Optimization and Implementation of Intelligent PID Controller Based on FPGA

Kaiyu Wang[1,a], Zhenan Tang[1,b], Yongrui Zhang[1,c], Zhiming Song[1,d], Zhiheng Li[1,e] and Mengmeng Xi[1,f]
[1]Dalian University of Technology
Electronic Science and Technology Institute
Dalian, China
[a]wkaiyu@dlut.edu.cn, [b]tangza@dlut.edu.cn, [c]cyzwzyr@mail.dlut.edu.cn, [d]339255245@qq.com
[e]1285097055@qq.com, [f]610877221@qq.com

***Abstract*—To meet the real time and reliability requirements in the occasions containing high frequency disturbance and frequent changes of the parameters, this paper optimized the proportion and integral coefficients of fuzzy PID controller by combining the theory analysis with experts' experience. Simultaneously, the Fuzzy PID IP (Intellectual Property) Core based on Field Programmable Gate Array (FPGA) was implemented by using Verilog HDL and modular design method. Results prove the superiority of the optimization and the IP Core can be flexibly used in system design of SOPC (system on programmable chip).**

Keywords-Fuzzy-PID; FPGA; Knowledge Base; IP Core;

I. INTRODUCTION

With the ever-increasing complexity and performance demanding of control systems, the market of automation in industry is being developed towards following trends: integrated, networking, digitalization and intelligence [1].

To achieve real-time and efficient control performance, new or optimized controllers can be used. Moreover, implementation method is vital. A control system based on FPGA (Field Programmable Gate Array) can be seen as SOPC (system on programmable chip) [2]. Designers can combine one or several RISC processors with dedicated peripherals and computing hardware accelerators. And high frequency and parallel computing can be obtained by pure hardware. The researches of FPGA-based intelligent controllers or systems become increasingly active [3-5]. Applications of FPGA to industries has been transforming from auxiliary devices to foundation cores. This paper completed the optimization of traditional Fuzzy PID Controller and its hardware implementation in view of algorithm optimization and digitalization.

II. THE PRINCIPLE OF FUZZY-PID CONTROLLER

PID controller is widely used in various industrial applications for its simple structure, easy design method and stable performance. Fuzzy logic technology is an intelligent digital control method on basis of fuzzy mathematics, knowledge representation and fuzzy logic [6].

In terms of small scale and linear system, high precision performance can be obtained by PID algorithm. But it depends on the accurate models. Fuzzy technique is quite suitable for the control of MIMO (Multiple Input Multiple Output), strong coupling, time-varying, pure lag and nonlinear system or process. Even so, because of the defuzzification, a loss of control accuracy is unavoidable. And the inference rules will be immediately augmented for a more complicated system. Fig.1 shows the structure diagram of Fuzzy self-tuning PID controller.

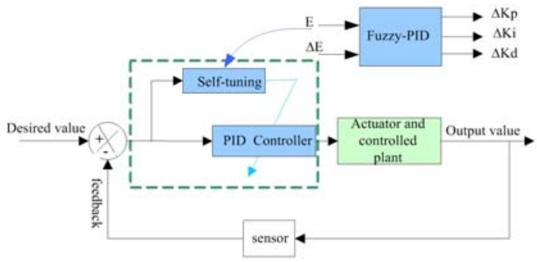

Fig.1 Structure diagram of Fuzzy self-turning PID

The core idea of this compound controller is online adjustment of those three PID parameters by fuzzy inference based on the relationships among e, Δe and the three variables. This gain-scheduling control can satisfy the high requirements of static index and steady state performance, making up the shortage of single controller.

III. OPTIMIZATION OF FUZZY-PID BASED ON MATLAB

A. Experiment system platform

Fig.2 shows the system platform based on Simulink and FIS Editor of MATLAB.

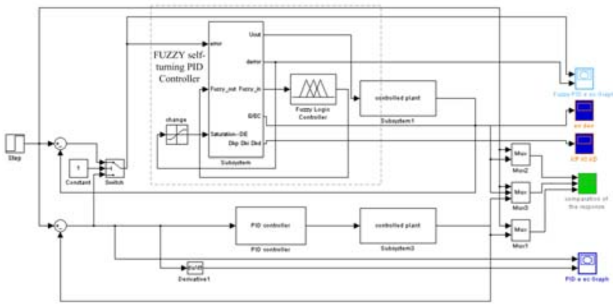

Fig.2 Test platform for the optimization of Fuzzy self- turning PID control rules based on the MATLAB

On this platform, We can easily set parameters. And with real-time curves of e, Δe and the three PID variables and the phase diagram of e and Δe, it is convenient to optimize the controllers.

B. Design and Optimization of Fuzzy-PID controller

Via making trade-offs of the precise and the complexity of implementation, input and output variables were divided into seven segments. The input fuzzy domain {-3, 3} and output are {-9, 9}. The map of physical and fuzzy area are realized and adjusted by quantitive and ratio factors. Mamdani fuzzy controller is chosen in this design and corresponding algorithms are shown in TABLE I.

TABLE I. MAMDANI FUZZY LOGIC ALGORITHMS OF THIS DESIGN

Add	***Or***	***Implication***	***Aggregation***	***Defuzzification***
Min	Max	Min	Max	Centroid

Fuzzy language variables are {PB, PM, PS, ZE, NS, NM, NB}, which respectively refer to {positive big, positive middle, positive small, zero, negative small, negative middle, negative big}. Memberships are triangular, except that the type of PB is S and NB is Z.

Following rules of the parameters adjustment are summarized based on experts' experience. (a) If |e| is large, a larger Kp and smaller Kd should be taken, to ensure fast response and avoid big overshoot. (b) If |e| is medium, in order to ensure stability and smaller overshoot, a smaller Kp, appropriate Ki and Kd should be chosen. (c) If |E| is small, to make the system has good steady-state performance, while avoiding the system oscillates around the set value, you should take a larger Kp and Ki, and take the appropriate Kd. In summary, we can draw fuzzy rule table, namely literature [7] and many other articles used as shown in Table II.

To test the effectiveness of inference rules, a typical second-order system was chosen as (1).

$$G(s) = \frac{25}{s^2 + 25s + 1} \tag{1}$$

Fig.3 shows the fuzzy adjustment curves of control parameters before optimization by a step input.

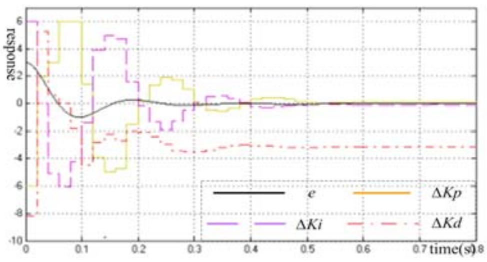

Fig.3 Adjustment rules of control parameters before Optimization

When e is large, the inference rules reduce the proportion value, yet increase integral coefficient. But when e is close to zero and continues to increase in the negative direction, the integral coefficient is still reduced. If the initial error is large, in order to avoid integral saturation and ensure a fast response speed, integral coefficient should be taken a smaller value and a bigger proportion value. When an overshoot appears with negative e and Δe, integral coefficient should be increased to reduce overshoot immediately. Therefore, the inference results do not meet the expectations. Because of the consistence of ΔK_d with requirements, this paper focuses on the optimization of *Kp* and *Ki* inference rules.

Inspired by literatures [8-9], a new fuzzy reasoning rule is obtained by testing the different rules and theoretical analysis. Firstly, the process of optimization is discussed. Fig.4 shows the response of e and Δe of a typical second-order system under a step input condition. According to the characteristic information, original inference rules are optimized by segmenting method.

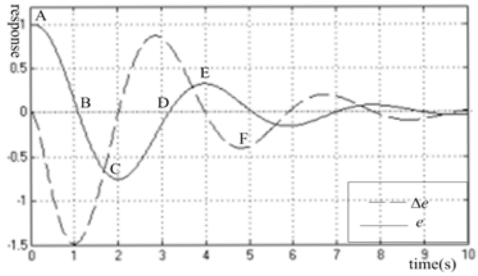

Fig.4 Response of e and Δe by a step input

In the segment AB, at first, a bigger Kp and smaller Ki should be taken to obtain a fast response and avoid integral saturation. In the intermediate stage, in order to ensure the stability, reduce Kp and Ki and increase Kd suitably. When it is close to B, to make sure the system has good steady performance while avoiding oscillations, take a larger Kp and Ki, and appropriate Kd. At BC, proportional and integral action should be increased, so that the system can returns to the steady state as soon as possible. At CD, to prevent the system from being adjusted backwards, reduce Kp and Ki; In DE segment, with the error increasing to a lesser extent, the system should be controlled by PI controller. Table III shows the optimized control rule table.

TABLE II. INFERENCE RULES TABLE OF OPTIMIZED KP AND KI OF FUZZY PID CONTROLLER

ΔKp/ΔKi		E						
		NB	*NM*	*NS*	*ZE*	*PS*	*PM*	*PB*
ΔE	*NB*	PB/PB	PB/PB	PM/PB	PM/PM	PS/NM	NS/NS	PS/NS
	NM	PB/PB	PB/PB	PM/PB	PS/PS	PM/NS	ZE/ZE	PM/ZE
	NS	PM/PM	PM/PM	PM/PM	PS/PS	PM/ZE	PS/ZE	PB/PS
	ZE	PM/PM	PS/PS	PS/PS	ZE/ZE	PS/PS	PB/PS	PB/PS
	PS	PM/PM	PS/PS	ZE/ZE	NS/NS	PM/PM	PB/PM	PB/PM
	PM	PS/PS	ZE/ZE	NS/NS	NM/NM	PB/PB	PB/PB	PB/PB
	PB	ZE/ZE	NS/NS	NM/NM	NM/NM	PB/PB	PB/PB	PB/PB

Fig.5 displays the fuzzy adjustment curves of control parameters after optimization, which meet the expectations.

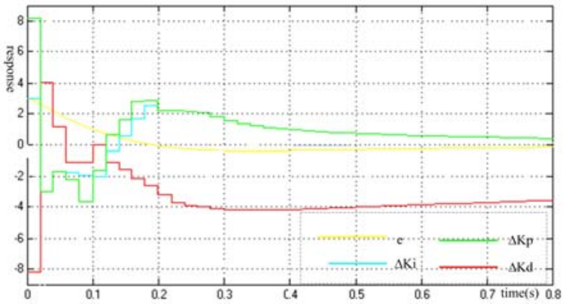

Fig.5 Adjustment rules of control parameters after optimization

C. Comparative analysis of the results

Based on the same controlled system and initial parameters, the responses of three controllers are showed in Fig.6.

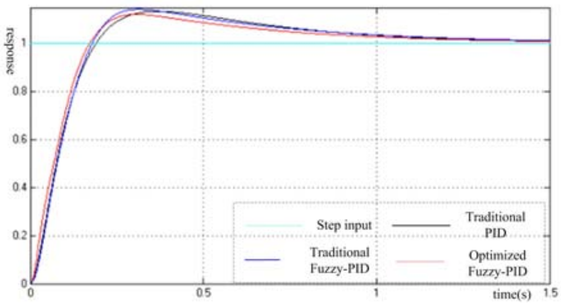

Fig.6 Different responses of the three controllers

Fig.7 shows the comparative analysis of three effects from the perspective of rising time, peak time, settling time and overshoot. In comparation with the traditional Fuzzy PID, the optimized one reduced the rise time and peak time by 8ms, the overshoot by 2.18% and the setting time($\pm 5\%$) by 105ms which is vital in the occasions containing high frequency disturbance and frequent changes of parameters.

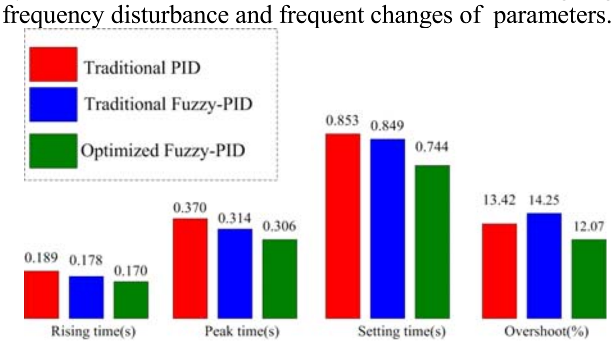

Fig.7 Comparation of the three controllers' responses

IV. Implementation And Verification Of Fuzzy Self-Tuning PID Controller Based On FPGA

The core part of this design is PID controller. There are several hardware implementation strategies such as serial and DA (distributed algorithm) methods [10-11]. Given the strong real-time performance of parallel structure and parallel computing capacity of FPGA, parallel structure was implemented in this design. Incremental PID algorithm can be simplified as (2),

$$\Delta u(k) = k_0 e(k) + k_1 e(k-1) + k_2 e(k-2) \tag{2}$$

where $k_0 = k_p + k_i + k_d, k_1 = -k_p - 2k_d, k_2 = k_d$.

Therefore, transforming the control parameters firstly, the calculation will be completed only with three multipliers and two adders. This approach simplifies the fuzzy reasoning process, accelerates the running speed of the controller and shortens the development cycle. Then, this paper, based on Top-Down design ideas, implemented the fuzzy adaptive PID controller IP (Intellectual Property) core internal hardware structure as shown in Fig.8.

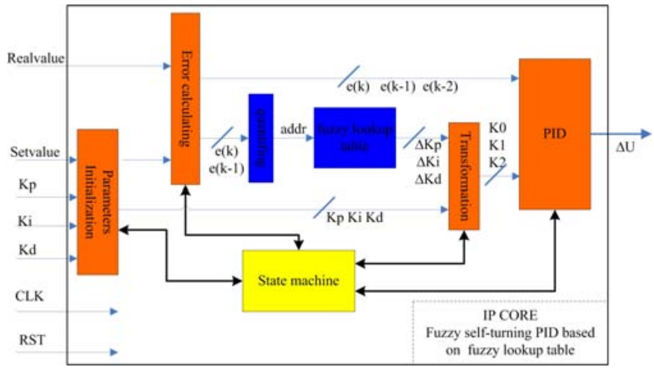

Fig.8 Internal structure of the IP core of Fuzzy PID controller

The IP core includes following sub-modules: parameters initialization, error calculating, quantization and fuzzy lookup table, parameter transfer, PID calculation and the state machine module. On the basis of modular design technique, this IP core was completed in Xilinx ISE 14.1 platform, by using Verilog HDL. Figure.9 is a state transition diagram of the state machine module.

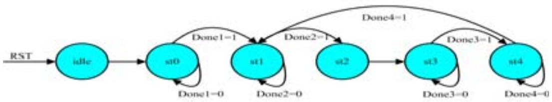

Fig.9 State transition diagram of the state machine

The state machine was designed by one-hot coding style, including a total of six states (idle, st0 ~ st4). Done signal marks the end of each state. Idle state is to reset the system when there is a falling edge of the reset signal. In the status of st0, parameters (Set value, *Kp, Ki and Kd*) initialization will be completed. State st1 finishes error calculating and outputs the latest three deviations (e (k), e (k-1) and e (k-2)). St2, a pure combinational logic unit, realizes fuzzy reasoning process, in which the incremental control parameters can be output from the lookup table, based on the address variable. Parameters conversion will be complemented in state st3. Then St4 completes incremental PID control algorithm, updates the output, then returns to state st1 and enters the next circle.

Fig.10 shows the functional simulation wave of this IP Core on the platform of Modelsim 10.1b.The results proves to be effective and meets the design idea.

V. CONCLUSIONS

This paper briefly introduces the development trends of industrial automation and the role of FPGA in this field. The Fuzzy self tuning PID controller was optimized and its IP Core was implemented based on FPGA. The simulation results verified its functional effectiveness.

With increasing demanding of higher performance and lower power consumption, it is necessary to use new technologies, in addition to intelligent control strategies. Due to the prominent advantages of FPGA in addressing the cost, power, speed and reliability, this device will be more widely used in the future.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of the Fundamental Research Funds, which is for the Central Universities of China.

REFERENCES

[1] LU N., “Innovation of FPGA platform environment for industrial applications,” Electronic technology design, pp. 19-20, 2013.

[2] Eric Monmasson, Lahoucine Idkhajine, and Marcian N.Cirstea, "FPGAs in Industrial Control Applications," IEEE Transactions on Industrial Informatics, vol.7, pp.224-243, May 2011.

[3] Leong, P.H.W., “Recent Trends in FPGA Architectures and Applications,” the Forth IEEE International Symposium on Electronic Design, Test & Application, Hong Kong, pp. 137-141, January 2008.

[4] Suratia, P., “FPGA based Fuzzy Logic Controller for plasma position control in ADITYA Tokamak,” Fusion Engineering and Design,.vol. 87, pp.1866-1871, 2012

[5] Huang, H., “Intelligent motion control for Omni directional mobile robots using ant colony optimization,” Applied Artificial Intelligence, vol. 27, pp. 151-169, 2013.

[6] Kyoung Kwan Ahn, and Bao Kha Nguyen, “Position Control of Shape Memory Alloy Actuators Using Self Tuning Fuzzy PID Controller,” International Journal of Control, Automation, and System, vol.4, pp.756-762, December 2006.

[7] Wei Xing, Liu-ting Ke, Pan Hai-peng, "Resistance Stove Temperature Control System based on Fuzzy Self-turning PID," IEEE International Conference on Control and Automation, ICCA, pp. 1123-1125, May 30 2007-June 1 2007.

[8] Basil Hamed, Moayed Al-Mobaied, “Fuzzy PID Controllers Using FPGA Technique for Real Time DC Motor Speed Control,” Intelligent Control and Automation, vol. 2, pp. 233-240, 2011.

[9] Wen Chen, Huimei Yuan, and Yan Wang, “Design and Implementation of Digital Fuzzy-PID Controller Based on FPGA,” the 4th IEEE Conference on Industrial Electronics and Applications, Xi’an, P. R. China, pp. 393-398, May 2009.

[10] Somsubhra Ghosh, and Ranjit Kumar Barai, “An FPGA Based Implementation of a Flexible Digital PID Controller for a Motion Control System,” International Conference on Computer Communication and Informatics, Coimbatore, INDIA, pp.1-6, 2013.

[11] Kravets, and P.I., “Methods of hardware and software realization of adaptive neural network PID controller on FPGA-chip,” Journal of Automation and Information Sciences, vol. 43, pp. 70-77, 2011.

TABLE III. INFERENCE RULES TABLE OF FUZZY PID BEFORE OPTIMIZATION

ΔKp/ΔKi/ΔKd		E						
		NB	*NM*	*NS*	*ZE*	*PS*	*PM*	*PB*
ΔE	*NB*	PB/NB/PS	PB/NB/PS	PM/NB/ZE	PM/NM/ZE	PS/NM/ZE	PS/ZE/PB	ZE/ZE/PB
	NM	PB/NB/NS	PB/NB/NS	PM/NM/NS	PM/NM/NS	PS/NS/ZE	ZE/ZE/NS	ZE/ZE/PM
	NS	PM/NM/NB	PM/NM/NB	PM/NS/NM	PS/NS/NS	ZE/ZE/ZE	NA/PS/PS	NM/PS/PM
	ZE	PM/NM/NB	PS/NS/NM	PS/NS/NM	ZE/ZE/NS	NS/PS/ZE	NM/PS/PS	NM/PM/PM
	PS	PS/NS NB	PS/NS/NM	ZE/ZE/NS	NS/PS/NS	NS/PS/ZE	NM/PM/PS	NM/PM/PS
	PM	ZE /ZE/NM	ZE/ZE/NS	NS/PS/NS	NM/PM/NS	NM/PM/ZE	NM/PB/PS	NB/PB/PS
	PB	ZE/ZE/PS	NS/NS/ZE	NS/PS/ZE	NM/PM/ZE	NM/PB/ZE	NB/PB/PB	NB/PB/PB

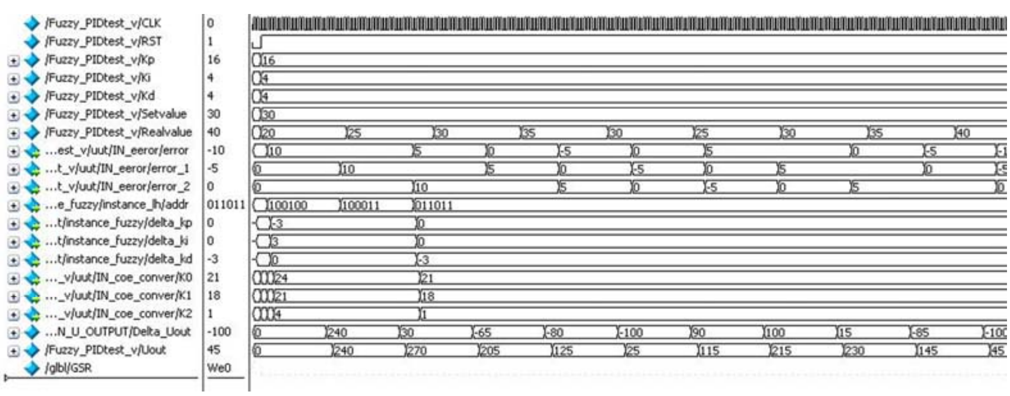

Fig.10 Waveform of functional simulation of the Fuzzy self-turning PID IP Core