Realization of Fuzzy PID Controller on FPGA for Source Measurement Unit

Cai-Sheng Fan
School of Information Engineering, Wuhan University of Technology, Wuhan, China
3035581811@qq.com

Xue-Dong Liu*
School of Information Engineering, Wuhan University of Technology, Wuhan, China
zxndp@126.com

Fei-Yu Li
School of Information Engineering Wuhan University of Technology, Wuhan, China
2724625332@qq.com

Yu-Xin Liang
School of Information Engineering Wuhan University of Technology, Wuhan, China
1498172972@qq.com

ABSTRACT

The classic Source Measurement Unit's (SMU: Source Measurement Unit) shortcomings include a sluggish transient response and a significant overshoot. This paper proposed an optimization method that made use of fuzzy PID controller on FPGA for SMU. With the aid of the software Matlab, we first analyzed the SMU's circuit diagram and obtained the system's transfer function. Next, we constructed the system simulation model adjusted by a fuzzy PID controller in Simulink. The initial P, I, and D parameters were obtained by the PID tuner, and they were adjusted by the fuzzy controller until they can achieve the optimal performance. Finally, we designed a Verilog program with the initial parameters to implement the fuzzy PID controller, in which the parameters were modified dynamically according to error and its increment to make sure they were optimal. We tested the controller using the HDL simulation tool Modelsim to ensure the simulation results matched those from Simulink. The results of simulation showed that the proposed method outperformed the conventional approach. In detail, our fuzzy PID controller can achieve shorter rising time, shorter setting time, and smaller overshoot than the NI SourceAdapt technology does.

KEYWORDS

Transient response, Fuzzy PID control, Simulink, Verilog, FPGA

ACM Reference Format:
Cai-Sheng Fan, Xue-Dong Liu, Fei-Yu Li, and Yu-Xin Liang. 2022. Realization of Fuzzy PID Controller on FPGA for Source Measurement Unit. In *5th International Conference on Information Technologies and Electrical Engineering (ICITEE 2022), November 04–06, 2022, Changsha, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3582935.3583010

*Corresponding author

ICITEE 2022, November 04–06, 2022, Changsha, China

ACM ISBN 978-1-4503-9680-6/22/11…$15.00
https://doi.org/10.1145/3582935.3583010

1 INTRODUCTION

The source meter is both a power supply and a measurement unit. The "source" is the voltage source or the current source, and the "meter" is the measurement meter [1]. "Source Meter" means a measuring instrument that provides accurate voltage or current as a four-quadrant voltage or current source while simultaneously measuring current or voltage values. Hence, it is widely applied in electronic product production testing, integrated circuit packaging testing, electronic measurement and other fields, so it is of great significance to optimize it [2].

The more mature technology available in the world is the NI SourceAdapt technology, which is different from the analog control loop, and its digital control loop can be configured entirely through software so that the optimal response to the load can be achieved by optimizing the control loop. However, there are still a lot of defects. For one thing, the system response time is still relatively slow. For another thing, the transient response exhibits oscillation and massive overshoot problems. Additionally, it cannot automatically change the control parameters in real-time.

In order to solve the abovementioned problems, this paper proposed realization of fuzzy PID controller for Source Measurement Unit system on FPGA. First of all, the circuit diagram of SMU was analyzed and the system transfer function was obtained by using MATLAB's symbolic mathematical toolbox to simplify the theoretical derivation. After that, we built the simulation model of the system in Simulink, whose original PI control model was replaced by the fuzzy PID control model. Finally, the fuzzy PID controller was implemented through Verilog, and was tested successfully in Modelsim. Through the above optimization methods, the PID parameters could be automatically improved in real-time, which greatly reduced the response time, obtained a more stable output with less error, eliminated overshoot and oscillation, and ensured the speed and stability of the system.

2 THE BASIC THEORY OF FUZZY CONTROL

fuzzy control is a controlling method that is based on fuzzy set theory, fuzzy language, and fuzzy logic. It uses human knowledge to control an object [3]. It is a type of nonlinear intelligent control that employs fuzzy mathematics in control systems. Its fundamental framework is shown in Figure 1.

When implementing the one-step fuzzy control theory, the sensor samples the precise value of the controlled variable and then

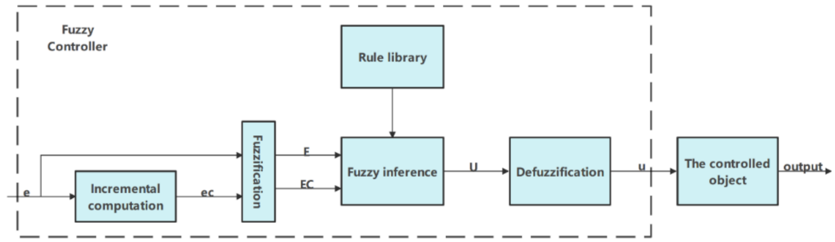

Figure 1: Fuzzy control schematic diagram

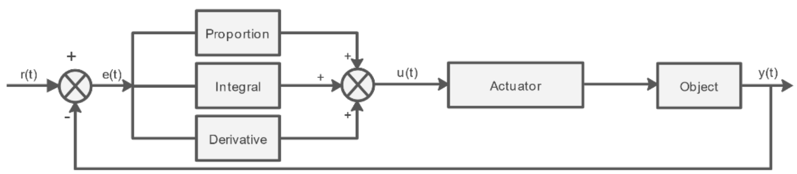

Figure 2: PID control schematic diagram

compares it to the provided value to get the error signals e and ec. In general, the error signals e and ec are chosen as the fuzzy controller's input variables, and the precise variable of e and ec is fuzzily quantized into a fuzzy variable E and EC [4]. They can be represented using the matching fuzzy language, making it simple to obtain the subset E and EC of the fuzzy language set of the error e and ec. Then, using the fuzzy control rule R of reasoning, E and EC reach a fuzzy decision and acquire the fuzzy control variable U [5]. To get the precise control u over the controlled object, the fuzzy variable U must be deblurred and converted into the exact value u. In one step, the accurate digital variable is translated into an accurate analog variable by the DA converter and supplied to the actuator to control the controlled object. The above process is then repeated to achieve fuzzy control of the controlled object.

3 THE BASIC THEORY OF PID CONTROL

The proportional-integral-derivative (PID) controller consists of a proportional unit (Proportional), an integral unit (Integral), and a derivative unit (Derivative) [6]. The gain of these three components may be modified to change its properties. Furthermore, the PID controller is best suited for systems that are essentially linear and whose dynamic properties do not vary over time. Figure 2 shows its schematic diagram.

The differential equation of PID control is as follows:

$$\mathrm{u}\,(t) = K_p\left[e\,(t) + \frac{1}{T_i}\int_o^t E\,(t)\,dt + T_d\frac{de\,(t)}{dt}\right] \tag{1}$$

In which $u(t)$ is the control variable, $e(t)$ is the difference between output and input, K_p is the proportional coefficient, T_i is the integral time, and T_d is the differential time.

The precise value of the controlled variable is gathered by the sensor in the PID control realization process, and this value is then compared with the supplied value to produce the error signal e [7]. Typically, it serves as the PID controller's input, and the linear combination of the proportional, differential, and integral operations of the e is employed to adjust the controlled object.

4 THE FRAMEWORK AND ALGORITHM OF FUZZY PID

4.1 Technical Framework

fuzzy PID control algorithm mainly includes two parts: fuzzy control module and PID control module. Figure 3 displayed the block diagram implemented on the FPGA.

Fuzzy control module: First of all, we must determine the domain of e, ec(error increment), K_p, K_i(integral coefficient), and K_d(differential coefficient), which will be divided into eight intervals. Then, we select a proper membership function to calculate the respective membership, and select the appropriate fuzzy rule table to obtain the membership of K_p, K_i and K_d through fuzzy reasoning according to the existing experience. Ultimately, we use the center of gravity method to deblur so as to obtain the precise values of K_p, K_i and K_d.

PID control module: According to the fuzzy control module, the proportional, integral and differential coefficients can be obtained, which will be transmitted to the PID controller. So, the final control

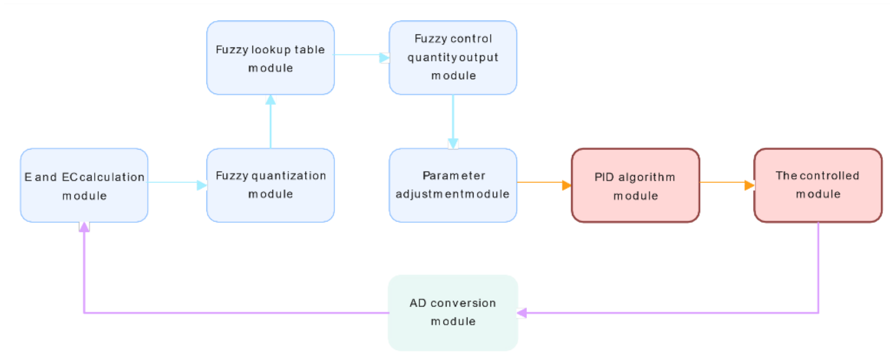

Figure 3: Fuzzy PID block diagram

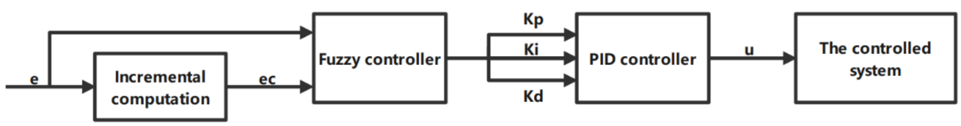

Figure 4: Fuzzy PID control schematic

variable can be obtained through the proportional, integral and derivative actions, thereby controlling the controlled system. This process can be implemented using an incremental PID algorithm.

4.2 Adaptive Fuzzy PID algorithm

4.2.1 Incremental PID Control Algorithm. Based on the formula (1) in the basic theory of PID control, the following equation can be obtained by performing Laplace transform on it.

$$\frac{U(s)}{E(s)} = P + I\frac{1}{s} + DS \tag{2}$$

In which P is equal to K_p, I is equal to K_i , and D is equal to K_d. Since differentiation is prone to noise interference, adding a first-order low-pass filter to the differential part leads to the following equation.

$$\frac{U(s)}{E(s)} = P + I\frac{1}{s} + D\frac{N}{1 + N\frac{1}{s}} \tag{3}$$

In which N is the filter coefficient, for a digital IC such as an FPGA, it can only compute the control variable's value based on the deviation value at each sampling moment. Therefore, in the digital control system, equation (3) has to be discretized firstly. For convenience, we use Euler method (s = $\frac{z-1}{T_s}$) to obtain the equation (4).

$$\frac{U(z)}{E(z)} = P + IT_s\frac{1}{z-1} + D\frac{N}{1 + NT_s\frac{1}{z-1}} \tag{4}$$

We transformed the transfer function from s domain to z domain and perform inverse Z transform to obtain the following difference equation.

$$u(k) = b_0 e(k) + b_1 e(k-1) + b_2 e(k-2) - a_1 u(k-1) - a_2 u(k-2) \tag{5}$$

In which $a_1 = NT_s - 2$, $a_2 = 1 - NT_s$, $b_0 = P + DN$, $b_1 = NT_sP - 2P - 2DN + IT_s$, $b_2 = P(1 - NT) + IT_s(NT_s - 1) + DN$.

4.2.2 Fuzzy PID Control Algorithm. The PID controller and fuzzy controller combine to form the fuzzy PID controller. The fuzzy controller includes three aspects that input fuzzification, fuzzy reasoning, and deblurring. The PID also covers three parts that proportion, differential and integration. Its algorithm schematic is shown in Figure 4.

First of all, we fuzzified the input variables e and ec, and set the input domain to [-6,6]. We selected seven linguistic variables to indicate the fuzzy subsets of the input variables in this universe: negative big [NB], negative medium [NM], negative small [NS], zero [ZO], positive small [PS], positive medium [PM], and positive large [PB]. Then the input is quantized by a linear function, and the quantization function is as follows [8].

$$\mathrm{f}(e) = \frac{6e}{\mathrm{MAX}(e) - \mathrm{MIN}(e)} \tag{6}$$

$$\mathrm{f}(ec) = \frac{6\mathrm{ec}}{\mathrm{MAX}(ec) - \mathrm{MIN}(ec)} \tag{7}$$

In which MAX presents the maximum value and MIN presents the minimum value. After quantizing, we selected the appropriate membership function to calculate the membership of the input on each language variable, thus blurring the input.

After overall consideration, we opted for the triangle membership function, and the fuzzy universe was shown in Figure 5.

After the input was blurred, the fuzzy rule table need to be established. Because the fuzzy rule table largely determined the effect of fuzzy control, the appropriate fuzzy rule table is established to adjust the three parameters K_p, K_i and K_d.

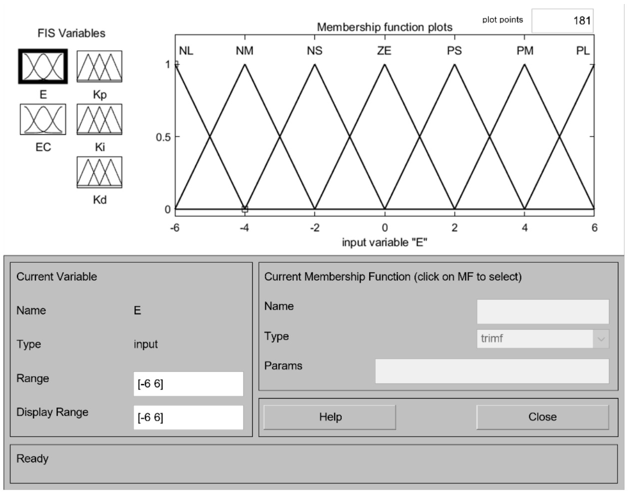

Figure 5: The fuzzy universe of *e*

- K_p fuzzy rule design: The choice of K_p value determines the response speed of the system. The response speed may be accelerated, and the steady-state deviation can be decreased by raising K_p. A high overshoot will result from a too-high K_p value, and the system may possibly become unstable. While decreasing K_p can lessen overshoot and increase stability, K_p that is too little will cause the response time as well as the adjustment period to be longer. In order to improve response speed, the initial adjustment should suitably use a higher K_p value. K_p should take a reduced value in the middle of the adjustment to reduce the system's overshoot and guarantee a certain response time. The K_p value is increased at a later stage of the adjustment process to lower the static difference and enhance control precision [9]. Based on the above principle, the fuzzy rules for defining K_p are displayed in Table 1.
- K_i fuzzy rule design: Integral control is primarily utilized to reduce the system's steady-state deviation. Therefore, the integral impact should be weaker, or perhaps nil, at the start of the adjustment process to avoid the integral being saturated. Integral effect should be relatively moderate in the adjustment's middle stage to prevent impairing stability. To lower the adjustment static difference, the integral effect should be improved at a later stage of the process. Based on the above analysis, the K_i fuzzy rules formulated are shown in Table 2.
- K_d fuzzy rule design: The differential coefficient's function is to alter the system's dynamic properties. The braking will advance in the adjustment process if K_d is very big, which will prolong the adjustment time. If the K_d number is too little. On the contrary, the braking will lag behind during the correction process, increasing overshoot. According to real process experience, the differential action should be enhanced at the initial stage of adjustment so as to reduce or even prevent overshoot. In the medium term, the K_d value should be properly small and should not fluctuate since the adjustment characteristics are more susceptible to changes in the K_d value. The K_d value should then be reduced in the latter stages of adjustment to lessen the braking impact of the controlled process, which can make up for the delay brought on by the high K_d value at the start of the adjustment process. Based on the above analysis, the fuzzy rules for K_d are shown in Table 3 [10].

According to the above rules, 49 fuzzy rules can be obtained, which can adjust the three parameters of the output in real-time. And Figure 6 showed a 3D view of the fuzzy rules.

In the end, we need to deblur. The fuzzy subset corresponding to the three parameters K_p, K_i, and K_d may be obtained using the preceding fuzzy rule table. The quantization value of each output variable is calculated using the center of gravity approach once the fuzzy quantization value corresponding to the fuzzy subset of the output parameters has been established. The following is its formula.

$$\mathrm{VO} = \frac{\sum_{\mathrm{i=0}}^{\mathrm{n}} M_i * F_i}{\sum_{\mathrm{i=0}}^{\mathrm{n}} M_i} \tag{8}$$

In which M_i is the membership degree and F_i is fuzzy quantification. If you are using a quantized value, you also need to revert to the actual value. Based on the obtained exact values of K_p, K_i and K_d, each coefficient in formula (5) can be calculated, which render the final control variable u can be solved by formula (5) to control the controlled system. As long as this cycle is repeated, the three PID controller parameters will be automatically adjusted and corrected to their optimal values, causing the final control variable u to tend toward zero and the system to attain the specified index and stabilize.

Table 1: K_p fuzzy rule design

K_p		*EC*						
		NB	NM	NS	ZO	PS	PM	PB
	NB	PB	PB	PM	PM	PS	ZO	ZO
	NM	PB	PB	PM	PS	PS	ZO	NS
	NS	PM	PM	PM	PS	ZO	NS	NS
E	ZO	PM	PM	PS	ZO	NS	NM	NM
	PS	PS	PS	ZO	NS	NS	NM	NM
	PM	PS	ZO	NS	NM	NM	NM	NB
	PB	ZO	ZO	NM	NM	NM	NB	NB

Table 2: K_i Fuzzy rule design

K_i					*EC*			
		NB	NM	NS	ZO	PS	PM	PB
E	NB	NB	NB	NM	NM	NS	ZO	ZO
	NM	NB	NB	NM	NS	NS	ZO	ZO
	NS	NB	NM	NS	NS	ZO	PS	PS
	ZO	NM	NM	NS	ZO	PS	PM	PM
	PS	NM	NS	ZO	PS	PS	PM	PB
	PM	ZO	ZO	PS	PS	PM	PB	PB
	PB	ZO	ZO	PS	PM	PM	PB	PB

Table 3: K_d fuzzy rule design

K_d					*EC*			
		NB	NM	NS	ZO	PS	PM	PB
E	NB	PS	NS	NB	NB	NB	NM	PS
	NM	PS	NS	NB	NM	NM	NS	ZO
	NS	ZO	NS	NM	NM	NS	NS	ZO
	ZO	ZO	NS	NS	NS	NS	NS	ZO
	PS	ZO	ZO	ZO	ZO	ZO	ZO	ZO
	PM	PB	NS	PS	PS	PS	PS	PB
	PB	PB	PM	PM	PM	PS	PS	PB

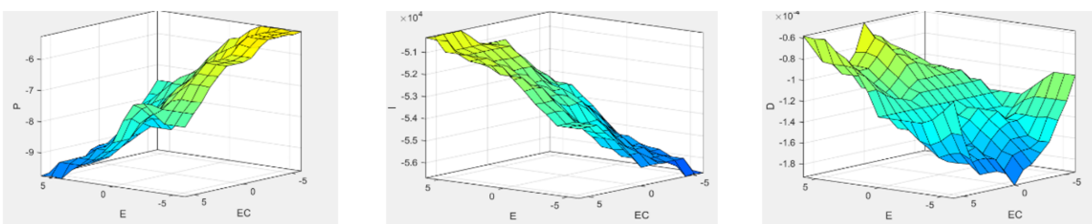

Figure 6: The 3D stereoscopic view of the blur control rule

4.2.3 Decimal multiplication based on Fixed-point Decimal Algorithm on FPGA. It is Because the FPGA cannot directly manipulate the floating-point number but the fixed-point number that we use the fixed-point decimal method to solve this problem that the coefficients of the difference equation are signed floating-point numbers. We are able to multiply the floating-point number x to be calculated by 2^n rounded and enlarged to an integer q which can perform the corresponding operation in the FPGA, and finally divide by 2^n to restore to the actual floating-point number. However, n depends on the actual required precision and operation speed, and the specific principle is as follows.

$$q = \left(x{*}2^{\mathrm{n}}\right) \tag{9}$$

$$x = \left(\frac{q}{2^n}\right) \tag{10}$$

where q (an integer) is the fixed-point decimal number of type Q_n and x is the actual number (a floating-point number) [11]. According to the above principle, the multiplication formula for fixed-point decimals can be obtained:

$$q_3 = \frac{q_1{*}q_2}{2^n} \tag{11}$$

Based on the above principle, in order to maintain high precision to ensure the accuracy of the operation result, we multiplied the coefficients of the difference equation by 2^{15} into Q_{15} fixed-point decimals, which were operated according to the equation (11) to solve the output of the difference equation. We divided it by 2^{15} to restore it to a floating-point number in the end.

5 SIMULATION AND ANALYSIS

5.1 Simulation results

5.1.1 Simulink simulation result. In Simulink, a fuzzy PID control system model was built, and the simulation model was shown in Figure 7.

Testing the output of the control variable u under its unit step response, and Figure 8 showed the discrete output of the control variable.

As can be seen from Figure 9, the three parameters of the PID be automatically adjusted to the optimal in real-time, causing the control variable u rapidly reached its peak and eventually tended

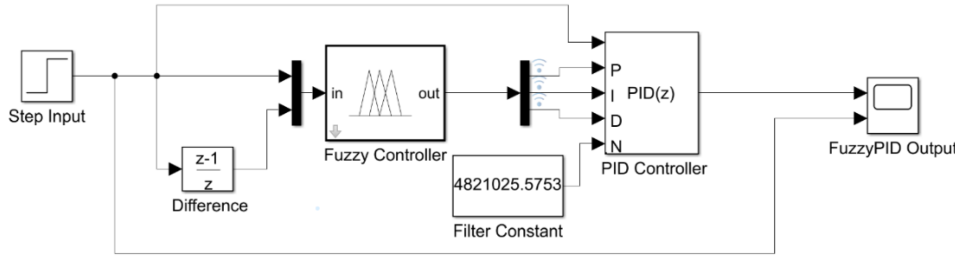

Figure 7: Simulation model in Simulink

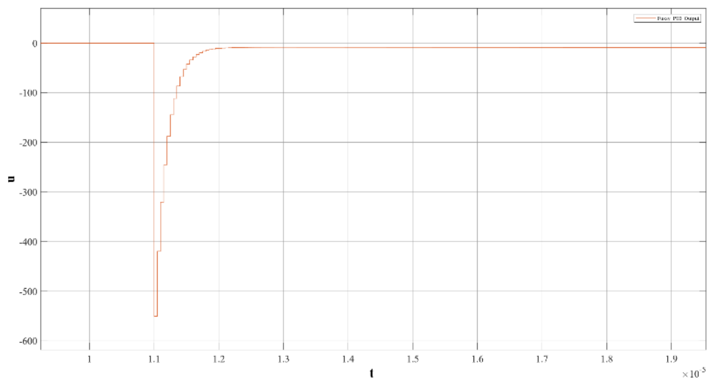

Figure 8: the output of the variable *u*

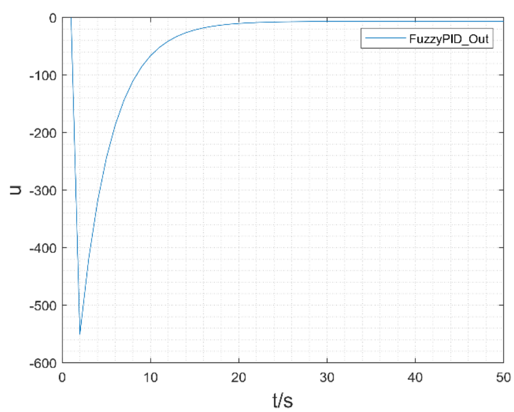

Figure 9: Matlab simulation results of fuzzy PID

to zero. Not only is the output overshoot less than 3%, but also the response time is greatly improved.

5.1.2 Matlab script simulation result. According to equation (5), Matlab script program was written to realize a fuzzy PID controller. After inputting the unit step signal, we drew and observed its output response, and its unit step was shown in Figure 9.

It is observed that the unit step response of the fuzzy PID controller implemented by the Matlab script program was exactly the same as the response simulated by the Simulink model. Therefore, the fuzzy PID controller could be described and implemented by HDL language.

5.1.3 Modelsim simulation result. Based on equation (5) and the fixed-point decimal method, it was achievable to write a Verilog program in Quartus II to realize a fuzzy PID controller. After the program was written well, the simulation debugging was carried out using Modelsim to test the output response under the input unit step response, and Figure 10 displayed the response curve.

Comparing the unit step response of the fuzzy PID implemented by the Verilog program with the simulation results of the Simulink model, the results of the two are exactly the same, which fully showed that the adaptive fuzzy PID controller could be realized on the FPGA to optimize the response of the SMU.

5.2 Results analysis

5.2.1 Technical feasibility analysis. By comparing the unit step response curve of the fuzzy PID simulation model in Simulink with the simulation results of the fuzzy PID program implemented by Verilog in the Modelsim, the comparison showed that their response results were exactly the same, which indicated the adaptive fuzzy PID controller could realize on FPGA.

5.2.2 The optimization of transient response. The system simulation models of PI control and Fuzzy PID control were built in Simulink, which were inputted unit step response at the same time. Figure 11 showed the unit step response curves of the two system simulation models. The comparison results revealed that the fuzzy

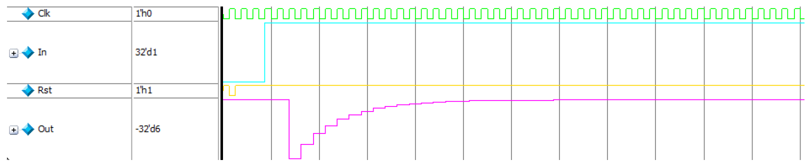

Figure 10: Modelsim simulation results of fuzzy PID

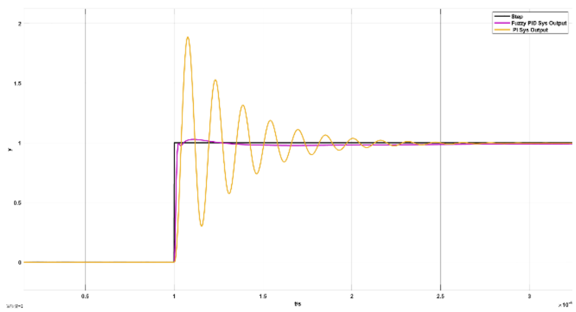

Figure 11: The unit step response for PI model and fuzzy PID model

Table 4: Response Parameters of Pi and Fuzzy Pid Controllers

Response parameters	Rise time	Setting time
PI controller	0.00349 seconds	0.0184 seconds
fuzzy PID controller	1.15e-06 seconds	0.000108 seconds

PID control was obviously better than the traditional PI control. Furthermore, not only could the fuzzy PID control automatically adjust the control parameters in real-time, but also reduce or even eliminate the overshoot and oscillation existing under the PI control. In a nutshell, the adaptive fuzzy PID control lessened the overshoot to a large extent, and made the system output more stable and accurate.

Additionally, comparing the output response of PI control and fuzzy PID control in Table 4, it was obvious that the response speed of fuzzy PID control was significantly faster than PI control, fully demonstrating that fuzzy PID control greatly decreased the response time compared with PI control, improved the speed of system transient response, and considerably optimized the transient response and robustness of the SMU [12].

6 CONCLUSION

In view of the certain defects in the PI control used in the NI Source Adapt technology currently used internationally, this paper innovatively proposed realization of fuzzy PID controller for Source Measurement Unit system on FPGA. First of all, analyzing the circuit diagram of the SMU and using MATLAB's Symbolic Math Toolbox to calculate as well as simplify the transfer function of the system. Secondly, using the fuzzy Control Toolbox in Matlab to design a fuzzy controller, and then constructing a system simulation model in Simulink adjusted by the fuzzy PID controller model. After the simulation debug successfully, writing the Verilog program of the fuzzy PID controller in Quartus II and using the HDL simulation tool Modelsim to simulate and debug the program. Ultimately, we can realize the SMU of fuzzy PID control on the FPGA.

ACKNOWLEDGMENTS

This paper was supported by Wuhan University of Technology Provincial College Students Innovation and Entrepreneur ship Training Project under Grant S202210497311.

REFERENCES

[1] C. Xiangjun, "An Electrical Parameter Measurement Method for IGBT Based on Source Measurement Unit[J]," Instrumentation Users, 2016, 23(3):4.
[2] Y. Can, "Research on key technologies of source Measurement unit based on FPGA[D]," Guangdong University of Technology.
[3] Xu Chongjie. "Research on some problems of automobile bumper based on fuzzy control [D]." Sichuan: University of Electronic Science and Technology, 2021.
[4] Juntong Yun, Ying Sun, Cuiqiao Li, Du Jiang, Bo Tao, Gongfa Li, Ying Liu, Baojia Chen, Xiliang Tong, Manman Xu, Self-adjusting force/bit blending control based on quantitative factor-scale factor fuzzy-PID bit control, Alexandria Engineering Journal, 2022,61(06):4389-4397.
[5] Wang Meiqi. "Design of One-Dimensional Fuzzy PID Controller [J]." Information Technology and Informatization, 2021(3):71-72. DOI: 10.3969/j.issn.1672-9528.2021.03.022.
[6] Feng Yanhong, Tang Lan, Chen Linbin, *et al.* "Comparison of fuzzy PID and PID control strategies based on ESP [J]." Automotive Practical Technology, 2022, 47(17): 109-112. DOI: 10.16638/j.cnki.1671-7988.2022 .017.021.
[7] Xi Fangyuan, Huang Zhousong, Li Yan, Hu Guohua. "Improved Incremental PID Algorithm Based on FPGA DC Motor Control [J]." Journal of Xiangnan University, 2022,43(05):24-29.
[8] Yu Changshun, Yuan Ruibo. "Adaptive fuzzy PID control of electro-hydraulic position servo system based on AMESim-Simulink [J]." Agricultural Equipment and Vehicle Engineering, 2022,60(08):158-161.
[9] Mann G K I, Hu B G, Gosine R G. Analysis of direct action fuzzy PID controller structures[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1999, 29(3): 371-388.
[10] Qing Jinhui, Hu Huangshui, Wang Hongzhi. Design of BLDCM fuzzy PID controller based on FPGA [J]. Journal of Changchun University of Technology, 2021, 42(2): 168-174. DOI: 10.15923/j.cnki.cn22-1382/t .2021.2.12.
[11] Y. Lan, "Research and implementation of vehicle vibration signal processing system based on FPGA[D]," Chang' an University, 2012.
[12] Q. Xiaoxiao, and Z. Jing, "Pressure Control of Flash Tank Based on Grey Predictive Fuzzy PID Algorithm[J]," Chemical Automation and Instrumentation, 2019(3):5.