Fuzzy Logic PID controller based on FPGA

V. Tipsuwanporn, T. Runghimmawan, V. Krongratana, T. Suesut and P. Jitnaknan

Department of Instrumentation Engineering, Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
(Tel: 66 – 2326 – 7347: Ext. 102; E – mail: ktvittay@kmitl.ac.th)

Abstract: Recently technologies have created new principle and theory but the PID control system remains its popularity as the PID controller contains simple structure, including maintenance and parameter adjustment being so simple. Thus, this paper proposes auto tune PID by fuzzy logic controller based on FPGA which to achieve real time and small size circuit board. The digital PID controller design to consist of analog to digital converter which use chip TDA8763AM/3 (10 bit high-speed low power ADC), digital to analog converter which use two chip DAC08 (8 bit digital to analog converters) and fuzzy logic tune digital PID processor embedded on chip FPGA XC2S50-5tq-144. The digital PID processor was designed by fundamental PID equation which architectures including multiplier, adder, subtracter and some other logic gate. The fuzzy logic tune digital PID was designed by look up table (LUT) method which data storage into ROM refer from trial and error process. The digital PID processor verified behavior by the application program ModelSimXE. The result of simulation when input is units step and vary controller gain (K_p, K_i and K_d) are similarity with theory of PID and maximum execution time is 150 *ns/action* at frequency are 30 *MHz*. The fuzzy logic tune digital PID controller based on FPGA was verified by control model of level control system which can control level into model are correctly and rapidly. Finally, this design use small size circuit board and very faster than computer and microcontroller.

Keywords: FPGA, PID, Fuzzy logic, Process.

1. INTRODUCTION

The fuzzy - PID controller was designed by consist of analog to digital 10 bit converter, digital 12 bit to analog converter and digital fuzzy - PID processor. In this paper not present designing analog to digital 10 bit converter and digital 12 bit to analog converter but will be show its schematic circuit. The fuzzy - PID processor to consist of the digital PID processor will be use algorithm of digital PID-equation, which come from fundamental PID equation and the fuzzy logic processor that have 2 inputs, 8 membership function in each fuzzy input and 3 output for PID controller gain.

Section II explains theory PID and fuzzy logic. The next section discusses designing the fuzzy – PID processor based on chip FPGA by VHDL. Section IV show the verification and implementation of the fuzzy - PID controller. Final section is summary.

2. THEORY

2.1 Principle and theory of digital PID controller

PID controller has a fundamental equation [1].

$$V_o(t) = K_p\left[e(t) + \frac{1}{T_i}\int e(t)dt + T_d\frac{de(t)}{dt}\right] \quad (1)$$

And

$$e(t) = sp(t) - pv(t) \quad (2)$$

When *Vo(t)* = controller output.
e(t) = error.
st(t) = set point.
pv(t) = process output.
Kp = controller gain.
Ti = integral time (sec.).
Td = derivative time (sec.).

From Eq. (1), change to digital PID controller by discrete differential equation following.

$$V_o = V_{o_{n-1}} + K_p\left[(e_n - e_{n-1}) + \frac{\Delta T}{T_i}e_n + \frac{T_d}{\Delta T}(e_n - 2e_{n-1} + e_{n-2})\right] \quad (3)$$

When ΔT = the sampling period.
$V_{o_{n-1}}$ = Controller output at the *n*th sampling instant.
e_n = error at the *n*th sampling instant.

Define $K_i = \frac{K_p\Delta T}{T_i}$

$K_d = \frac{K_pT_d}{\Delta T}$

So that digital PID equation show below.

$$V_o = V_{o_{n-1}} + (K_p + K_i + K_d)e_n - (K_p + 2K_d)e_{n-1} + K_de_{n-2} \quad (4)$$

2.2 Principle and theory of fuzzy logic controller

The fuzzy logic, unlike conventional logic system, is able to model inaccurate or imprecise models. Fuzzy logic can handle problems that may be too complex for conventional modeling techniques. The fuzzy logic approach offers a simpler, quicker and more reliable solution that is clear advantages over conventional techniques. Fuzzy logic may be viewed as form of set theory [2]. The universe of discourse **U** is a collection of objects under consideration. The fuzzy subset **A** in **U** is characterized by a membership function μ_A which takes the values in the interval [0,1], with 0 representing no membership and 1 representing full membership. In designing a fuzzy system, universe of discourse and membership functions of inputs and outputs (linguistic variables) are first determined. To allow fuzzy approximate reasoning, the fuzzy inference mechanism based on the generalized modus ponens (GMP) shown below is usually adopted:

Promise 1: if input *x* is **A** then output *y* is **B**
Premise 2: input *x* is **A′**
Consequence: output *y* is **B′**

Where **A**, **B**, **A′** and **B′** are fuzzy sets. Premise 1 (the "if…then" part) can be represented by fuzzy relation R which can be derived from different fuzzy implication functions. The one based on the "min" implication function requires the following computation:

$\mu_R(u,v) = \min(\mu_A(u), \mu_B(v));\quad u \in \mathbf{U}, v \in \mathbf{V}$.

Where $\mu_R(u,v)$ is the membership function of R. With a specific input, the fuzzy inference is completed through the fuzzy composition show below:

$\mathbf{B'} = \mathbf{A'} \circ \mathbf{R}$

The fuzzy composition based on the max – min operator requires the flowing computation:

$\mu_B(v) = \max \min(\mu_A(u), \mu_R(u,v))$

To obtain a crisp output from the fuzzy inference engine, a defuzzification algorithm is needed. This design uses the center of gravity (COG) algorithm show below:

$$\mathrm{Zo} = \frac{\sum \mu_i z_i}{\sum \mu_i} \tag{5}$$

Where μ_i and z_i are the weight (value) and center (in the universe of discourse), respectively, of an output MF which is in effect; z_o is crisp output of the fuzzy inference engine.

2.3 Fuzzy gain scheduling

According to the methodology proposed by Zhao et al. [3], the three current PID parameters are determined as follows

$$K_p = (K_{p,\max} - K_{p,\min})K'_p + K_{p,\min}$$
$$K_d = (K_{d,\max} - K_{d,\min})K'_d + K_{d,\min} \tag{6}$$
$$K_i = \frac{K_p^2}{\alpha K_d}$$

Where K'_p, K'_d and α are determined by means of fuzzy mechanism and $K_{p,max}$, $K_{p,min}$, $K_{d,max}$ and $K_{d,min}$ are adopted to normalize the values of K_p and K_d into the range between zero and one. These constants are determined by the following rule of thumb.

$$K_{p,min} = 0.32K_u,\quad K_{p,max} = 0.6K_u$$
$$K_{d,min} = 0.08K_uT_u,\quad K_{d,max} = 0.15K_uT_u \tag{7}$$

Where K_u and T_u are respectively the gain and the period of oscillation at the stability limit under P-control.

The input *e(t)* and Δ*e(t)* of the fuzzy module are use seven triangular membership function, while the output K'_p and K'_d are use two and four singletons define the output of α. The fuzzy rule determined heuristically based on the step response of process. Thus a set of rules as shown in table 1 for tuning rules K'_p. The tuning rule for K'_d and α are given in table 2 and Table 3, respectively.

Table 1 Fuzzy tuning rules for K'_p

		Δ*e(t)*						
		NB	NM	NS	ZO	PS	PM	PB
e(t)	NB	B	B	B	B	B	B	B
	NM	S	B	B	B	B	B	S
	NS	S	S	B	B	B	S	S
	ZO	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	B	B	B	B	B	B	B

Table 2 Fuzzy tuning rules for K'_d

		Δ*e(t)*						
		NB	NM	NS	ZO	PS	PM	PB
e(t)	NB	S	S	S	S	S	S	S
	NM	B	B	S	S	S	B	B
	NS	B	B	B	S	B	B	B
	ZO	B	B	B	B	B	B	B
	PS	B	B	B	S	B	B	B
	PM	B	B	S	S	S	B	B
	PB	S	S	S	S	S	S	S

Table 3 Fuzzy tuning rules for α

		Δ*e(t)*						
		NB	NM	NS	ZO	PS	PM	PB
e(t)	NB	2	2	2	2	2	2	2
	NM	3	3	2	2	2	3	3
	NS	4	3	3	2	3	3	4
	ZO	5	4	3	3	3	4	5
	PS	4	3	3	2	3	3	4
	PM	3	3	2	2	2	3	3
	PB	2	2	2	2	2	2	2

3. DESIGN

3.1 PID processor

The digital PID processor embedded on chip FPGA which use algorithm from digital PID controller equation. Fig. 1, show block diagram which will be define variable SP and PV have size 10 bit, K_p, K_i and K_d, have size 8 bit and V_o has size 12 bit.

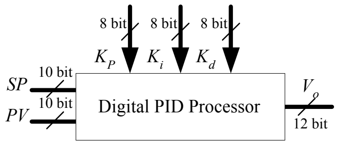

Fig. 1 Block diagram digital PID processor.

An internal digital PID processor to consist of 3 process show in Fig. 2. Process 1, 2 and 3 are share operator in order to reduce number gate of FPGA. Thus all operator are compose 3 input 20 bit full adder, 2 input 10 bit full adder, both 2 input 10 bit multipliers. To work each process defined by state machine which relate to clock signal.

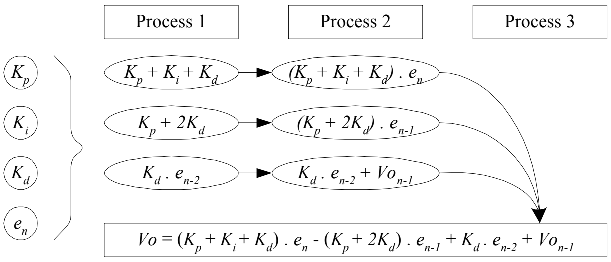
Process 1
Process 2
Process 3
K_p
K_i
K_d
e_n
$K_p + K_i + K_d$
$(K_p + K_i + K_d) . e_n$
$K_p + 2K_d$
$(K_p + 2K_d) . e_{n-1}$
$K_d . e_{n-2}$
$K_d . e_{n-2} + Vo_{n-1}$
$Vo = (K_p + K_i + K_d) . e_n - (K_p + 2K_d) . e_{n-1} + K_d . e_{n-2} + Vo_{n-1}$

Fig. 2 The process within digital PID processor.

Fig. 3 State diagrams for share operator within PID processor.

3.2 Fuzzy – PID Processor

The algorithm internal fuzzy processor will be use the fuzzy gain scheduling method [3] which the integral time constant is determined from Eq. (6), which consist of two input (*e(t)* and Δ*e(t)*), in each input have seven triangle membership function. The five triangle membership functions are used for two outputs K_p and K_d and five singletons for output α. Figs. 3 and 4 show membership function of inputs and outputs. The fuzzy rule determined from heuristically based on the step response of process.

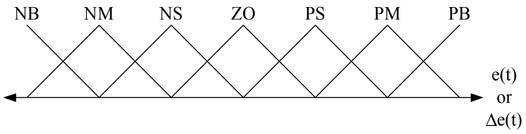

Fig. 3 Membership function for *e(t)* and Δ*e(t)*

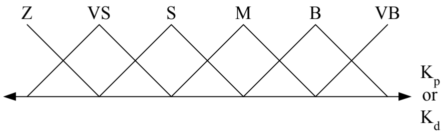

Fig. 4 Membership function for K_p and K_d.

When linguistic variables following:

NB:	Negative Big;	NM:	Negative Medium;
NS:	Negative Small;	Z:	Zero;
PS:	Positive Small;	PM:	Positive Medium;
PB:	Positive Big;	VS:	Very Small;
S:	Small;	M:	Medium;
B:	Big;	VB:	Very Big;

The Fig. 5, show block diagram fuzzy processor which the Fuzzification block are use Look Up Table method: LUT for transfer crisp value to fuzzy set (Notice: the overlapping degree of its membership function is at most two). The Inference rule block performs approximate reasoning by associating input variables with fuzzy rule. The Defuzzification block converts the fuzzy output to crisp value by Center of Gravity (COG) method which shown in Eq. (5). Every block was controlled by clock signal.

The architecture of fuzzy processor on FPGA in each block shown in Figs. 6 - 8.

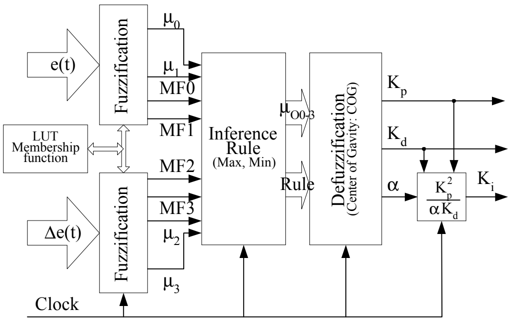

Fig. 5 Block diagram of Fuzzy processor.

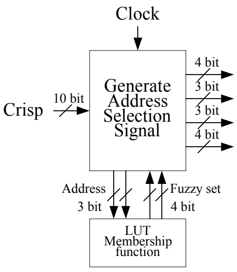

Fig. 6 architecture of the Fuzzification block.

Fig. 6, is show architecture of the fuzzification block. The generate address selection signal block will be generate by relative to crisp value for read fuzzy set from memory and send to inference rule block. The inference rule block will be find reasoning from membership function by LUT and find fuzzy value use max – min operator which fuzzy value will be relate to reasoning show architecture in Fig. 7.

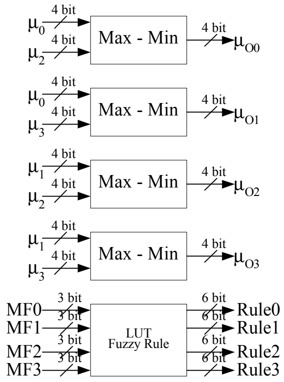

Fig. 7 architecture of the Inference rule block.

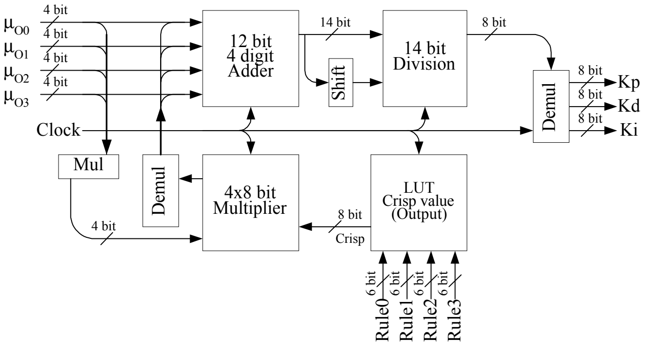

Fig. 8 architecture of the defuzzification block.

Fig. 8, shown architecture of the defuzzification block, which algorithm are use Eq. (5). Then this block consist of adder module, multiplier module [4], division module, memory crisp output value and some other logic. Of cause fuzzy – PID processor desire find parameter of PID (K_p, K_d and K_i) then each module necessary use once again due to desire reduce gate internal chip FPGA. The recall module controlled by clock signal. From designing adder and multiplier module use one clock signal while division module will use 14 clock signals. When fuzzy – PID processor find parameter K_p, K_d, and K_i so that processor will be use 16 clock for multiplier, 3 clock for adder, 56 clock for division, 5 clock for PID processor and some other 5 clock which altogether is 85 clock. Fig. 9 show block diagram of the fuzzy – PID controller which consist of ADC, DAC and fuzzy – PID processor.

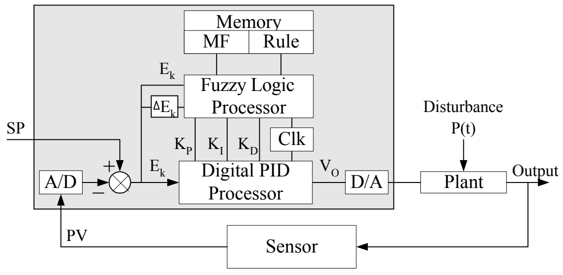

Fig. 9 block diagram of the fuzzy – PID controller.

The fuzzy – PID controller are verification by experiment with level process system which show diagram in Fig. 10. This process control level within bottom tank, which relate to the On – Off valve. The On – Off valve was controlled by fuzzy – PID controller (control signal is 4 – 20 *mA*). Then from observing process has K_u = 75 and T_u = 5 *Seconds* at sampling rate = 100 *ms*.

So that from Eq. (7) then range of parameter following:

$K_{p,min} = 0.32 \times 75 = 24$, $K_{p,max} = 0.6 \times 75 = 45$

$K_{d,min} = 0.08 \times 75 \times 5 = 20$, $K_{d,max} = 0.15 \times 75 \times 5 = 38$

The fuzzy rule of this process shown in Tables 4 – 5 when membership function are shown in Figs. 4 and 5.

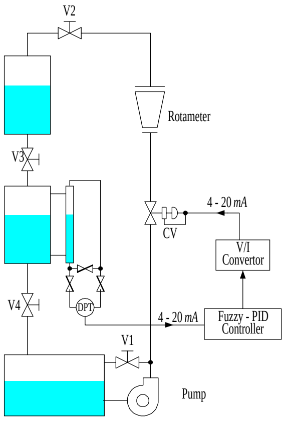

Fig.10 The diagram of experiment level process system.

Table 4 Fuzzy tuning rules for K_p

				Δ*e(t)*				
		NB	NM	NS	ZO	PS	PM	PB
	NB	VB	VB	VB	VB	VB	VB	VB
	NM	Z	B	S	VB	S	B	Z
	NS	Z	VS	VS	VB	VS	VS	Z
e(t)	ZO	Z	Z	VS	VB	VS	VZ	Z
	PS	Z	VS	VS	VB	VS	VS	Z
	PM	Z	B	S	VB	S	B	Z
	PB	VB	VB	VB	VB	VB	VB	VB

Table 5 Fuzzy tuning rules for K_d

				Δ*e(t)*				
		NB	NM	NS	ZO	PS	PM	PB
	NB	Z	Z	Z	Z	Z	Z	Z
	NM	VB	VB	VS	Z	VS	VB	VB
	NS	VB	B	B	Z	B	B	VB
e(t)	ZO	VB	VB	VB	VB	VB	VB	VB
	PS	VB	B	B	Z	B	B	VB
	PM	VB	VB	VS	Z	VS	VB	VB
	PB	Z	Z	Z	Z	Z	Z	Z

Table 6 Fuzzy tuning rules for α

				Δ*e(t)*				
		NB	NM	NS	ZO	PS	PM	PB
	NB	2	2	2	2	2	2	2
	NM	3	3	2	2	2	3	3
	NS	4	3	3	2	3	3	4
e(t)	ZO	5	4	3	3	3	4	5
	PS	4	3	3	2	3	3	4
	PM	3	3	2	2	2	3	3
	PB	2	2	2	2	2	2	2

4. EXPERIMENTAL RESULT

4.1 Simulation

This processor verified by simulate which will be use application program ModelSimXE. Fig. 11, is show Fuzzy-PID processor simulation for observe changing parameter of processor.

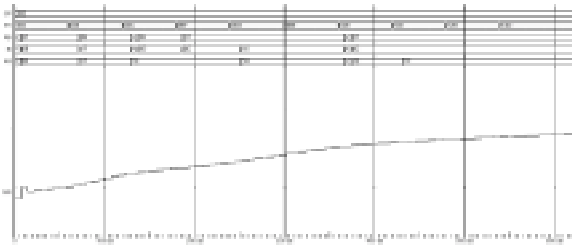

Fig. 11 Simulate the Fuzzy - PID processor.

From result when change process have effect to parameters of PID are change relate to the fuzzy tuning rule.

From the XilinxWebpack Synthesis this processor can execute at maximum frequency: 40.550 *MHz* but one action it use 85 clock cycle so that execute times about 24.661 *ns* × 85 = 2.096 *µs per action* show timing summary below.

Timing Summary:

Speed Grade: -5

Minimum period: 24.661*ns* (Maximum Frequency: 40.550*MHz*)
Minimum input arrival time before clock: 14.496*ns*
Maximum output required time after clock: 9.992*ns*
Maximum combinational path delay: 13.384*ns*

4.2 Implementation fuzzy – PID controller result

The experimental result show in fig 12, fig 13 and fig 14 which are control level at 50 % after that step change to 75 %, control level at 75 % after that step change to 50 % and control level at 75 % after that close valve about 5 sec. (Disturbance) respectively.

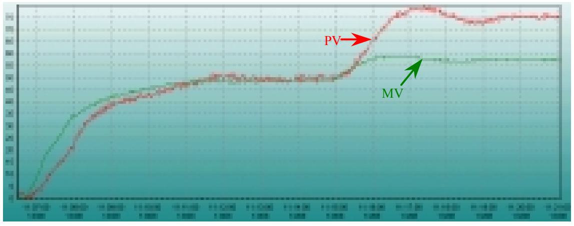

Fig. 12 The experimental result when control level at 50 % and then step change to 75 % (Step up).

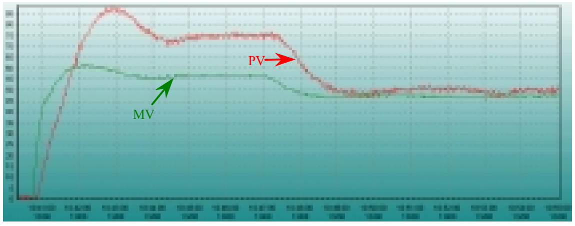

Fig. 13 The experimental result when control level at 75 % and then step change to 50 % (Step down).

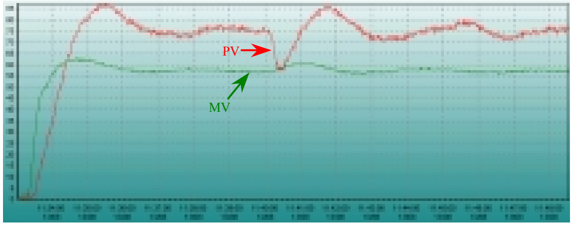

Fig. 14 The experimental result when control level at 75 % and then close valve about 5 sec. (Disturbance).

From the result of controlling the process has overshoot about 18 % and stable about 5 *minutes.*

5. CONCLUSIONS

The fuzzy – PID processor can change parameter of PID processor in order to relate to fuzzy rule. Then other process, controller is necessary to adjust fuzzy rule inference in order to suitability with process. From the timing summary maximum execute time about 2.096 *μs per action.*

The fuzzy – PID controller based on chip FPGA are verified by experimental with level process. The fuzzy rule inference are obtain form observe process (trial and error) which K_u = 75 and T_u = 5 *Seconds* at sampling rate = 100 *ms.* After consider manipulate variable (MV) of the result are change follow to algorithm of PID but nonlinear due to parameter change depend on process variable (PV). According to the overshoot about 18 % and stable about 5 *minutes.*

Finally, the fuzzy PID processor can embedded on chip FPGA. This designing had maximum execute time about 2.096 *μs per action.* Then it can apply to control process faster than this process only if adjust fuzzy rule inference and sampling time.

REFERENCES

[1] Chi-Tsong Chen, et al., “Analog and digital control system design,” *Saunders College Publishing*, 1993.

[2] J. Yan, M. Ryan and J. Power, “Using fuzzy logic toward intelligent system,” *Prentice Hall*, New York, 1994.

[3] Z.Y. Zhao, et al., “Fuzzy gain scheduling of PID controller,” *IEEE*, pp. 698-703, 1992.

[4] Charles H. Roth, Jr, “Digital System Design Using VHDL,” *PWS Publishing Company*, Boston, 1998.

[5] R. Townsend, “Digital Computer Structure and Design,” *Butterworth Scientific*, london, 2nd, pp.117-143, 1982.

[6] A. Visioli, “Tuning of PID controllers with fuzzy logic,” *IEE proc.-Control Theory Appl.*, *Vol. 148.No.1*, 2001.