FPGA-Based Neural Fuzzy Controller Design for PMLSM Drive

Ying-Shieh Kung, Nguyen Khanh Quang and Le Thi Van Anh
Department of Electrical Engineering
Southern Taiwan University
Tainan, Taiwan, R. O. C.  710
kung@mail.stut.edu.tw

Abstract—Based on the technology of field programmable gate array (FPGA), a realization of fuzzy control (FC) system with radial basis function neural network (RBF NN) tuning is presented to a permanent linear synchronous motor (PMLSM) drive in this paper. Firstly, a mathematic model of the PMLSM drive is defined; then to increase the performance of the PMLSM drive system, an FC constructed by a fuzzy basis function and its parameter adjustable mechanism using RBF NN is applied to the position control loop of the PMLSM drive system to cope with the effect of the system dynamic uncertainty and the external load. Secondly, FPGA by using finite state machine (FSM) method is presented to realize the aforementioned controllers, and VHIC hardware description language (VHDL) is adopted to describe the circuit of the FSM. Finally, an experimental system is established to verify the effectiveness of the proposed FPGA-based neural fuzzy control system for PMLSM, and some experimental results are confirmed theoretically.

Keywords- FPGA; Neural fuzzy controller; PMLSM; Finite state machine; VHDL;

I. INTRODUCTION

PMLSM has been increasingly used in many automation control fields as actuators [1-3], due to its advantages of superior power density, high-performance motion control with fast speed and better accuracy. However, the PMLSM does not use conventional gears or ball screws, so the payload upon the mover greatly affects the positioning performance [4]. To cope with this problem, many intelligent control techniques [5-6], such as FC, neural networks control (NNC), etc. have been developed and applied to the position control of the PMLSM drive to obtain high operating performance. Compared with other nonlinear approaches, FC has two main advantages, as follows: (1) FC has a special non-linear structure that is universal for various or uncertainty plants. (2) The formulation of FC rule can be easily achieved by control engineering knowledge, such as dynamic response characteristics, and it doesn’t require a mathematical model of controlled plant. However, it is not an easy task to obtain an optimal set of fuzzy membership functions and rules in FC. In literatures, the genetic algorithm method [7] or gradient descent method are all possible methods to solve this problem. But, to obtain an optimal set of fuzzy membership functions and rules, the FC will become further complication in overall computation; therefore, limited fuzzy rules are used in their proposed method. In this paper, a neural fuzzy controller (NFC) is proposed. For easy realization consideration, the membership functions in FC part are fixed and only defuzzifier parameters need to be tuned by using the gradient descent method. And a RBF NN is used to identify the plant dynamic and provide more accuracy plant information during parameters tuning of FC.

Although the execution of NNC or FC requires many computations, digital signal processor (DSP) and FPGA can provide a solution in this issue [8-9]. 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 [10-12] than DSP. Recently, Li, T.S. [13] utilized an FPGA to implement autonomous fuzzy behavior control on mobile robot. Lin, F.J. [9] 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 this problem, a FSM [14] joined by some multipliers, some adders, a look-up table (LUT), some comparators and registers are proposed to model the NFC algorithm of the PMLSM drive system. Due to the FSM belongs to the sequential processing method; the FPGA resources usage can be greatly reduced. In this paper, FPGA chip employed is an Altera Stratix II EP2S60F672C5 [15] which has 48,352 ALUTs, maximum 492 user I/O pins, 36 DSP blocks, 2,544,192 bits of RAM, and a Nios II processor, which can be embedded into FPGA. 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 NFC controller.

II. SYSTEM DESCRIPTION OF PMLSM DRIVE AND THE CONTROLLER DESIGN

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

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

\[
\frac{di_d}{dt} = -\frac{R_d}{L_d} i_d + \frac{\pi}{\tau} L_q i_q + \frac{1}{L_q} v_d
\]  
(1)

\[
\frac{di_q}{dt} = -\frac{R_q}{L_q} i_q - \frac{\pi}{\tau} \lambda_j \dot{x}_p + \frac{1}{L_q} v_q
\]  
(2)

where \(v_d, v_q\) are the d and q axis voltages; \(i_d, i_q\) are the d and q axis currents; \(R_d\) is the phase winding resistance; \(L_d, L_q\) are the d and q axis inductance; \(\dot{x}_p\) is the translator speed; \(\lambda_j\) is the permanent magnet flux linkage; \(\tau\) is the pole pitch. The developed electromagnetic thrust force is given by

\[
F_e = \frac{3\pi}{2\tau} ((L_d - L_q) i_d + \lambda_j) i_q
\]  
(3)

The current control of a PMLSM drive is based on a vector control approach. That is, if we control \(i_d\) to 0, 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,

\[
F_e = \frac{3\pi}{2\tau} \lambda_j i_q \Delta K_i i_q
\]  
(4)

with

\[
K_i = \frac{3\pi}{2\tau} \lambda_j
\]  
(5)

and the mechanical dynamic equation of PMLSM is

\[
F_e - F_f = M_n \frac{d^2 x_p}{dt^2} + B_n \frac{dx_p}{dt}
\]  
(6)

where \(F_e, F_f, M_n, B_n\) and \(F_f\) represent the motor thrust force, the force constant, the total mass of the moving element, the viscous friction coefficient and the external force, respectively.

B. Neural fuzzy controller (NFC) in position control loop

The dash rectangular area in Fig. 1 presents the architecture of an NFC for the PMLSM drive. It consists of a FC, a reference model and a RBF NN based 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

\[
e(k) = x_m(k) - x_p(k)
\]  
(7)

\[
de(k) = e(k) - e(k-1)
\]  
(8)

and \(e\), \(de\) and \(ut\) are input and output variables of FC, respectively. 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 linguistic variables as \(E\) and \(DE\). The linguistic 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 linguistic value of \(E\) and \(DE\) is based on the symmetrical triangular membership function which is shown in Fig.2.

(b) Compute the membership degree of \(e\) and \(de\). Figure 2 shows that the only two linguistic values are excited (resulting in a non-zero membership) in any input value, and the membership degree \(\mu_{A_i}(e)\) can be derived by

\[
\mu_{A_i}(e) = \frac{e_{i+1} - e - e_i - 6 + 2(i+1)}{2}\quad\text{and}\quad\mu_{A_i+1}(e) = I - \mu_{A_i}(e)
\]  
(9)

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

(c) Select the initial fuzzy control rules by referring to the dynamic response characteristics, such as,

\[
IF\quad e \quad is \quad A_i \quad and \quad DE \quad is \quad B_j \quad THEN \quad u_f \quad is \quad c_{ij}
\]  
(10)

where \(i\) and \(j\) are \(0\) to \(6\), \(A_i\) and \(B_j\) are fuzzy number, and \(c_{ij}\) is real number. The graph of fuzzification and fuzzy rule table is shown in Fig. 2.

(d) Construct 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. 3 will be inferred, actually only 4 fuzzy rules can be effectively excited to generate a non-zero output. Therefore, the (11) can be replaced by the following expression:

\[
u_f(e,de) = \frac{\sum_{i=0}^{6} \sum_{j=0}^{6} c_{ij} \mu_{A_i}(e) \mu_{B_j}(de)}{\sum_{i=0}^{6} \sum_{j=0}^{6} \mu_{A_i}(e) \mu_{B_j}(de)}
\]  
(11)

\[
\Delta \sum_{i=0}^{6} \sum_{j=0}^{6} c_{ij} d_{im} = \sum_{i=0}^{6} \sum_{j=0}^{6} \mu_{A_i}(e) \mu_{B_j}(de)
\]
where \( d_m \) is the adjustable parameters. In addition, by using (9), it is straightforward to obtain
\[
\sum_{r=1}^{s} d_m = 1 \text{ in (11)}.
\]

![Fig. 2. The symmetrical triangular membership function of e and de, fuzzy rule table, fuzzy inference and fuzzification.](image)

(2) **Radial basis function neural network (RBF NN)**

The RBF NN adopted here is a three-layer architecture which is shown in Fig. 3 and comprised of one input layer, one hidden layer and one output layer.

The RBF NN has three inputs by \( n_r \), \( x_r(k-1) \) and \( x_r(k-2) \) and its vector form is represented by
\[
X = [u(k), x_r(k-1), x_r(k-2)]^T
\]

Furthermore, the multivariate Gaussian function is used as the activated function in hidden layer of RBF NN, and its formulation is shown as follows.
\[
h_r = \exp(-\frac{\|X - c_r\|^2}{2\sigma_r^2}), r = 1,2,3,4,...,q
\]

where \( c_r = [c_{r1}, c_{r2}, ..., c_{rq}] \) and \( \sigma_r \) denote the node center and node variance of \( r \)-th neuron, and \( \|X - c_r\| \) is the norm value which is measured by the inputs and the node center at each neuron. And the network output in Fig. 3 can be written as
\[
x_{rpf} = \sum_{r=1}^{q} w_r h_r
\]

where \( x_{rpf} \) is the output value; \( w_r \) and \( h_r \) are the weight and output of \( r \)-th neuron, respectively.

The instantaneous cost function is defined as
\[
J_\varepsilon = \frac{1}{2} (x_{rpf} - x^*)^2
\]

then according to the gradient descent method, the learning algorithm of weights, node center and variance are as follows:
\[
w_r(k+1) = w_r(k) + \eta c_{m,n} \frac{\partial J_\varepsilon}{\partial w_r}
\]
\[
c_r(k+1) = c_r(k) + \eta c_{m,n} \frac{\partial J_\varepsilon}{\partial c_r}
\]
\[
\sigma_r(k+1) = \sigma_r(k) + \eta c_{m,n} \frac{\partial J_\varepsilon}{\partial \sigma_r}
\]

where \( r=1,2,...,q \), \( s=1,2,3 \) and \( \eta \) is a learning rate. Further, the \( \frac{\partial J_\varepsilon}{\partial u} \) is Jacobian transformation and can be derived from Fig.3
\[
\frac{\partial J_\varepsilon}{\partial u} = \sum_{r=1}^{q} w_r \frac{\partial J_\varepsilon}{\partial x_r}
\]

(3) **Adjusting mechanism of fuzzy controller**

The gradient descent method is used to derive the FC control law in Fig. 1. The adjusting of FC parameters is to minimize the square error between the mover position and the output of the reference model. The instantaneous cost function is defined by
\[
J_\varepsilon = \frac{1}{2} (x(k) - x^*)^2
\]

and the parameters of \( c_{m,n} \) are adjusted according to
\[
\Delta c_{m,n} = -\frac{\partial J_\varepsilon}{\partial c_{m,n}} = -\alpha \frac{\partial J_\varepsilon}{\partial u}
\]

From (11), we can get
\[
\frac{\partial u_j(k)}{\partial c_{m,n}} = d_{m,n}
\]

And using Jocobian formulation from (19)
\[
\frac{\partial J_\varepsilon}{\partial u_j} \approx (K_p + K_i) \frac{\partial J_\varepsilon}{\partial u} = (K_p + K_i) \sum_{r=1}^{q} w_r \frac{\partial J_\varepsilon}{\partial x_r}
\]

Therefore, (23) and (24) are substituted into (22), and then the parameters \( c_{m,n} \) of fuzzy controller described by (11) can be adjusted using the following expression.
\[
\Delta c_{m,n}(k) = \alpha \eta (K_p + K_i) \sum_{r=1}^{q} w_r \frac{\partial J_\varepsilon}{\partial x_r}
\]

with \( m = j, j+1 \) and \( n = i,i+1 \).
III. DESIGN OF A FPGA-BASED NFC FOR PMLSM DRIVE

The internal architecture of the proposed FPGA-based motion control IC for PMLSM drive is shown in Fig. 4. The FPGA uses Altera Stratix II EP2S60 which has 48,352 ALUTs, maximum 718 user I/O pins, total 2,544,192 RAM bits, and a Nios II embedded processor is downloaded into FPGA to construct an SoPC environment. The motion control IC which comprises a Nios II embedded processor IP and a position control IP, is designed under the SoPC environment. The position control IP implemented by hardware is adopted to realize the function of a position NFC and speed P controller, a current controller and coordinate transformation (CCCT), SVPWM generation, QEP detection and transformation, ADC interface, etc. The sampling frequency of current control is designed with 16 kHz. The operating clock rate of the designed FPGA controller is 50 MHz and the frequency divider generates 50 MHz (Clk), 25 MHz (Clk-step), 12 kHz (Clk-cur) and 2 kHz (Clk-sp) clock to supply all module circuits of the position control IP.

An FSM is employed to model the NFC in position loop and P controller in speed loop which is shown in Fig. 5, which uses adders, multipliers and registers, etc. and manipulates 102 steps machine to carry out the overall computation. When 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 the NFC is highly complexity, the FSM can give a very adequate modeling and easily be described by VHDL. Furthermore, steps S0~S5 execute the computation of reference model output; steps S6~S9 are for the computation of velocity, position error and error change; steps S10~S13 execute the fuzzification and look-up fuzzy table; S14~S15 are for the defuzzification; S16~S27 are the computation of velocity and current command; S28~S91 describe the computation of RBF NN and Jacobian transformation; finally S92~S101 execute the tuning of fuzzy rule parameters. The operation of each step in Fig.5 can be completed within 40ns (25 MHz clock) in FPGA; therefore total 102 steps need a 4.08μs operation time.

![Diagram of motion control IC](image)

The Nios II embedded processor IP is depicted to perform the function of the position command in software, which includes main program and the interrupt service routine (ISR) by 2ms sampling interval. All programs are coded in the C programming language. Then, through the compiler and linker operation in the Nios II IDE (Integrated Development Environment), the execution code is produced and can be downloaded to the external Flash or SDRAM via JTAG interface. Finally, the FPGA utility of the motion control IC is evaluated. The circuit of a NFC uses 19,225 ALUTs resource and the overall circuits included a Nios II embedded processor IP (4,744 ALUTs and 45,824 RAM bits) as well as a position control IP (22,954 ALUTs and 301,056 RAM bits) in Fig.4, use 57.3% ALUTs resource and 13.6% RAM resource of Stratix II EP2S60.

IV. EXPERIMENTS AND 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: \( R_t = 27 \Omega \), \( L_s = L_q = 23.3 \text{ mH} \), \( K_t = 79.9 \text{ N/A} \). 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 5um 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.

The dynamic performance of PMLSM drive is evaluated while the NFC is applied in the position control loop of Fig. 1. 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 NFC are all realized by hardware in FPGA. The speed controller adopts a P controller with gain \( K_p = 1.1 \). The NFC is used in the position loop, the membership function and the initial fuzzy rule table are designed, and the PI gains are chosen by \( K_p = 0.3 \), \( K_i = 0.003 \). The transfer function of the reference model is a second order system with the natural frequency of 20 rad/s and damping ratio of 1. Figure 6 shows the position step responses of the mover using the FC and NFC when the position command is a 0.5Hz square wave with amplitude varied at 0~10mm and 25~35mm. The parameters \( c_{ij} \) of the fuzzy rule table are adequately selected at the 0kg external load condition, and
the step response shows a good dynamic response with a rising time of 0.2s, no overshoot and a near-zero steady state in Fig. 6(a). However, when 11 kg external load is added upon the mover and the same fuzzy control rule table and controller parameters are used, the position dynamic response worsens and exhibits a 19.5% overshoot in Fig. 6(b). It reveals that the dynamic performance of the PMLSM is affected by the external load on the mover. Accordingly, a NFC is adopted in Fig.1 to solve this problem. When the proposed NFC is used with learning rate being 0.05, the tracking results are highly improved and presented in Fig. 6(c). Initially, the mover of the PMLSM tracks the output of the reference model with overshoot. After one or two square wave commands, the position tracking error by using the NFC is only about 0.35 times of that obtained by using the FC in Fig.7. However, Fig. 7 reveals that the phase lag phenomenon using the FC is more serious than using the NFC in Fig.8. Therefore, the experimental results in Figs. 6 to 8 demonstrate that the proposed FPGA-based NFC for the PMLSM drive is effective and robust.

V. CONCLUSIONS

This study successfully presents a NFC for PMLSM drive based on FPGA technology. The work herein is summarized as follows. (1) The functionalities required to build a fully digital motion controller of PMLSM drive have been integrated in one FPGA chip. (2) The behavior of a NFC has been successfully described by VHDL. Finally, some experimental results are verified the effectiveness of the proposed controller system.

Fig. 5 State diagram of an FSM for describing the neural fuzzy controller in position loop and P controller in speed loop.
Fig. 6  Step response at 0~10mm to 25~35mm square save command under case of (a) FC without external load (b) FC with 11 Kg external load (c) NFC with 11Kg external load

Fig. 7 (a) Position frequency and (b) error response for a 1Hz-3Hz sinusoid input signal using FC under external load 11Kg

Fig. 8 (a) Position frequency and (b) error response for a 1Hz-3Hz sinusoid input signal using NFC under external load 11Kg

ACKNOWLEDGMENT
This work was supported by National Science Council of the R.O.C. under grant no. NSC 98-2221-E-218-050-MY2.

REFERENCES