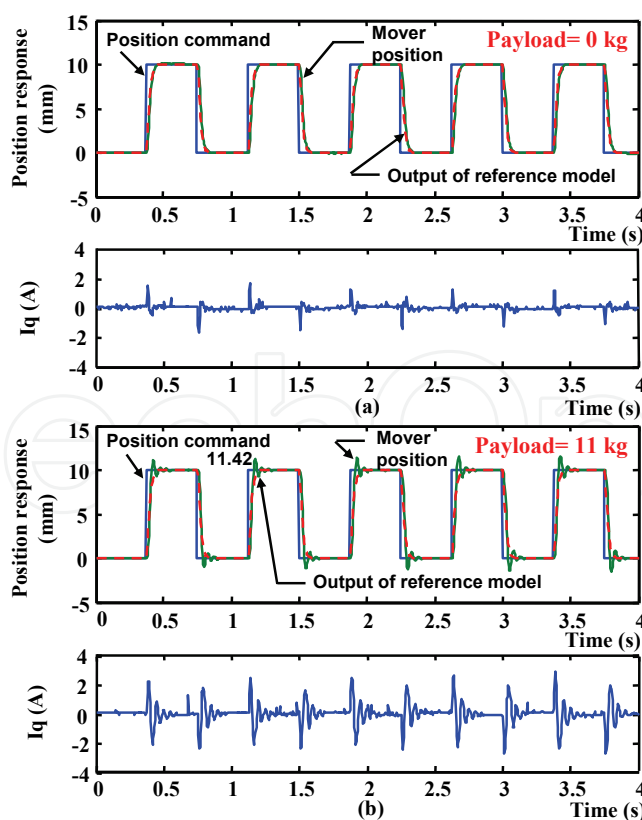
Fig. 14. The position and current responses of a step position command using the FC under (a) 0kg and (b) 11 kg payload
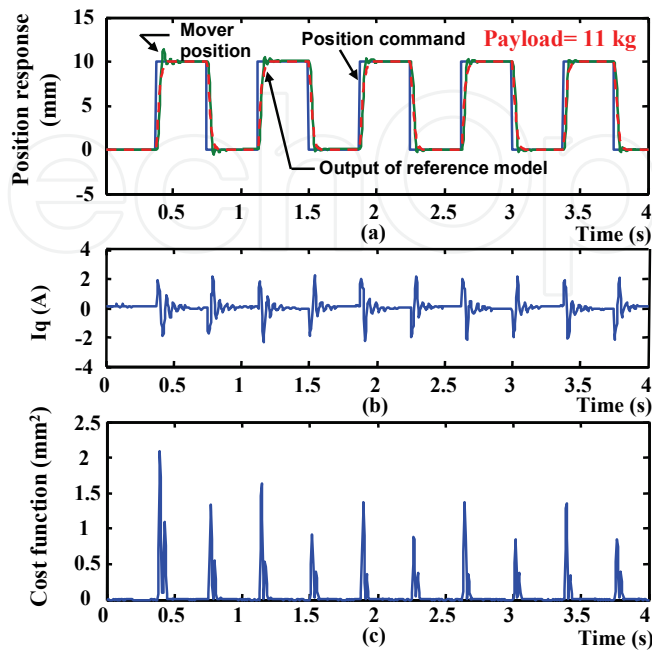
Fig. 15. The (a) position (b) current and (c) cost function responses of a step position command using the AFC under 11 kg payload
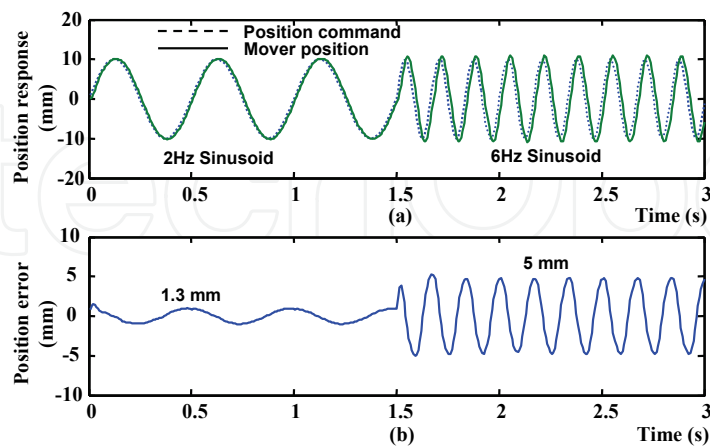

Fig. 16. (a) The frequency and (b) the position error responses of a 2Hz to 6Hz sinusoid input signal using the FC under 11 kg payload
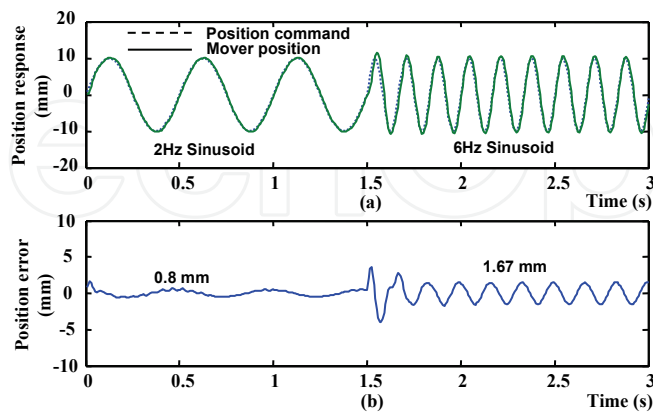
Fig. 17. (a) The frequency and (b) the position error responses of a 2Hz to 6Hz sinusoid input signal using the AFC under 11 kg payload
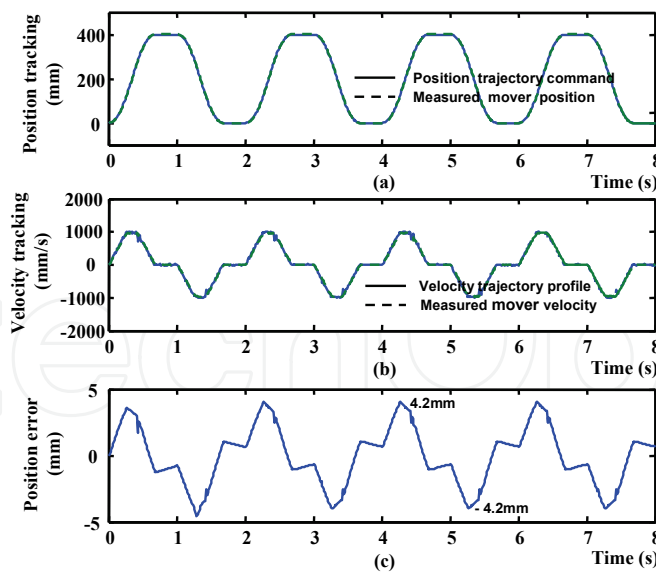

Fig. 18. The PTP motion trajectory experiment using the FC under a maximum velocity of 1m/s, acceleration/deceleration of 3.6m/s² and with 11kg payload and its (a) displacement tracking (b) velocity tracking (c) displacement tracking error responses
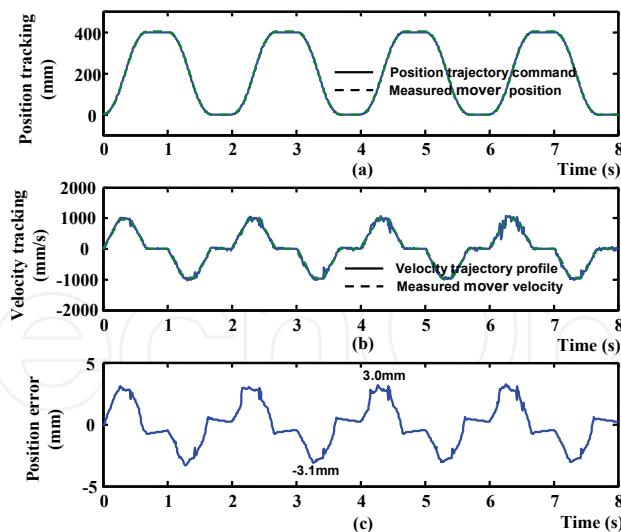
Fig. 19. The PTP motion trajectory experiment using the AFC under a maximum velocity of 1m/s, acceleration/deceleration of $3.6m/s^2$ and with 11kg payload and its (a) displacement tracking (b) velocity tracking (c) displacement tracking error responses

6. References

Altera (2008): www.altera.com.

Budig, P.K. (2000). The application of linear motors, *Proceedings of PIEMC 2000*. vol. 3, pp.1336-1340.

Cho, J. U.; Le, Q. N. & Jeon, J. W. (2009) An FPGA-based multiple-axis motion control chip, *IEEE Trans. Ind. Electron.*, vol. 56, no.3, pp.856-870.

Gieras, F. & Piech, Z.J. (2000). *Linear synchronous motors – transportation and automation systems*, CRC Press.

Hsu, Y.C.; Tsai, K.F.; Liu, J.T. & Lin, E.S. (1996) *VHDL modeling for digital design synthesis*, KLUWER ACADEMIC PUBLISHERS, TOPPAN COMPANY (S) PTE LTD.

Huang, H.C. & Tsai, C.C. (2009) FPGA Implementation of an embedded robust adaptive controller for autonomous omnidirectional mobile platform, *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1604-1616.

Jung, S. & Kim, S.S. (2007). Hardware implementation of a real-time neural network controller with a DSP and an FPGA for nonlinear systems, *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 265-271.

Kung, Y.S.; Tseng K.H. & Tai T.Y. (2006) FPGA-based servo control IC for X-Y table, *Proceedings of the IEEE International Conference on Industrial Technology (ICIT'06)*, pp. 2913-2918.

Kung, Y.S. & Tsai, M.H. (2007). FPGA-based speed control IC for PMSM drive with adaptive fuzzy control, *IEEE Trans. on Power Electronics*, vol. 22, no. 6, pp. 2476-2486.

Kung, Y.S. (2008) Design and implementation of a high-performance PMLSM drives using DSP chip, *IEEE Trans. on Ind. Electron.*, vol. 55, no. 3, pp. 1341-1351.

Kung, Y.S.; Fung, R.F. & Tai, T.Y.(2009) Realization of a motion control IC for X-Y Table based on novel FPGA technology, *IEEE Trans. Ind. Electron.*, vol. 56, no. 1, pp. 43-53.

Li, T.S.; Chang S.J. & Chen, Y.X. (2003). Implementation of human-like driving skills by autonomous fuzzy behavior control on an FPGA-based car-like mobile robot, *IEEE Trans. Ind. Electro.*, vol. 50, no.5, pp. 867-880.

Lin, F.J.; Wang, D.H. & Huang, P.K. (2005) FPGA-based fuzzy sliding mode control for a linear induction motor drive, *IEE Proc.- Electr. Power Application*, vol. 152, no.5, pp. 1137-1148.

Lin, F. J.; Huang, P. K. & Chou, W. D. (2007). Recurrent fuzzy neural network controlled linear induction motor servo drive using genetic algorithm, *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1449-1461.

Liu, T.H.; Lee Y.C. & Chang, Y.H. (2004). Adaptive controller design for a linear motor control system," *IEEE Trans. on Aerospace and Electronics System*, vol. 40, no.2, pp. 601-613.

Qingding, G.; Qingtao H. & Yanli, Q. (2002). Neural network real-time IP position controller on-line design for permanent magnetic linear synchronous motor, *Proceedings of the International on Advanced Motion Control*, pp. 386-389.

McLean, G.W. (1988) Review of recent progress in linear motors, *IEE Proc., Part B*, vol. 135, no. 6, pp. 380-416, November 1988.

Monmasson, E. & Cirstea, M.N. (2007). FPGA design methodology for industrial control systems – a review, *IEEE Trans. on Ind. Electron.*, Vol. 54, No. 4, pp.1824-1842.

Naouar, M.W.; Monmasson, E.; Naassani, A.A.; Slama-Belkhodja, I. & Patin, N. (2007). FPGA-based current controllers for AC machine drives – a review, *IEEE Trans. Ind. Electron.*, vol. 54, no.4, pp.1907-1925.

Wai, R.J. & Chu, C.C. (2007). Motion control of linear induction motor via petri fuzzy neural network, *IEEE Trans. Ind. Electron.*, vol.54, no. 1, pp. 281-295.

Wei, R.; Gao, X.H.; Jin, M.H.; Liu, Y.W.; Liu, H.; Seitz, N.; Gruber, R. & Hirzinger, G. (2005). FPGA based hardware architecture for HIT/DLR hand, *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and System*, pp. 523~528.
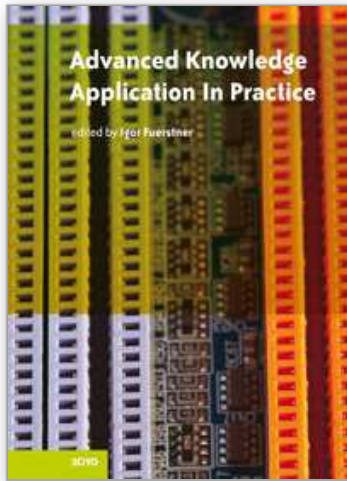
Advanced Knowledge Application in Practice

Edited by Igor Fuerstner

The integration and interdependency of the world economy leads towards the creation of a global market that offers more opportunities, but is also more complex and competitive than ever before. Therefore widespread research activity is necessary if one is to remain successful on the market. This book is the result of research and development activities from a number of researchers worldwide, covering concrete fields of research.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds