An FPGA Based Implementation of a Flexible Digital PID Controller For a Motion Control System

Somsubhra Ghosh[1], Ranjit Kumar Barai[2], Samar Bhattarcharya, Prarthana Bhattacharyya[3],
Shubhobrata Rudra[4], Arka Dutta[5], Rownick Pyne[6]

Dept. of Electrical Engineering
Jadavpur University
Kolkata, India
E-mail: [1]smghosh6@gmail.com, [2]ranjit.k.barai@gmail.com, [3]prarthana1223@gmail.com,
[4]shubhobrata.rudra@gmail.com, [5]arka088@gmail.com, [6]rwnckpn@gmail.com

Abstract— **Implementation of digital controllers in embedded environment suffers from the inherent problems associated with analog-digital signals interfacing in hard real-time, therefore, the control algorithms are invariantly subjected to approximations. This paper presents a novel technique for implementation of an efficient FPGA based digital Proportional-Integral-Derivative (PID) controller for the motion control of a permanent magnet DC motor. The implementation technique circumnavigates the problem of interfacing analog and digital systems in real-time. The controller is used in a speed control loop. The hardware implementation has been done on a Xilinx Spartan 3 FPGA chip. A novel technique has been adopted for the generation of the control input as a PWM signal for controlling the motor driver circuit and decoding the optical encoder data for using it for the speed feedback in the PID control loop. The VHDL algorithm for the proposed implementation has also been presented in this paper. A comparison of the experimental results with the Matlab® based simulation shows the effectiveness of the proposed method.**

Keywords- Digital PID controller, FPGA, Xilinx Spartan 3, permanent magnet DC motor, motion control.

I. INTRODUCTION

Digital control systems have become very widely popular due to the rapid advancement and consequent reduction in cost of digital computer and embedded systems technology. They are not susceptible to environmental noise and very easy to reconfigure. Introduction of the speed and position control loops in the motion control systems, to achieve fast response and high accuracy, is a very popular technique in industry. Therefore, digital control techniques are becoming very popular in motion control systems that involve various sources of noise and demand reconfiguration of the controller as and when a new job is to be performed with guaranteed control performance. Although there are numerous developments in advanced control theory, the Proportional-Integral-Derivative (PID) controller are still dominating in the motion control systems in the industry due to the well acquaintance of the operating personnel with PID controllers. Therefore, digital PID controllers are taking the place of their analog counterpart in the industrial motion control systems due to the widespread popularity of digital control as stated above.

The digital controllers available at present also suffer from quantization error, differential linearity error (DNE), integral linearity error (INE) and also the non-monotonicity due to the use of analog - digital converters. Thus the controllers, despite having an ingenious design, suffer from errors. However, the development of embedded systems in recent years provides a scope for circumnavigating this particular problem while leaving room for a much greater scale of flexibility, speed and cost minimization. Although, efficient use of the chip area and resources of an embedded system poses a great challenge while developing algorithms in embedded platforms for hard real-time applications, like control systems, digital signal processing, vision based sensing, and so on. Now, addressing the problem of implementing an efficient yet flexible and fast controller at the software level may not result into an optimal and dependable solution in hard real-time. Therefore, some hardware based solution at the chip development level is most suitable for this case where a dedicated digital circuit would be responsible for performing the arithmetic and logical operations as demanded by the application. However, development of such a digital circuit so as to perform the arithmetic and logical operations on them is quite difficult at the chip level due to the high level of complexities involved. The recent advancements in the area of Field Programmable Gate Array (FPGA) has provided many useful techniques and tools for the development of dedicated and reconfigurable hardware employing complex digital circuits at the chip level. Therefore, FPGA technology can be gainfully utilized in order to develop digital circuits so that the problem of realizing efficient, flexible and fast control systems could be solved at the hardware level. This investigation presents a novel technique to implement an efficient digital PID controller that may be customized and yet be flexible for future requirements. A number of works have been reported in the literature with an aim to achieve a realization of control systems with maximum efficiency, reduced latency and flexibility [1-10]. In [6], a two stage interpolation scheme has been used to reduce down the computational burden of NURBS interpolation and it employs a parallel operation of PC and FPGA system, whereas in [1], an adaptive backstepping approach has been taken to rule out the system uncertainties in the LIM, however, this technique is dependent on the FPGA architecture only. A similar work on PI controller has also been presented in [3], which uses an

978-1-4673-2907-1/13/$31.00 ©2013 IEEE

Altera Quartus II FPGA. [7] uses an FPGA based adaptive fuzzy controller for the purpose of speed control of PMSM drive. However our work on the implementation of a digital controller presents a flexible way of introducing the digital controllers in a motion control system that may be customized and yet be flexible for future requirements. Moreover, a novel technique has been adopted for the generation of the control input as a PWM signal for controlling the motor driver circuit and decoding the optical encoder data for using it for the speed feedback in the PID control loop. The VHDL algorithm for the proposed implementation has also been presented in this paper. A comparison of the experimental results with the Matlab® based simulation shows the effectiveness of the proposed method. To the best of the knowledge of the authors, such a development of FPGA based reconfigurable digital PID controller with custom built functional input and output ports for delivering the control input signal to the motor driver circuit and decoding optical encoder data for speed sensing on the same chip to circumvent the problems associated with the analog-digital signal interfacing has never been published in the literature.

The rest of the paper has been organized in the following way, section II presents the mathematical modeling and parameter identification of the plant, the permanent magnet DC (PMDC) motor, along with the description of the formulation of the control algorithm. Section III presents the system state modeling and the stability of the system. Sections IV and V present the implementation details and the experimental results of the system and finally the conclusions are drawn in section VI.

II. Control Algorithm Formulation

A. *Mathematicalmodel of the PMDC motor*

The armature voltage V_a has the following components:

$$I_a(s)[R_a + sL_a] + E_b(s) = V_a(s) \qquad (1)$$

Now for a PMDC motor,

$$T(s) = K_t I_a(s) \qquad (2)$$

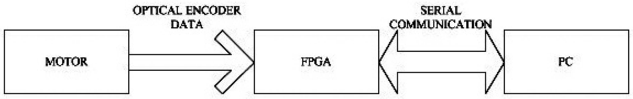


Fig. 1. Schematic diagram for the identification process.

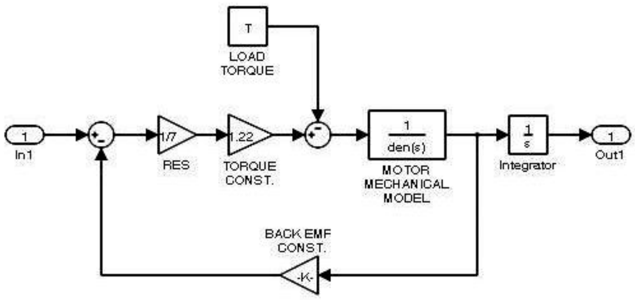


Fig. 2. Schematic diagram of the plant.

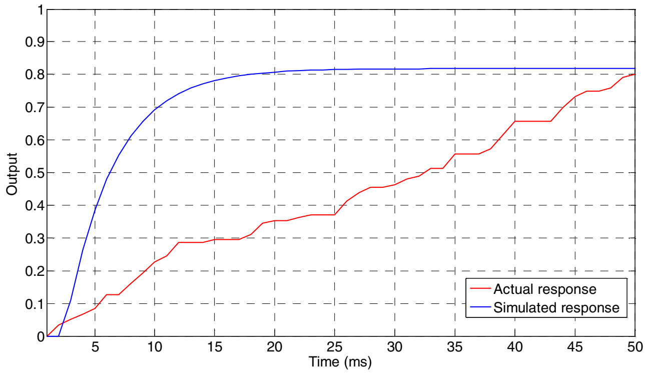


Fig. 3. Unit step response of the open loop system.

The back emf of the motor can be defined as

$$E_b(s) = K_b s\theta(s) \qquad (3)$$

The mechanical model of the motor can be expressed as:

$$J\frac{\partial^2\theta}{\partial t^2} + B\frac{d\theta}{dt} = \tau(t) \qquad (4)$$

Or,

$$T(s) = \theta(s)[Js + B]\cdot s \qquad (5)$$

So,

$$I_a(s) = \frac{s[Js + B]}{K_t}\cdot\theta(s) \qquad (6)$$

Substituting (6) & (3) in (2) gives us,

$$\frac{\theta(s)}{V_a(s)} = \frac{K_t}{s[(Js + B)(sL_a + R_a) + K_t K_b]} \qquad (7)$$

B. *Parameter Identification*

The circuit schematic for estimating the control parameters of the motor has been illustrated in Fig. 1. The response of the motor was tracked by using the FPGA, and the generated data was transferred to the PC by means of serial communication. The I/O data was used in the matlab parameter estimation toolbox GUI to generate the proper values of the parameters of the motor with the help of iteration. The result of the final iteration is given in Table-I. For the sake of simplicity in the implementation process, the motor model was linearized and the non-linearity due to the presence of the dry-friction, backlash and other model uncertainties were neglected. The plant model has been given in Fig. 2. The output of the motor optical encoder with a step response to the system has been given in Fig. 10. The motor speed data has been obtained from the rate of pulses of the optical encoder. A comparison of the actual and simulated step response from the identified model of the plant has been given in Fig. 3, which shows a rise time (T_r) of 1.75ms and a settling time (T_S) of 3.11ms.

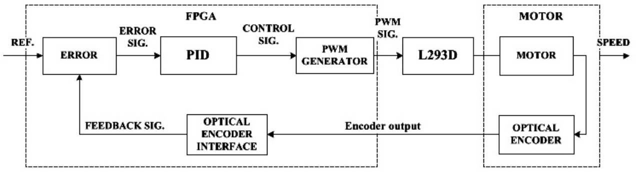


Fig. 4. Block level schematic of the complete system.

C. Design of the controller

The PID based controllers offer a very simple approach to control the system with its proportional, integral and derivative actions. Thus, it is used widely in a verity of different process control systems. These controllers are used in cascaded position and in velocity loops as well as in current loops and nested loops ([11],[12],[13],[14]), The PID controller can be used in cascaded position of velocity and current loops to achieve flexible and efficient dynamic and static characteristics of the system. The mathematical model of the controller has described as following, the output of the PID controller, with respect to an error of e(t) is,

$$u(t) = k_p e(t) + k_i \int e(t)dt + k_d \frac{de(t)}{dt} \qquad (8)$$

i.e.

$$U(s) = E(s)\left[K_p + \frac{K_i}{s} + K_d s\right] \qquad (9)$$

TABLE I. MOTOR PARAMETERS

Motor Parameters	Value
R_a	7 Ω
L_a	0.705 x 10-7 µH
J	0.001728 Nms2/rad
B	0.00443 Nms/rad
K_t	1.22 Nm/A
K_b	1.22 Vs/rad

There are various methods to tune the PID controller, here the proportional (K_p), integral (K_i) and derivative (K_d) gains are obtained from the Simulink PID tuner with the help of the system step response. For implementation purpose, the gains are rounded off to the nearest integer values while stabilizing the system and the resulting system was again tuned until the desired response was achieved. The block-level representation of the complete system has been described in Fig. 4. The controller description has been given in table 2. The controller has been designed with a trapezoidal integral and filter algorithm with a sampling frequency of 10 KHz. The step response of the tuned and optimized system has been given in Fig. 5, with a rise time (T_r) of 5.10ms and a settling time (T_s) of 11.4ms and a maximum overshoot of 9.24 %.

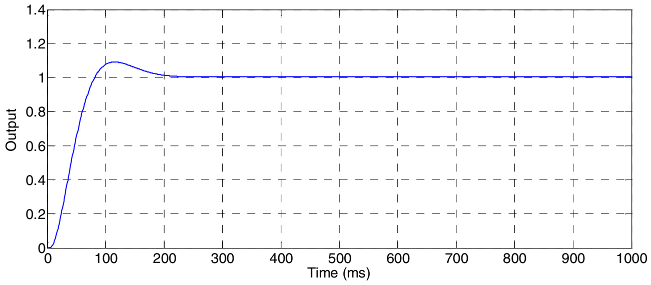


Fig. 5. Unit step response of the closed loop system.

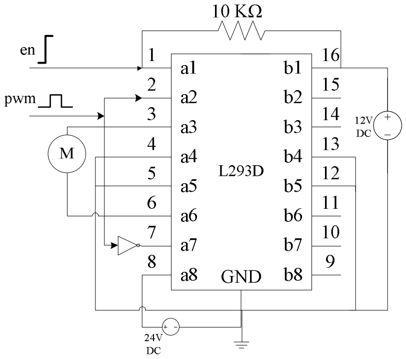


Fig. 6. L293D circuit schematic.

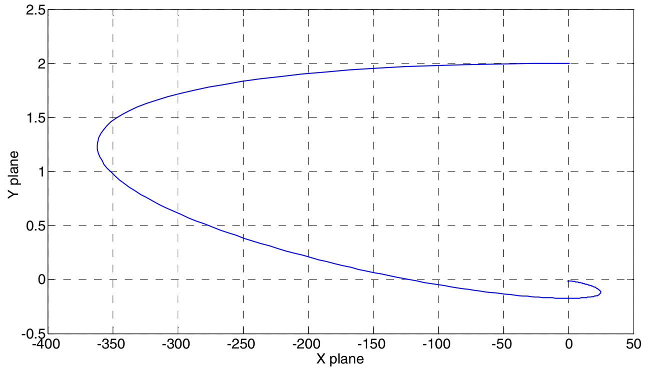


Fig. 7. Phase plane plot of the system.

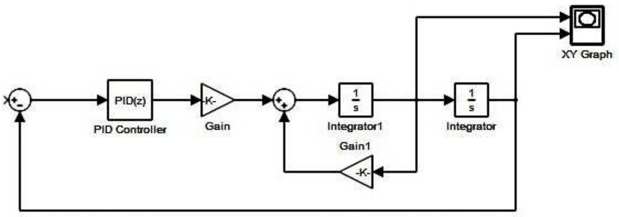
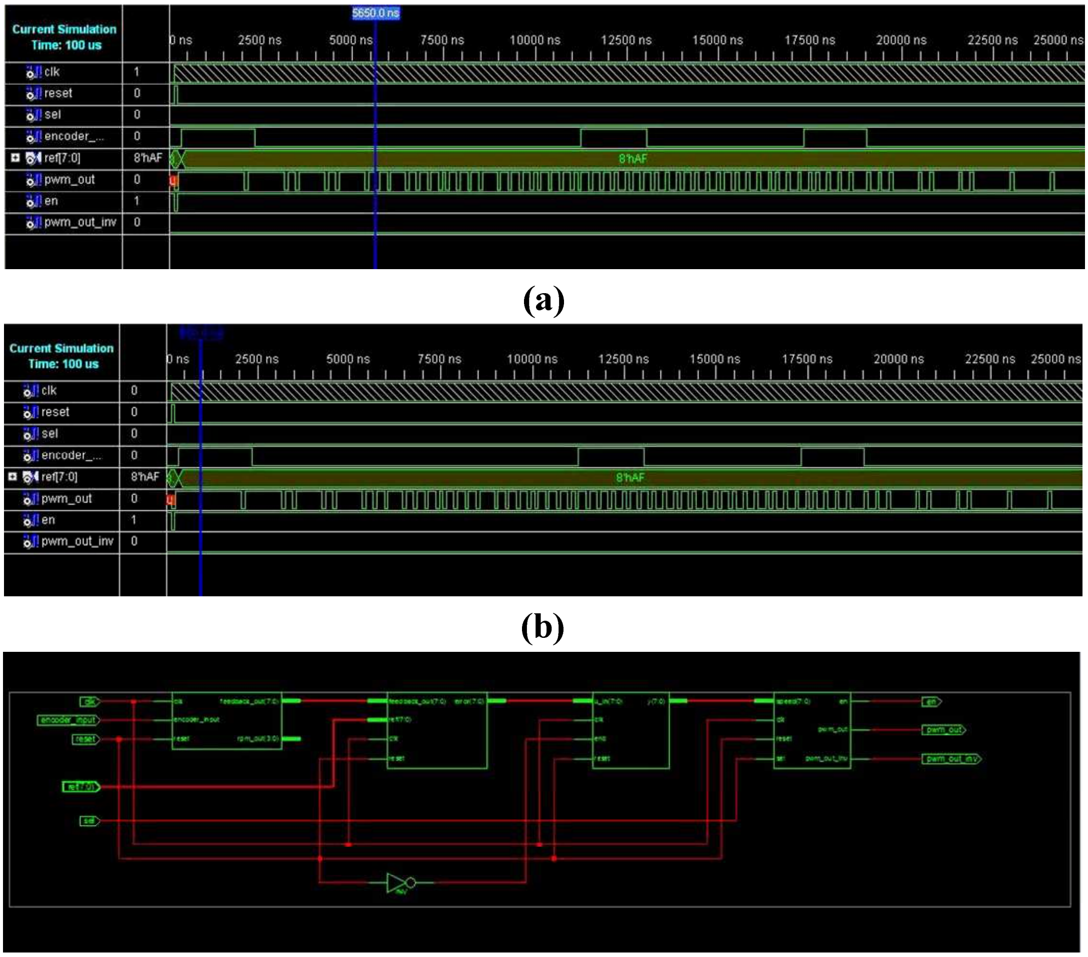


Fig. 8. System state model representation.



(a)

(b)

(c)

Fig. 9. System simulation results at Xilinx© ISE using the: (a) Behavioral simulation, (b) Post-route and synthesis simulation, (c) Technical schematic.

D. PMDC Motor Driver Circuit

Pulse Width Modulation is a very popular technique for delivering power to any circuit in a very precise and controlled manner. This design uses a PWM generator to power the L293D quadruple Half-H driver circuit, which, in-tern drives the motor. The necessary circuit diagram has been given in Fig. 6. The PWM is given a reference, and the duty cycle of the output signal is a function of the reference. Thus a reference of FF in hex corresponds to a 100% duty cycle of the output, and so on.

E. Error signal computation

The optical encoder mounted on the motor has a precision of 12°. So, if the motor RPM has a value of ‘x’ rotations per second, then the encoder will generate ‘30x’ amount of pulses in a second. The error signal is generated by calculating the number of encoder pulses from the feedback and converting it to an equivalent hexadecimal value of the reference to the PWM. Thus the PID is supplied with the difference between the PWM reference and the effective output.

TABLE II. CONTROLLER DESCRIPTION

Controller Gains	Value
K_p	380
K_d	816
K_i	2
N	526

III. SYSTEM STATE MODELLING

From (6) and (1), we can obtain the system state model by assigning variable x_1 to motor angular position θ, x_2 to angular velocity ω, and x_3 to armature current i_a, as described following,

$$\dot{x}_1 = x_2 \qquad (10)$$

$$\dot{x}_2 = ax_2 + bx_3 \qquad (11)$$

$$\dot{x}_3 = cx_3 + dx_2 + eu(t) \qquad (12)$$

$$y = x_2 \qquad (13)$$

The variables a, b, c, d, e are respectively,

$$-\frac{B}{J}, \frac{K_t}{J}, -\frac{R_a}{L_a}, -\frac{K_b}{L_a}, \frac{1}{L_a}$$

The approximated model of the system is obtained by ignoring the small value of the armature inductance, and is illustrated as following:

$$V_a = I_a R_a + K_b \omega \qquad (14)$$

$$T = J\dot{\omega} + B\omega \qquad (15)$$

$$T = K_t I_a \qquad (16)$$

$$E_b = K_b \omega \qquad (17)$$

So, the approximated state model is,

$$\dot{x}_1 = x_2 \qquad (18)$$

$$\dot{x}_2 = \hat{a} x_2 + \hat{b} u \qquad (19)$$

$$y = x_2 \qquad (20)$$

Where $\hat{a} = -\frac{1}{J}(B + \frac{K_t K_b}{R_a})$ and $\hat{b} = \frac{K_t}{JR_a}$.

TABLE III. DEVICE UTILIZATION

Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	202	1,536	13%
Number used as Flip Flops	98		
Number used as Latches	104		
Number of 4 input LUTs	162	1,536	10%
Logic Distribution			
Number of occupied Slices	171	768	22%
Number of Slices containing only related logic	171	171	100%
Number of Slices containing unrelated logic	0	171	0%
Total Number of 4 input LUTs	297	1,536	19%
Number used as logic	162		
Number used as a route-thru	135		
Number of bonded IOBs			
Number of bonded	15	124	12%
Number of MULT18X18s	3	4	75%
Number of BUFGMUXs	3	8	37%

The phase plane plot of the resulting system model has been given in Fig. 7. The system with some initial disturbances has attained stability, i.e. it returned to the origin, which denotes that the system is stable. The schematic state space model of the complete system has been presented in Fig. 8.

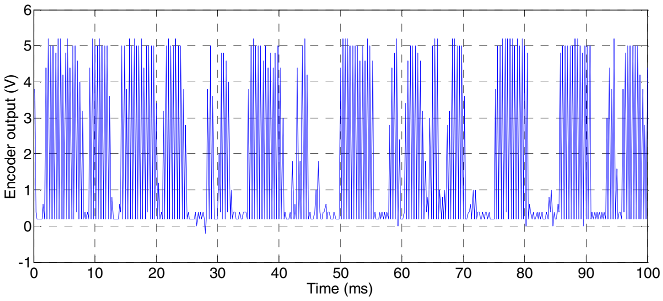


Fig. 10. Optical encoder output for a step input.

IV. IMPLEMENTATION DETAILS AND EXPERIMENTAL RESULTS

The control algorithm was implemented in a Xilinx Spartan 3 FPGA XC3S1500 chip. The average connection delay for the design is 15.194ns, of which, 67.4% was logic delay and 37.4% was due to route and placing. The chip utilization data has been presented in Table-III. The maximum frequency of this design is 65.814MHz. The controller, comparator, PWM generator and the encoder interfacing module has been implemented in the FPGA. The entire system has been simulated using the Xilinx ISE 9.2i. While the behavioral and post-place and route simulation describes the timing simulations of the signals, the schematic diagram shows the use of chip area and its utilization. Fig. 9.(a-b) shows the behavioural and the PAR simulation of the closed loop system, it shows the generation of the control output or the PWM pulses as per the given encoder input it represents that the output performance of the motion control system with the proposed FPGA based controller is capable enough to meet the expected performance as indicated by simulation. The chip level implementation and the logic utilization have been illustrated in Fig. 9. (b).

V. CONCLUSIONS

This paper has presented a novel way to implement digital controllers with a successful implementation of a digital PID controller. The design shows significant improvements over the present way of implementing digital controllers in Microcontroller Units in terms of latency, response, flexibility, and robustness. It can be extended to accommodate other advanced controllers that may also result into superior, reliable and flexible systems.

REFRENCE

[1] F.-J. Lin, L.-T. Teng, C.-K. Chang, "Adaptive Backstepping Control for Linear Induction Motor Drive Using FPGA," in Proceeding of IEEE conference on Industrial Electronics, pp. 1269-1274, 2006.

[2] Y.-Y. Tzou, T.-S. Kuo, "Design and Implementation of an FPGA-Based Motor Control IC," in Proceesings of IEEE IECON 1997, vol. 2, pp. 943-947.

[3] L. F. Castano, G. A. Osorio, "Design of a FPGA based position PI servo controller for a DC motor with dry friction," in Proceedings of IEEE VII Southern Conference on Programmable Logic, pp. 75-80, 2011.

[4] Y.-S. Kung, P.-G. Huang, C.-W. Chen, “Development of a SOPC for PMSM drives,” in Proceedings of 47th IEEE International Midwest Symposium on Circuits and Systems, vol. 2, pp. II-329-II-332, 2004.

[5] Y. Li, S. Zhuang, L. Zhang, “Development of an FPGA-Based Servo Controller for PMSM Drives,” in Proceedings of IEEE International Conference on Automation and Logistics, pp. 1398-1403, Aug. 2007.

[6] M.-T. Lin, H.-T. Yau, H.-W. Nien, M.-S. Tsai, “FPGA-Based Motion Controller with Real-time Look-ahade Function,” in Proceedings of IEEE international Conference on Advanced Intellegent Mechatronics, pp. 1406-1411, 2008.

[7] Y.-S. Kung, M.-H. Tsai, “FPGA-Based Speed Control IC for PMSM Drive With Adaptive Fuzzy Control, ” IEEE Trans. On Power Electronics, vol. 22, no. 6, pp. 2476-2486, Nov. 2007.

[8] J. Lamping, Z. Runjing, L. Zhian, “Realization of Position Tracking Syatem Based on FPGA,” in Proceedings of IEEE International Conference on Signal Processing, vol. 3, pp. 2588 – 2591, 2004.

[9] Y.-S Kung, K.-H. Tseng, T.-Y. Tai, “FPGA-based Servo Control IC for X-Y Table,” in Proceedings of IEEE Interantional Conference on Industrial Technology, pp. 2913-2918, 2006.

[10] T. Takahashi, J. Goetz, “Implementation of Complete AC Servo Control in a low cost FPGA and Subsequent ASSP Conversion,” in Proceeding s of IEEE Applied Power Electronics Conference and Exposition, vol. 1, pp. 564-570, 2004.

[11] H. Zhou, “Dc servo motor pid control in mobile robots with embedded dsp,” in Proc. IEEE Int. Conf. Intelligent Computation Technology and Automation, vol. 1, pp. 20 – 22, Oct. 2008.

[12] T. Y. Z. R. H. Xiaoxia, “Application of fpga in direct current motor servo system,” in Proc. IEEE 27 Chinese Control Conference, pp. 261–265, July 2008

[13] K. Y. H. G. D. W. L. Li, “Design of position servo-system of bldcm based on frequency domain method,” in Proc. IEEE Int. Conf. Electrical Machines and Systems, pp. 1612–1617, Oct. 2008.

[14] Z. R. D. Y. Y. Weiting, “Application of fuzzy-pi controller with feedforward control in direct current motor servo system,” in Proc. IEEE Int. Conf. Neural Networks and Brain, vol. 2, pp. 13–15, Oct. 2005