International Conference on Control, Automation and Systems 2007
Oct. 17-20, 2007 in COEX, Seoul, Korea

An Open-loop Stepper Motor Driver Based on FPGA

Ngoc Quy Le[1] and Jae Wook Jeon[2]

[1] School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea
(Tel : +82-31-290-7237 ; E-mail: quy1001@ece.skku.ac.kr)
[2] School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea
(Tel : +82-31-290-7129; E-mail: jwjeon@yurim.skku.ac.kr)

Abstract: In motion control systems, stepper motors are widely used and the stepper motor driver is mainly implemented using a DSP (Digital Signal Processor). Using the DSP allows the control algorithm to be easily applied. Both coding and debugging processes can be rapidly performed. The DSP based driver's exhibit good performance; they are reliable, precise, and responsive. However, when mass production is needed, FPGA (Field-programmable Gate Array) based drivers are preferred since the FPGA based design is compatible with conversation to an ASIC (Application Specific Integrated Circuit). Previous studies proposed FPGA based drivers but the quality is still limited because of the difficulty of handling complex computations in the FPGA. Nowadays, with the improvement of FPGA techniques, it is possible to perform various computations in the FPGA so it can be used instead of the DSP in motor driver design. This paper presents an open loop driver based on the FPGA. This driver can perform stable, precise control and is responsive. The experimental results verify the performance of the control driver algorithm.

Keywords: Current control, FPGA, Open loop driver, Stepper motor.

1. INTRODUCTION

Recently, stepper motors have become popular in motion control systems because of their simplicity and reliability. Precise positioning operation is easier by using a stepper motor without a gear system. To control the stepper motor, the driver can apply an open loop control algorithm or closed loop algorithm. With applications where load torque is stable and operations are at low speed, an open loop driver is preferred since it needs a simple control algorithm and an encoder is not required.

In an open loop stepper motor driver, in order to perform position control, the current in the phase of the motor must be changed gradually. Therefore, the performance of the current controller in an open loop motor driver is important. There are 2 approaches to control current in stepper motor drivers. The duty cycle of switching pulse can be adjusted by using a DAC (digital to analog converter) and analog comparator [1-3]. Another approach is using an ADC (analog to digital converter) and a regular sample to adjust the duty cycle of the switching pulse [4-11]. With the first approach, the analog comparator will determine the state (on or off) of the switching pulse. So the control algorithm in the microprocessor is simple. However, with this method, the control algorithm cannot be changed, and the flexibility and the accuracy of the controller depend on analog components such as resistor, capacitor, op-amp, etc. With the second approach, the current controller samples the feedback current using an ADC, then the digital control law is applied to adjust the duty cycle of switching pulses. With this method, it is easy to change the parameters of the controller and apply digital filters. The accuracy of the current controller can be increased by using more bits for representing a fixed point number when executing the control algorithm. In this approach, the control algorithm and program become more complicated than the first method.

The control algorithm can be built on different kinds of microprocessors. When the open loop driver is implemented with a DSP in the C programming language, the coding and debugging is straightforward [5-9]. The performance of the DSP based driver is excellent. On the other hand, the stepper motor driver can be implemented with FPGA and VHDL (VHSIC Hardware Description Language) programming [1, 10-11]. With this method, the programming and debugging are more difficult. Because of this reason, the performance of the FPGA based driver is still limited. However, with the FPGA based driver with VHDL program it is easier in ASIC manufacturing.

This paper introduces a FPGA based open loop driver. With the current controller using a PI control algorithm and at high sampling frequency, the open loop driver can perform precise and fast positioning. The driver is built on FPGA XC2V8000 from Xilinx. The performance of the driver is checked by using an encoder (10000 positions per round), so the speed and position error can be monitored. The experiment results verify the performance of the control driver algorithm.

This paper presents FPGA technology, current controller, an open-loop driver, implementation, and experiment results, to verify the performance of the control driver algorithm.

2. FPGA BASED POWER CONVERTER AND OPEN LOOP DRIVER

2.1 Overview of open loop motor driver

Fig. 1 shows the block diagram of an open loop stepper motor driver. The open loop stepper motor driver contains a pulse counter, sine wave LUT (look up table) and power converter. To rotate the stepper motor,

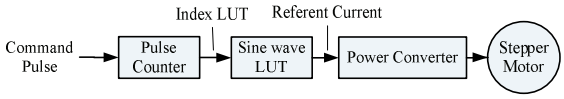

Fig. 1. Open loop stepper motor driver.

the currents in the phases of the motor are controlled to follow sine and cosine waves. In order to do this, the command pulse is counted to get the commanded position. And the commanded position is used as index of sine wave LUT. The output of sine wave LUT is the referent input to the power converter and the currents in the phases of the motor are controlled following the input referent. The subsequent sections of this paper will explain in detail the design of these blocks.

Fig. 2 is the block diagram of a power converter. There are H-bridge circuit, anti-aliasing filter, amplifier, ADC, PI controller and PWM (Pulse Width Modulation) generator. The H-bridge circuit is used for switching the current in phases of the motor. The ADC, anti-aliasing filter and amplifier are used to get the feedback current value for the PI controller. The PI controller will adjust the duty cycle of switching pulse according to the current error.

2.2 H-bridge circuit and current feedback device

The H-bridge circuit (Fig. 3) contains 4 MOSFETs and two sensing resistors [12] for each phase of the motor. The MOSFETs are controlled by an HIP4082 IC. With this H-bridge circuit, the currents flow through the phases of the motor as well as the sensing resistors R_{S1} and R_{S2}. The voltages V_{S1} and V_{S2} on the sensing resistors provide information about the feedback current by the relation $V_S = R_S I$. This voltage is amplified, filtered and sampled to produce a digital value. This digital value represents the value of the feedback current.

2.3 PI controller

A PI controller is used to adjust the duty cycle of the switching pulses. The input of the PI controller is the current error between the referent current and feedback current, and the output is the duty cycle of the switching pulses. To design a PI controller, first the model of the PWM converter is needed. The relation between the duty cycle and output current is described below.

The circuit H-bridge circuit (Fig. 3) contains 4 MOSFETs (S_1, S_2, S_3, S_4), DC voltage (V_d) and the phase of the motor (inductance L_M and resistance R_M). To simplify the analysis, the BEMF (Back Electromotive Force) in the phase of the motor is neglected. The four MOSFETs are treated as two pairs of switches (S_1, S_4) and (S_2, S_3). The power converter will set one of these two pairs of switches to the ON position. The period and duty cycle of the switching pulse are T and D, respectively. Fig. 4 shows the duty cycle (D) and voltage supplied to the phase of motor V_O.

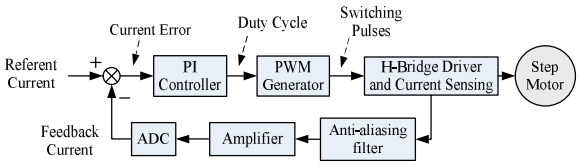

Fig. 2. Power converter.

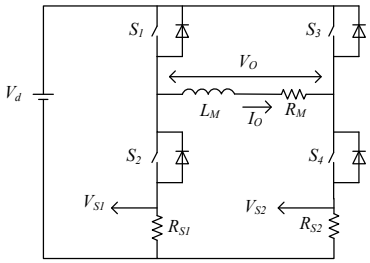

Fig. 3. H-bridge circuit.

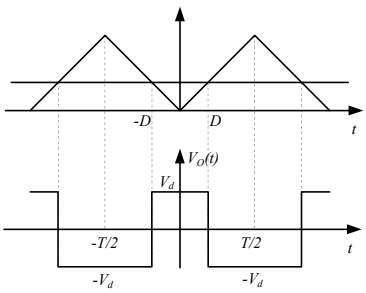

Fig. 4. Duty cycle and voltage on phase of motor.

In our current controller, the switching frequency is 40 kHz (T=25 uS) and the DC voltage V_d = 40 V. In one period [-T/2, T/2], when $-D \le t \le D$, pair (S_1, S_4) is on. When $-T/2 \le t < -D$ or $D < t \le T/2$, pair (S_2, S_3) is on. In Fig. 4, the output voltage $V_O = V_d$ when (S_1, S_4) is on and $V_O = -V_d$ when (S_2, S_3) is on. In this way, adjusting the value of D allows the average DC voltage output to be varied. The phase voltage can be summarized as (1).

$$V_O(t) = \begin{cases} -V_d \text{ when } -T/2 \le t < -D \\ V_d \text{ when } -D \le t \le D \\ -V_d \text{ when } D < t \le T/2 \end{cases} \quad (1)$$

Because V_O is a periodic function, the Fourier series of $V_O(t)$ is

$$V_O(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{2\pi nt}{T}\right) \quad (2)$$

Where

$$a_0 = \frac{2}{T}\int_{-T/2}^{T/2} V_O(t)dt = \frac{2V_d(4D-T)}{T}$$

$$a_n = \frac{2}{T}\int_{-T/2}^{T/2} V_O(t)\cos\left(\frac{2\pi nt}{T}\right)dt \quad (3)$$

$$b_n = \frac{2}{T}\int_{-T/2}^{T/2} V_O(t)\sin\left(\frac{2\pi nt}{T}\right)dt$$

In (3), the phase voltage $V_O(t)$ contains the DC term and harmonic terms. The frequency of the first order term of the harmonic is F=1/T=40 kHz. On the other hand, the relation between V_O and the phase current I_O is derived using the model of the RL circuit.

$$L_M = 7.35mH$$
$$R_M = 2\Omega \quad (4)$$
$$\rightarrow \frac{I_O}{V_O} = G_M = \frac{(1/R_M)}{(L_M/R_M)s+1} = \frac{0.5}{0.003675\ \text{s} + 1}.$$

The phase of the motor is considered as a low pass filter (4). Since the bandwidth is 271.5 Hz « 40 kHz, the harmonic terms of the voltage have no effect on the phase current of the motor. The phase current is determined by the DC term of V_O. Thus, from (2) and (3)

$$V_O = a_0/2 = (V_d 4D)/T - V_d = (V_d 4F)D - V_d \rightarrow \quad (5)$$
$$V_O = (40*4*40000)D - 40.$$

The duty cycle D is the output of the PI controller. In fact, the output of the PI controller is a digital value that is compared with the value of an up-down counter for the purpose of determining the switching moment of the pair of MOSFETs. The up-down counter represents the carrier wave, the max value of the counter (at T/2=25 uS/2=12.5 uS) is 1562. Therefore, to convert the value of D from integer to time unit (seconds), the scale gain is

$$K_C = (T/2)/1562 = 12.5\times10^{-6}/1562 \quad (6)$$

From (5) and (6)

$$V_O = (40*4*40000)K_C D - 40 \rightarrow$$
$$V_O = (40*4*40000)(12.5\times10^{-6}/1562)D - 40 \rightarrow \quad (7)$$
$$V_O = (80/1562)D - 40$$

Equation (7) shows a common result (8).

$$V_O = (80/1562)D - 40 \rightarrow$$
$$V_O = \begin{cases} 40 = V_d \text{ when } D = D_{\max} = 1562 \\ -40 = -V_d \text{ when } D = D_{\min} = 0 \end{cases} \quad (8)$$

From (4) and (8)

$$\frac{I_O}{V_O} = \frac{I_O}{(80/1562)D - 40} = \frac{0.5}{0.003675\ \text{s} + 1} \quad (9)$$
$$\rightarrow I_O = \frac{(40/1562)D - 20}{0.003675\ \text{s} + 1} \approx \frac{0.0256D - 20}{0.003675\ \text{s} + 1}$$

Because the constant does not affect the stability of the system, the phase current can be considered as (10), without the constant factor.

$$I_O = \frac{0.0256D}{0.003675\ \text{s} + 1} \quad (10)$$

Fig. 2 shows the power converter including the feedback loop. There is an amplifier, an ADC and an anti-aliasing filter with cut-off frequency F_C = 30 kHz. The transfer function of the anti aliasing filter is given by (11).

$$T_{FILTER} = 1/(2\pi*30000) \approx 0.00000530516$$
$$G_{FILTER} = \frac{1}{sT_{FILTER}+1} = \frac{1}{0.00000530516\ \text{s} + 1} \quad (11)$$

The value of the sensing resistor is R_S=0.05 Ω. The gain of the amplifier is K = 5. The ADC is a 12-bit signed ADC, with a maximum input of 1 V. At a phase current of 4 A, the input voltage for ADC is

$$V_{in} = I_O * R_S * K = 4A*0.05\Omega*5 = 1V \quad (12)$$

So the maximum phase current is 4 A. The output of ADC is maximum at $I_{fb} = 2^{11}$ -1= 2047 (12 bit signed number). The gain required to convert the current from amperes to a digital value is K_{ADC}=2047/4. From (10) and (11)

$$I_{fb} = \frac{0.0256D}{0.003675\ \text{s} + 1} \times \frac{2047}{4} \times \frac{1}{0.00000530516\ \text{s} + 1} \rightarrow \quad (13)$$
$$I_{fb} = \frac{13.1D}{1.95\text{x}10^{-8}\ \text{s}^2 + 0.00368\ \text{s} + 1}$$

The power converter is simplified as shown in Fig. 5. With the model of power converter as in (13), a PI controller is used for the current control loop. By using Matlab, gains for the PI controller are selected at the optimal value. The gains of the PI controller are given by (14).

$$K_P = 0.97$$
$$K_I = 0.002 \quad (14)$$
$$\rightarrow G_{PI} = \frac{(\text{s} + 0.002)0.97}{s}$$

Bode plot of the closed loop power converter is shown in Fig. 6.

The PI controller will adjust the duty cycle of the switching pulses according to the input current error. The PWM generator receives this duty cycle and produces the corresponding switching pulses to send to the HIP4082 IC, in order to control the MOSFETs. With this control scheme, the output current is controlled in such a way as to produce the desired currents.

2.4 Command pulse counter and Sine wave look-up table

The previous section discusses the power converter block. To rotate the motor, the current in phases of the motor must be controlled to follow sine and cosine waves. The desired position is sent to the driver by a series of command pulses. The frequency of these command pulses depends on the desired speed. The open loop driver uses a counter for counting the

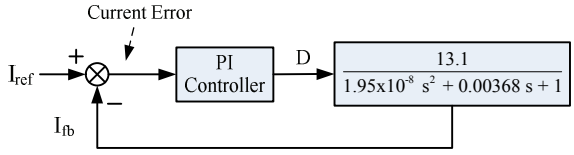

Fig. 5. Simplified block of a power converter.

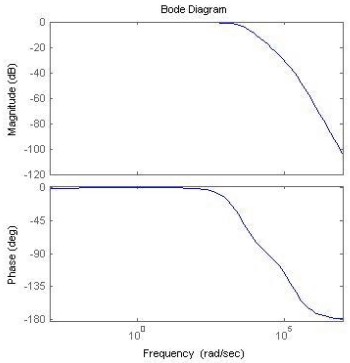

Fig. 6. Bode plot of PWM power converter.

command pulse. And the value of the counter is the index of the sine and cosine look-up table to produce the referent current of the power converter (Fig. 7).

3. IMPLEMENTATION AND EXPERIMENTAL RESULTS

This section shows the implementation of the control algorithms introduced in the previous sections in the FPGA-based stepper motor driver. The experimental results verify the performance of the control algorithm.

3.1 System configuration

Fig. 8 shows the structure of the experimental system. The FPGA based driver includes an EzM-56L stepper motor, an optical encoder which generates 2500 pulses per round (10000 positions per round), and a load which is an aluminum plate. The stepper motor, encoder and load are provided by FAS Technology Co., Ltd. The control algorithms are implemented on an FPGA module XC2V8000 of Xilinx and there are discrete devices including ADC for the other analog circuits.

Table 1 lists the electrical parameters of the driver. The sampling frequency of the current controller is 40 kHz, also the PWM switching frequency is 40 kHz, the supply voltage is 40 V, the phase inductance is 7.35 mH, and the phase resistance is 2 Ω, encoder resolution is 10000 positions per revolution.

Table 2 and 3 show the results of the implementation of the open loop driver on the FPGA device. Table IV is a summary of the utilized resources of the FPGA and Table V is a summary of the timing of the FPGA implementation. The maximum frequency of the input clock is about 26.5 MHz.

3.2 Experiments results

In this section, the performance of the driver is tested. First, an experiment was performed to check the performance of the current control loop. Second, the motor is controlled to follow a smooth motion.

Fig. 9 shows the referent current and feedback current in one phase of the motor. The referent current is a sine wave and the output current follows this sine wave. The maximum current is 2 A. The phase shift between the two currents is caused by the current control loop. This result shows that the current output can follow the desired current. This is because the position of the motor depends on the value of the current, when the output current is smooth, the precise position control can be achieved. In this FPGA based experimental system, the motor driver can perform open loop position control with 2 pulse error when the encoder feedback is 10000 positions per round.

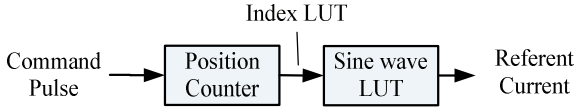

Fig. 7. Pulse counter and sine wave LUT.

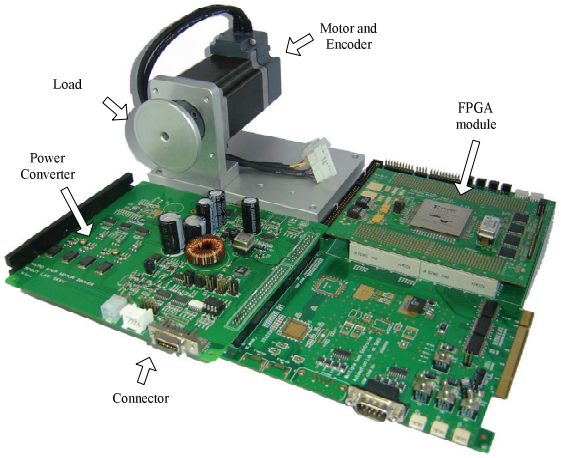

Fig. 8. Experimental system.

Table 1 Parameters of motor driver

Parameters	Value
Current controller sampling frequency	40 KHz
PWM switching frequency	40 KHz
Source voltage	40 V
Phase inductance	7.35 mH
Phase resistance	2 Ω
Encoder resolution	10000 position/revolution

Table 2 Device utilization summary

Resources	Quantity		
Selected Device	2v8000ff1152-4		
Number of Slices	12788	out of 46592	27%
Number of Slice Flip Flops	6721	out of 93184	7%
Number of 4 input LUTs	21682	out of 93184	23%
Number used as logic	21457		
Number used as Shift registers	225		
Number of IOs	176		
Number of bonded IOBs	173	out of 824	20%
Number of MULT18X18s	64	out of 168	38%

Table 3 Timing summary

Resources	Quantity
Minimum period	37.468ns (26.689MHz)
Minimum input arrival time before clock	13.688ns
Maximum output required time after clock	10.794ns
Maximum combinational path delay	8.175ns

Fig. 10 shows the currents in the 2 phases of the motor when the motor is controlled at a constant speed. Thus, the currents in the 2 phases follow a sine wave and shift in phase by an electric angle of 90 ° (this is a cosine wave). The maximum current is 2 A.

Fig. 11 shows the commanded speed and feedback speed of the motor. In this case, the maximum speed is 400rpm and acceleration and deceleration time is equal to 100ms. With a smooth motion, the commanded speed is changed following a smooth profile. The feedback speed shows that the motor can follow the command speed.

Fig. 12 shows the position error of motion. When the motor moves at max speed (400rpm) the position error is a maximum of about 45 micro-steps, with this max error, the rotor lags the command position by about 1 full-step, in this condition, generated torque is at maximum. And the maximum speed of the open loop

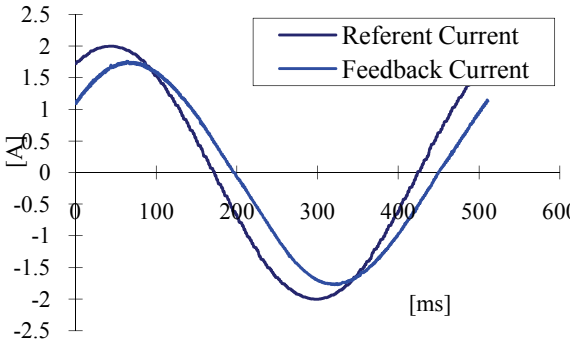

Fig. 9. Referent current and feedback current.

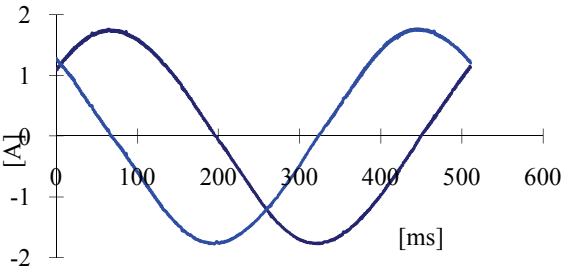

Fig. 10. Currents in 2 phases of motor.

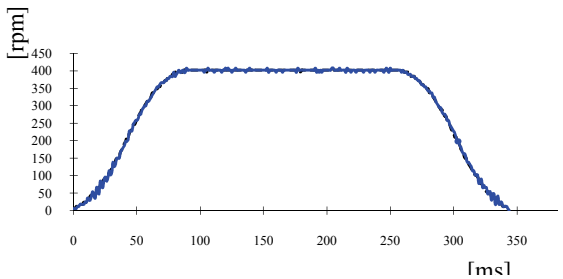

Fig. 11. Command speed and feedback speed.

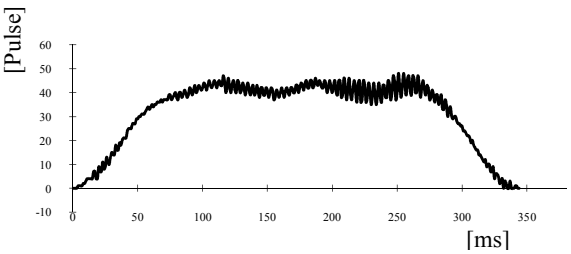

Fig. 12. Position error.

driver is almost reached. When the motor is moved, it generates BEMF, and the phase currents are reduced. Hence the maximum speed of the open loop driver is limited.

4. CONCLUSION

This paper presented an open loop stepper motor driver using a FPGA. The model of the current controller is described and gains of the PI controller are selected at optimal values. In this design, the FPGA is used to build a high performance open loop driver without using a DSP; furthermore it is compatible with ASICs and mass production methods. The experimental results verify the performance of the driver.

ACKNOWLEDEMENTS

This work was financially supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor (MOLAB), through the fostering project of the Lab of Excellency project.

REFERENCES

[1] Xiaodong Zhang, Junjun He, and Chunlei Sheng, "An approach of micro-stepping control for the step motors based on FPGA," *IEEE International Conference on Industrial Technology,* pp. 125 - 130, Dec. 2005.

[2] Muhammad H. Rashid, *Power Electronics Handbook*. Academic Press, 2001.

[3] D. Grahame Holmes and Thomas A. Lipo, *Pulse Width Modulation for Power Converters.* John Wiley & Sons, Inc., 2003.

[4] Blaabjerg, F.; Kjaer, P.C.; Rasmussen, P.O.; Cossar, C., "Improved digital current control methods in switched reluctance motor drives," *IEEE Transactions on Power Electronics*. Vol. 14, No. 3, pp. 563 - 572, May 1999.

[5] H. K. Bae and R. Krishnan, "A Study of Current Controllers and Development of a Novel Current Controller for High Performance SRM Drivers," *IEEE Industry Applications Conference,* vol. 1, pp. 68-75, Oct. 1996.

[6] P. Enjeti, J. F. Lindsay, P. D. Ziogas and M. H. Rashid, "New Current Scheme for PWM Inverters," *IEE Proceedings,* vol. 135, Pt. B, No. 4, Jul. 1988.

[7] Ali I. Maswood, "A PWM Voltage Source Inverter With PI Controller for Instantaneous Motor Current Control," *IEEE Proceedings of International Conference on Power Electronics and Drive Systems*, vol. 2, pp. 834-837, Feb. 1995.

[8] Halasz, S., "Analysis of pulse width modulation techniques for induction motor drives," *IEEE Conference Proceedings on Industrial Electronics,* pp. 200 - 204, Jun. 1993.

[9] Halasz, S.; Csonka, G.; Hassan, A.A.M.; Huu, B.T., "Analysis of the unipolar PWM techniques," *IEEE Electrotechnical Conference.* Vol. 1, pp. 353 - 356, May 1996.

[10] Toshio Takahashi and Jay Goetz, "Implementation of Complete AC Servo Control in a low cost FPGA and Subsequent ASSP Conversion," *IEEE Applied Power Electronics Conference and Exposition,* vol. 1, pp. 565-570, 2004.

[11] Xiaoyin Shao and Dong Sun, "Development of an FPGA-Based Motion Control ASIC for Robotic Manipulators," *IEEE Proceedings of The Sixth World Congress on Intelligent Control and Automation,* vol. 2, pp. 8221-8225, Jun. 2006.

[12] D. Rossi, and A. Cuomo, "Analog multiplex for sensing the magnitude and sense of the current through an h-bridge stage utilizing a single sensing resistance," *United States Patent*, Pat. Num. 4879641, Nov. 1988.