Turkish Journal of Electrical Engineering and Computer Sciences

Volume 13 | Number 2 Article 6

1-1-2005

Implementation of a New Self-Tuning Fuzzy PID Controller on PLC

ONUR KARASAKAL

ENGİN YEŞİL

MÜJDE GÜZELKAYA

İBRAHİM EKSİN

Turk J Elec Engin, VOL.13, NO.2 2005, © TÜBİTAK

Implementation of a New Self-Tuning Fuzzy PID Controller on PLC

Onur KARASAKAL[1], **Engin YEŞİL**[2],
Müjde GÜZELKAYA[2], **İbrahim EKSİN**[2]
[1] *Turkish Naval Academy, 34942 Tuzla; İstanbul-TURKEY*
e-mail: onurkarasakal@gmail.com
[2] *Control Engineering Division, Faculty of Electrical and Electronics Engineering,*
İstanbul Technical University, 34469 Maslak; İstanbul-TURKEY
e-mail: yesil@elk.itu.edu.tr gkaya@elk.itu.edu.tr eksin@elk.itu.edu.tr

Abstract

In this study, the self-tuning method for fuzzy PID controllers that has been developed in a previous study of the authors is implemented on PLC in order to control some standard processes formed on FEEDBACK PCS 327 Process Control Simulator. In this tuning method, the input scaling factor corresponding to the derivative coefficient and the output scaling factor corresponding to the integral coefficient of the fuzzy PID controller are adjusted using a fuzzy inference mechanism with a new input called "normalized acceleration". The results of the implementation have been compared with those of the classical fuzzy PID controller without a tuning mechanism and it has been observed that the tuning mechanism decreases the oscillations and the settling time while providing smoother system responses also in real time application.

Key Words: *Fuzzy PID controllers; relative rate observer; self-tuning mechanisms; Programmable Logic Controllers.*

1. Introduction

In literature, various structures for fuzzy PID (including PI and PD) controllers and fuzzy non-PID controllers have been proposed. Fuzzy PI control is known to be more practical than fuzzy PD because it is difficult for the fuzzy PD to remove steady state error. The fuzzy PI control, however, is known to give poor performance in transient response for higher order processes due to the internal integration operation. Thus, in practice the fuzzy PID controllers are more useful. To obtain proportional, integral and derivative control action all together, it is intuitive and convenient to combine PI and PD actions together to form a Fuzzy PID (FPID) controller. One way of constructing a FPID controller is achieved by summing the fuzzy PD controller output and its integrated part [1-3]. Such a fuzzy PID controller with a single rule-base, which is used in this study, is illustrated in Figure 1.

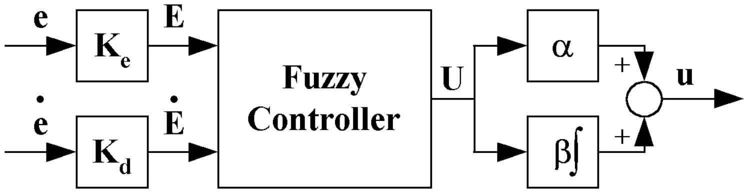

Figure 1. Two-input FPID controller structure with one rule-base.

The design parameters of the FPID controllers can be summarized within two groups [4]: Structural parameters, and tuning parameters.

Structural parameters include input/output (I/O) variables to fuzzy inference, fuzzy linguistic sets, membership functions, fuzzy rules, inference mechanism and defuzzification mechanism. Tuning parameters include I/O scaling factors (SF) and parameters of membership functions (MF). The structural parameters are usually determined during off-line design while the tuning parameters can be calculated during on-line adjustments of the controller to enhance the process performance, as well as to accommodate the adaptive capability to system uncertainty and process disturbance.

There exist various heuristic and non-heuristic tuning strategies for the adaptation of scaling factors of fuzzy controllers [5-7]. The relative rate observer idea given in [8] proposes a simple tuning structure. In this tuning method, the input scaling factor that corresponding to the derivative and the output scaling factor corresponding to the integral coefficients of the FPID controller are adjusted using a fuzzy inference mechanism in an on-line manner. In the inference mechanism there exist two inputs, one of which is the classical actuating "error" and the other one is the so-called "normalized acceleration" that provides relative rate information designating the fastness or slowness of the system response. In order to show the effectiveness and the power of this new approach, the self-tuning FPID has been implemented on Simatic S7-200 PLC to control some standard processes formed on FEEDBACK PCS 327 Process Control Simulator. The results are compared with the classical two-input FPID controller without a tuning mechanism.

2. Two-input Fuzzy PID Controller Structure

The closed-loop control structure that will be considered in this study is shown in Figure 2. The output of the fuzzy PID controller is given by

$$u = \alpha U + \beta \int U dt \tag{1}$$

where U is the output of the FLC.

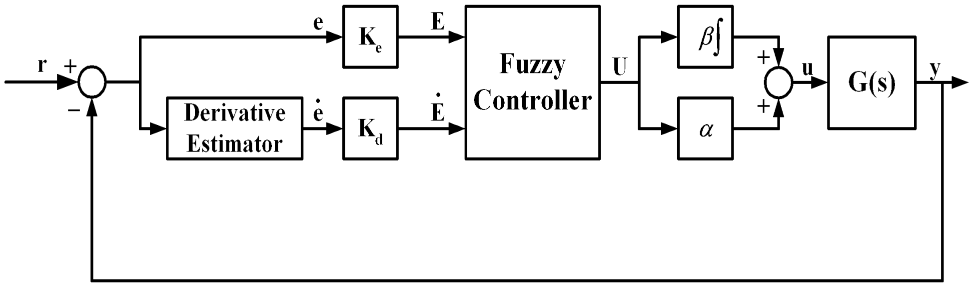

Figure 2. The closed-loop control structure for two-input FPID controller.

It has been shown in [9] that the fuzzy controllers with product–sum inference method, center of gravity defuzzification method and triangular uniformly distributed membership functions for the inputs and a crisp output, the relation between the input and the output variables of the FLC is given by

$$U = A + PE + D\dot{E} \tag{2}$$

where $E = K_e.e$ and $\dot{E} = K_d.\dot{e}$. The same result is shown to be valid for the minimum inference engine in [10]. Therefore, from (1) and (2) the controller output is obtained as

$$u = \alpha A + \beta At + \alpha K_e Pe + \beta K_d De + \beta K_e P \int edt + \alpha K_d D\dot{e} \tag{3}$$

Thus, the equivalent control components of the fuzzy PID controller are obtained as follows:

$$\begin{array}{lll} \textbf{P}\text{roportional gain} & : & \alpha K_e P + \beta K_d D \\ \textbf{I}\text{ntegral gain} & : & \beta K_e P \\ \textbf{D}\text{erivative gain} & : & \alpha K_d D \end{array} \tag{4}$$

3. Relative Rate Observer Based Self-tuning of Two-input FPID Controller

Parameter adaptive two-input FPID controller using a relative rate observer has been proposed in [8]. The block diagram of the proposed method is shown in Figure 3.

This method adjusts the scaling factors that correspond to the derivative and integral coefficients of the fuzzy PID controller using a fuzzy inference mechanism in an on-line manner. The fuzzy inference mechanism that adjusts the related coefficients has two inputs one of which is "system error" designated as e and the other one is a new variable r_v named as "normalized acceleration". The normalized acceleration gives "relative rate" information about the fastness or slowness of the system response.

The normalized acceleration $r_v(k)$ is defined as:

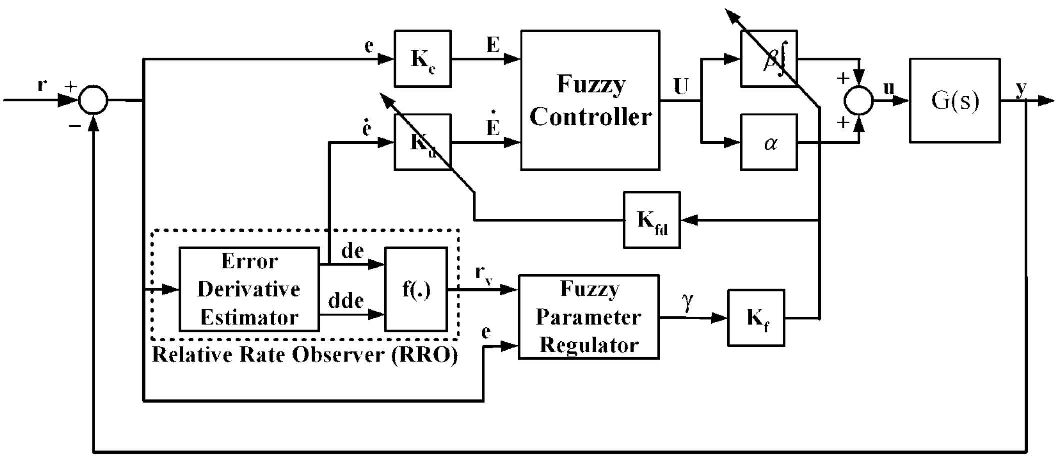

Figure 3. The closed-loop control structure for parameter adaptive two-input FPID controller via relative rate observer.

$$r_v(k) = \frac{de(k) - d(k-1)}{de(.)} = \frac{dde(k)}{de(.)} \tag{5}$$

Here, de(k) is the incremental change in error and it is given by

$$de(k) = e(k) - e(k-1) \tag{6}$$

and dde(k) is called the acceleration in error and it is given by

$$dde(k) = de(k) - de(k-1) \tag{7}$$

In (5), de(.) is chosen as follows

$$de(.) = \begin{cases} de(k) & if \quad |de(k)| \geq |de(k-1)| \\ de(k-1) & if \quad |de(k)| < |de(k-1)| \end{cases} \tag{8}$$

The output of the fuzzy parameter regulator is designated as γ in Figure 3. The scaling factor K_d is adjusted by multiplying its predetermined value K_{ds} by γ whereas the scaling factor β is adjusted by dividing its predetermined valueβ_S the same coefficient factor as it is given below.

$$K_d = K_{ds} \,.\, K_{fd} \,.\, K_f \,.\, \gamma \tag{9}$$

$$\beta = \frac{\beta_s}{K_f \,.\, \gamma} \tag{10}$$

It is seen that K_f is the output scaling factor for the Fuzzy Parameter Regulator block and K_{fd} is the additional parameter that effects only the derivative factor of the FLC. If K_{fd} is chosen as unity, then the equivalent proportional control strength given in Eq. (4) does not change. This case could be named as one-parameter adjustment, since there is only one parameter to be adjusted. On the other hand, when K_{fd} and K_f are both to be adjusted then it naturally becomes two-parameter adjustment case.

The meta-rules for determining γ can be summarized as follows:

(i) When the system response is slow, the derivative effect of the two-input FPID controller must decrease.

(ii) When the error is small and the system response is fast, the derivative effect of the two-input FPID controller must increase.

4. Implementation

In the implementation mode, the number of the fuzzy rules is reduced to four as given in Table 1; whereas, the rule-base of the Fuzzy Controller consists of twenty five rules in simulation studies given in [8].

Table 1. Simple rules used in Fuzzy Controller.

$\dot{e}$ \ e	**N**	**P**
N	N	Z
P	Z	N

Again, in the simulation studies given in [8], the membership functions used in both antecedent and consequent parts of the fuzzy rules are fifty percent overlapped triangular membership function; whereas, in the implementation mode, the membership functions used in the consequent parts of the fuzzy rules are chosen as singleton to reduce computational effort. The input and output membership functions are both illustrated in Figure 4. As an "and" operator the "product" is used while the weighted average method is preferred in the defuzzification mechanism to center of gravity method in implementation phase. These preferences provide a very simple FPID controller structure that can be programmed on PLC just to show the effect of the fuzzy self-tuning method.

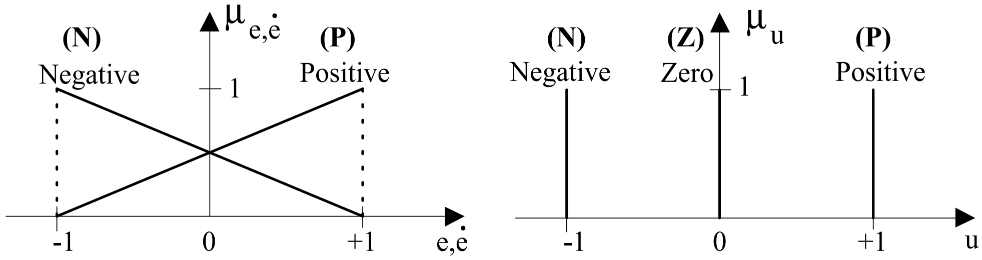

Figure 4. The membership functions used in Fuzzy Controller.

The rule-base of Fuzzy Parameter Regulator that has been given in Table 2 is proposed using meta-rules given in Section 3. The inputs of the fuzzy rule-base are the absolute value of error "$|e|$" and the normalized acceleration "r_v". The linguistic levels assigned to the input variable $|e|$ and the output variable "γ" are as follows:

L : large; M : medium; SM : small medium; S : small

For the other input variable r_v the following linguistic levels are assigned:

F : fast; M : moderate; S : slow

As it is done in forming main Fuzzy Controller, the symmetrical triangular uniformly distributed membership functions are assigned for the input and output linguistic variables for Fuzzy Parameter Regulator in simulation studies given in [8]. On the other hand, in this study, singleton membership functions are assigned for consequent part of the fuzzy rules agin to reduce computational effort in implementation.

Table 2. Rule-base for Fuzzy Parameter Regulator.

$\|e\|$ \ r_v	**S**	**M**	**F**
S	M	M	L
SM	SM	M	L
M	S	SM	M
L	S	S	SM

The relative rate observer based self-tuning two-input FPID controller has been quite easily implemented using Simatic S7-200 CPU 214 processor and EM 235 analog I/O unit on FEEDBACK PCS 327 Process Control Simulator. A PLC can be defined as a microprocessor-based control device, with the original purpose of supplementing relay logic. Early PLCs were able to perform only logical operations. PLCs can now perform more complex sequential control algorithms with the increase in microprocessor performance. On the other hand, they can now admit analog inputs and outputs. Therefore, today the majority of specialists agree that the real future of PLCs lies not only in traditional discrete process control, but also in the area of demanding continuous and, particularly, batch processes, which are a combination of continuous and discrete processes. Thus, today a typical PLC-based application deals with several hundreds of analog and digital inputs and outputs, while performing quite complex control procedures [12].

The most of the high-order processes can usually be modeled as first-order plus dead time (FOPDT) or second-order plus dead time (SOPDT) systems [13, 14]. The FOPTD and SOPDT systems that have the following respective transfer functions are formed on the process control simulator:

$$G_{FOPDT}(s) = \frac{1}{s+1}e^{-s} \quad ; \quad G_{SOPDT}(s) = \frac{1}{(s+1)^2}e^{-s} \tag{11}$$

Figure 5. The experimental setup used in the study.

The overall view of the experimental system is given in Figure 5. The reference value is chosen as 5 V for all the experiments.

4.1. Experiment on first-order plus dead time (FOPDT) system

One-parameter adjustment case:

The scaling factors of the two-input FPID controller are chosen as follows:

$$K_e = 0.9, \;\; K_d = 0.7, \;\; \beta = 0.4, \;\; \alpha = 1$$

just to ensure the stability of the system and to keep the control effort within the boundaries, which is set as [0 10] for this case. Then, the parameter of the relative rate observer is searched using a standard genetic search algorithm within the range [0 30]. The optimum value for this parameter is found out to be K_f =6.45. The step responses and the related control signals are given in Figure 6. In all of the figures below the dashed-lines show input and output of systems without a tuning mechanism while the solid-lines show input and output of systems with the relative rate observer based self- tuning mechanism.

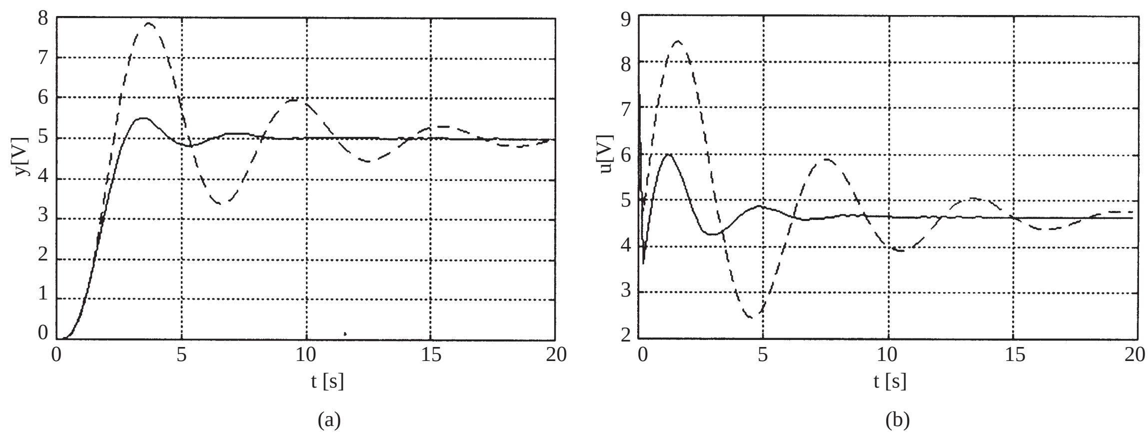

Figure 6. (a) The step responses of the FOPDT system; (b) the control signals for the system.

Two-parameter adjustment case:

While the scaling factors of the two-input FPID controller for this case are kept the same, the parameters of the relative rate observer based FPID are searched again using a standard genetic search algorithm within the range [0 30]. The optimum values for these parameters are found as

$$K_f = 7.8 \; and \; K_{fd} = 0.25.$$

For this case, the proportional gain in Equation. (4) is not kept constant anymore. The step responses and the related control signals are given in Figure 7.

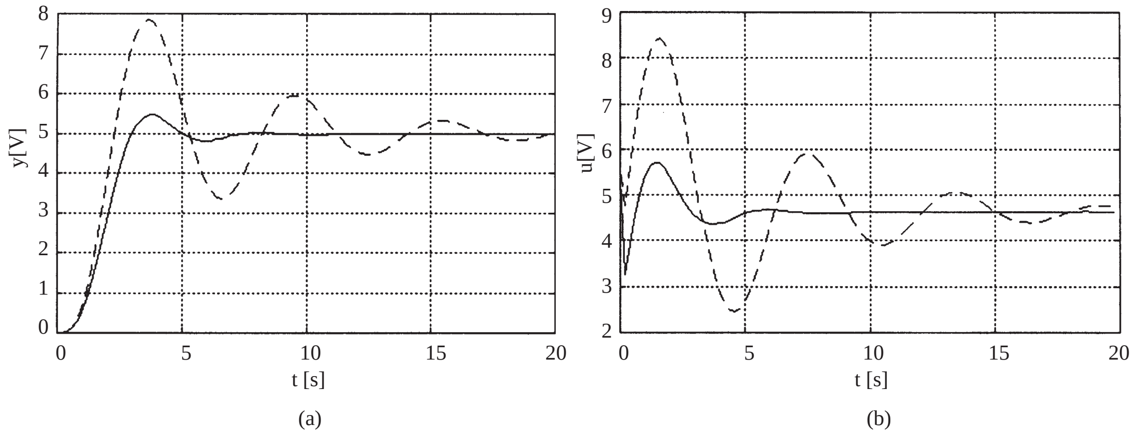

Figure 7. (a) The step responses of the FOPDT system; (b) the control signals for the system.

4.2. Experiment on second-order plus dead time (SOPDT) system

One-parameter adjustment case:

The scaling factors of the two-input FPID controller are chosen as follows:

$$K_e = 0.4,\ \ K_d = 0.7,\ \ \beta = 0.6,\ \ \alpha = 1$$

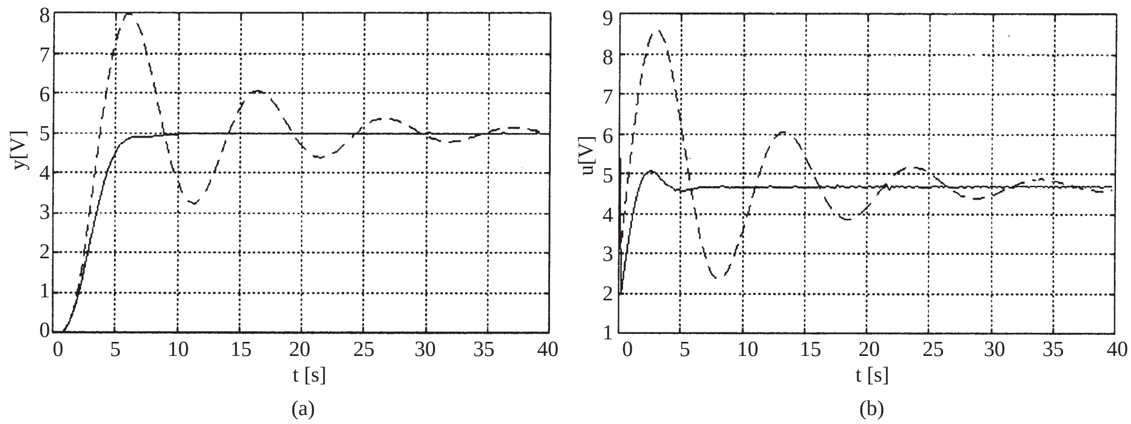

Figure 8. (a) The step responses of the SOPDT system; (b) the control signals for the system.

When the parameter of the relative rate observer based FPID is searched again using a standard genetic search algorithm within the range [0 30], the optimum value for the parameter is found out to be $K_f = 8$. The step responses and the related control signals are given in Figure 8.

Two-parameter adjustment case:

While the scaling factors of the two-input FPID controller for this case are kept the same, the parameters of the relative rate observer based FPID are searched using a standard genetic search algorithm within the range [0 30]. The optimum values for these parameters are found as

$$K_f = 7.5,\ \ K_{fd} = 1.4$$

The step responses and the related control signals are given in Figure 9.

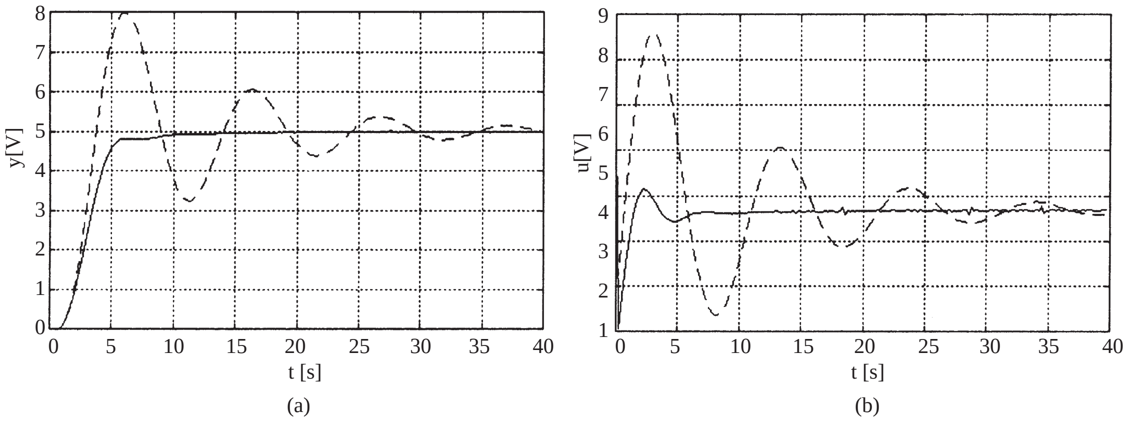

Figure 9. (a) The step responses of the SOPDT system; (b) the control signals for the system.

5. Conclusion

A two-input FPID controller that has been implemented on a PLC and it has been used to control first and second-order systems with dead time formed on a process simulator. The input scaling factor corresponding to the derivative coefficient and the output scaling factor corresponding to the integral coefficient of the two-input FPID controller has been adjusted using a relative rate observer based tuning method. The relative rate observer method provides a satisfactory response with only one parameter adjustment. For this case, the proportional gain in (4) is kept constant. When two parameters of the controller are tuned, a little better performance at settling time is achieved comparing with one parameter adjustment case. However, in this case, the proportional gain cannot be kept unchanged.

When either a parameter variation occurs in system parameters or the controller parameters are not set appropriately, the implementations have proved us that new self-tuning method would provide a significant enhancement in the system performance.

References

[1] J.X. Xu, C.C. Hang, C. Liu, “Parallel structure and tuning of a fuzzy PID controller,” Automatica, Vol. 36, pp. 673-684, 2000.

[2] S. Z. He, T. D. Shaoua, F. L. Xu, “Fuzzy self-tuning of PID,” Fuzzy Sets and Systems, Vol.56, pp. 37-46, 1993.

[3] E. Yeşil, M. Güzelkaya, İ. Eksin, ”Self-tuning Fuzzy PID-type Load-frequency controller,” Electrical Conversion and Management, Vol. 45/3, pp. 377-390, 2004.

[4] B. Hu, G. K. I. Mann, R. G. Gosine, “A new methodology for analytical and optimal design of fuzzy PID controllers,” IEEE Trans. Fuzzy Systems, Vol. 7(5), pp. 521-539, 1999.

[5] C. H. Chou, H. Lu, “A heuristic self-tuning fuzzy controller,” Fuzzy Sets and Systems, Vol. 61, pp. 249-264, 1994.

[6] C. H. Jung, C. S. Ham, K. I. Lee, “A real-time self-tuning fuzzy controller through scaling factor adjustment for adjustment for the steam generator of NPP,” Fuzzy Sets and Systems, Vol. 74, pp. 53-60, 1995.

[7] R. K. Mudi, N. R. Pal, “A robust self-tuning scheme for PI- and PD-type fuzzy controllers,” IEEE Trans. Fuzzy Systems, Vol. 7(1), pp. 2-16, 1999.

[8] M. Güzelkaya, İ. Eksin, E. Yesil, "Self-tuning of PID-type fuzzy logic controller coefficients via relative rate observer," Engineering Applications of Artificial Intelligence, Vol. 16, pp. 227-236, 2003.

[9] W. Z. Qiao, M. Mizumoto, "PID type fuzzy controller and parameters adaptive method," Fuzzy Sets and Systems, Vol. 78, pp. 23-35, 1996.

[10] T. T. Huang, H. Y. Chung, J. J. Lin, "A fuzzy PID controller being like parameter varying PID," IEEE International Fuzzy Systems Conference Proceeding, Seoul, Korea, pp. 269-275, 1999.

[11] İ. Eksin, M. Güzelkaya, F. Gürleyen, "A new methodology for deriving the rule-base of a fuzzy logic controller with a new internal structure," Engineering Applications of Artificial Intelligence, Vol. 14(5), pp. 617-628, 2001.

[12] G. Godena, M. Colnaric, "Exception handling for PLC-based process control software," Microprocessors and Microsystems, Vol. 24, pp. 407-414, 2000.

[13] H.-P. Huang, M.-L. Roan, J.-C. Jeng, "On-line adaptive tuning for PID controllers," IEE Proc.-Control Theory Appl., Vol. 149(1), pp. 60-67, 2002.

[14] O. Camacho, C. Smith, W. Moeno, "Development of an Internal Model Sliding Mode Controller," Ind. Eng. Chem. Res., Vol. 42, pp. 568-573, 2003.