

Tuning PID Control: A Hybrid Approach using Fuzzy Logic and Genetic Programming

Gustavo Maia de Almeida, *Student Member, IEEE*, Valceres Vieira Rocha e Silva, Erivelton Geraldo Nepomuceno, José Denti Filho, and Ryuichi Yokoyama

Abstract—The majority of the research work on fuzzy PID controllers focuses on the conventional two-input PI or PD type controller proposed by Mamdani method. However, fuzzy PID controller design is still a complex task due to the involvement of a large number of parameters in defining the fuzzy rule base. In this work, a method is proposed to optimal tuning the parameters of PID controllers. Different from traditional techniques, the tuning procedure of the proposed method is described in terms of fuzzy rules, in which the input variable is the error signal, and the output variables are the PID parameters. Genetic programming (GP) is then used to search for the optimal PID parameters that will minimize the integral of the squared error. The hybrid method to tune PID controllers was compared to the performance achieved by four classical PID tuning schemes that are widely used in industry. The simulations show that hybrid method always achieves a performance that is at least as good as those achieved of the classical PID tuning schemes, and often better: faster settling time and the minimal integral squared error. In addition, the parameters obtained through the four comparative methods, cannot always effectively control systems with changing parameters, and may need frequent on-line retuning, the controllers of the hybrid method are adapted on-line based on parameters estimation, requiring certain knowledge of the process.

Index Terms—Fuzzy PID controllers, hybrid method, Genetic Programming

I. INTRODUCTION

THE majority of the industrial processes are controlled by proportional-integral-derivative (PID) controllers [3], [13]. The popularity of PID controllers is due to their simplicity both from the design and the parameter tuning points of view. To implement such a controller, three parameters, namely the proportional gain K_p , the integral time T_i , and the derivative time T_d must be determined in order to make the system operation more efficient.

The PID controllers in the literature can be divided into two main categories. In the first one, the controller parameters are fixed during control after they have been tuned or chosen in a certain optimal way. The Ziegler-Nichols tuning formula is perhaps the most well-known tuning method [15]. The PID controllers of this category are simple, but cannot always effectively control systems with changing parameters, and may

need frequent on-line retuning. The controllers of the second category have a structure of the plant model. Such controllers are called adaptive PID controllers in order to differentiate them from those of the first category.

To enhance the capabilities of traditional PID tuning techniques, several new methods, such as genetic programming (GP) [1] and fuzzy logic controllers [12] and [7] have been recently developed to improve the tuning of the PID parameters controllers. With the abilities for global optimization and good robustness, the GP is expected to overcome the weakness of traditional PID tuning techniques and to be more acceptable to industrial practice. However, since the PID parameters generated by the GP are fixed, the PID controllers cannot always effectively control systems with changing parameters.

The tuned PID parameters, using the fuzzy method [14], are adaptive and expressed by the fuzzy rules. In this manner, the PID controllers tuned by this method will have more flexibility in controlling industrial processes. However, since fuzzy rules are obtained through human expertise or by experiments, further improvement must be undertaken if the fuzzy rules have to be generated automatically.

To deal with these problems, a hybrid method that combines the advantages of the fuzzy logic controllers and the GP is proposed to optimally tune the parameters of PID controllers. In this method, the PID parameters are adaptive and determined according to the values of the error signal. Therefore, the tuning procedure can be described in terms of fuzzy rules, in which the input variable is an error, and the output variables are the PID parameters. A GP is then used to search for the optimal PID parameters that will maximize the fitness function, which is defined as the reciprocal of the integral of the squared error. Since no human expertise is needed in the tuning procedure and since the PID parameters are adaptive, good control performance can be expected for the proposed method.

The present work presents perspectives of application of hybrid algorithm in control and is organized as follows: Section 2 describes the PID controllers. Section 3 gives an overview of fuzzy logic controllers, and Section 4 an overview of GP. In Section 5, the used methodology is described. The results are presented in Section 6 and in Section 7 conclusions about the use of the hybrid algorithm in control are presented.

II. PID CONTROLLER

In general, a classical PID closed loop control system can be depicted as shown in Fig 1, in which the PID controller output-

G. M. Almeida is with the Department of Electrical Engineering, Federal University of Espírito Santo, Vitória Brazil, CA 29060-970. Phone: +55 27 3389-0045, e-mail: gmaialmeida@yahoo.com.br

V. V. R. Silva is with the Department of Electrical Engineering, Federal University of São João del-Rei, São João del-Rei Brazil, CA 36307-352. Phone: + 55 32 3379-2552, e-mail: vvrsvilva@ufsj.edu.br

E. G. Nepomuceno is with the Department of Electrical Engineering, Federal University of São João del-Rei, São João del-Rei Brazil, CA 36307-352. Phone: + 55 32 3379-2367, e-mail: nepomuceno@ufsj.edu.br

J. D. Filho is with the Department of Electrical Engineering, Federal University of Espírito Santo, Vitória, Brasil, CA 29060-970. Phone: + 55 27 3335-2166, e-mail: j.denti@ele.ufes.br

R. Yokoyama is with the Department of Electrical Engineering, Tokyo Metropolitan University, Tokyo, Japan, CA 193.0397. Phone: + 81 426 77-1111 e-mail: yoko@eei.metro-u.ac.jp

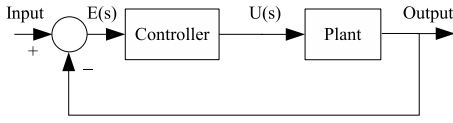


Fig. 1. Closed-loop system

input function is expressed as in Equation 1.

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right). \quad (1)$$

If different values of K_p , T_i and T_d are chosen, different responses of the plant are obtained. Therefore, the PID controller parameters tuning problem can be considered as selecting the three parameters K_p , T_i and T_d such that the response of the plant will be as desired.

III. FUZZY TUNING RULES

Different from traditional tuning techniques, in which the PID parameters are fixed after being tuned, the parameters generated by the proposed method are adaptive and expressed by the fuzzy rules.

Every element in the universe of discourse is a member of a fuzzy set to some grade, sometimes even the zero value. The grade of membership for all its members describes a fuzzy set. In fuzzy sets, elements are assigned a grade of membership, such that the transition from membership to non-membership is gradual rather than abrupt. The set of elements that have a non-zero membership is called the support of the fuzzy set. The function that ties a number to each element x of the universe is called the *membership functions* $\mu(x)$.

The designer is inevitably faced with the question of how to build the fuzzy sets. There are two specific questions to consider: (i) How does one determine the shape of the sets? (ii) How many sets are necessary and sufficient? According to fuzzy set theory the choice of the shape and width is subjective, but a few rules of thumb apply.

- A fuzzy set should be sufficiently wide to allow for noise in the measurement.
- A certain amount of overlap is desirable: otherwise the controller may run into poorly defined states

A preliminary answer to questions (i) and (ii) is that the necessary and sufficient number of sets in a family depends on the width of the sets, and vice versa. A solution could be to ask the process operators to enter their personal preferences for the membership curves. The manual for the TILShell product recommends the following [5]: in the proposed approach, PID parameters are determined based on the current error. The membership function (MF) of these fuzzy sets for $e(t)$ is shown in Figure 2 using the common triangular (2) and trapezoidal functions (3), where in the triangular function, the parameters a and c locate the feet of the triangle and the parameters c locates the peaks, in the trapezoidal function, the parameters a and d locate the feet of the trapezoid and the parameters b and c locate the shoulders. In this figure, N represents negative, P positive, Z0

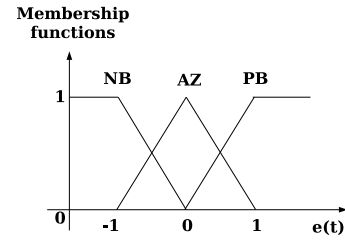


Fig. 2. Membership functions for $e(t)$

approximately zero, B big. Thus NB stands for negative big, PB stands for positive big.

$$f(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right). \quad (2)$$

$$f(x; a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right). \quad (3)$$

The fuzzy rules may be extracted from operator's expertise. Here we drive the experiment based on the step response of the process. Fig 3 shows an example of a desired time response. At beginning, i.e., around a_1 we need a big control signal in order to achieve a fast rise time. To produce a big control signal, the PID controller should have a large proportional gain, a large integral gain, and a small derivative gain. Therefore, the rule around a_1 reads:

if $e(t)$ is PB, then K_p is big, T_i is big, and T_d is small.

Around point b_1 in Fig 3, we expect a small control signal to avoid a large overshoot. That is, the PID controller should have a small proportional gain, a large derivative gain, and a small integral time. Thus the following fuzzy rule is taken:

if $e(t)$ is ZO, then K_p is small, T_i is small, and T_d is big.

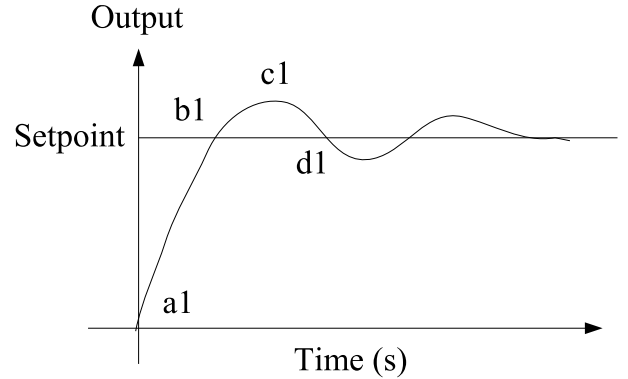


Fig. 3. Process step response.

IV. GENETIC PROGRAMMING

The research in GP has been growing recently due their difference from ordinary optimization tools.

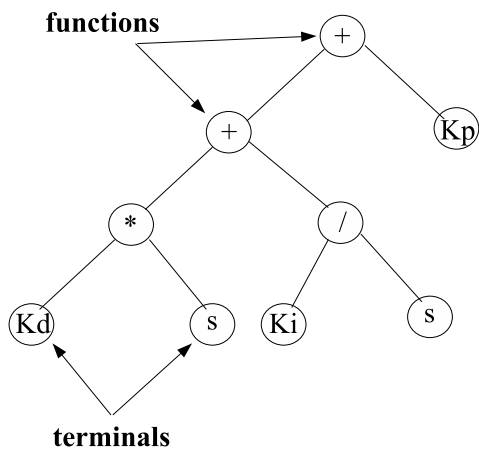


Fig. 4. Individual of the Genetic Programming.

The GP is part of the evolutionary computation that uses the concepts of the natural selection of Darwin and the genetics of Mendel in the computation environment. In such algorithms, the fittest among a group of artificial creatures can survive and form a new generation. In every new generation, a new set of offsprings is created using features of the fittest of the old generation [9].

Even a simple GP can give satisfactory results in a large variety of engineering optimization problems [2], [6], [4], [10]. Generally, GP consist of three fundamental operators: reproduction, crossover and mutation.

- a) *Reproduction*: The best individuals of the current generation are copied equal in the new offspring, so that these are not simply lost.
- b) *Crossover*: Produces new individuals that have some parts of both parent's genetic material for the next offspring preserving the individuals best features of the old generation in the new generation.
- c) *Mutation*: An individual is selected, and a function or a combination of terminals and functions flips with another ones at the *loci* selected to be the mutation point. The role of mutation is seen as providing a guarantee that the probability of searching any given function or a combination of terminal and function will never be zero.

In this case, a representation of an individual, that takes in account the structure of PID controller, can be shown in Figure 4.

Given an optimization problem, GP run iteratively using the three operators in a random way but based on the fitness function to perform evaluation.

Fitness is a numeric value assigned to each member of a population to provide a measure of the appropriateness of a solution to the problem in question. Fitness functions are generally based upon the error between the actual and predicted solutions. However, error based measures decrease for better solutions.

The overall operation of a GP may be explained through the flowchart shown in Figure 5, where i refers to an individual in the population of size M . The "Generation" gives the number of the current generation. The flowchart can be divided in three parts:

1. creation of an initial population of random functions and terminals;
2. iteratively perform of the following sub-steps until the termination criterion has been satisfied:
 - a) simulation of the algorithm for each individual in the population and assign a fitness value to it according to how well it behaves;
 - b) creation of a new population of computer programs by,
 - (i) copying existing computer programs into the new population;
 - (ii) creating new computer programs by genetically recombining randomly chosen parts of the two existing programs;
 - (iii) creating a new computer program introducing random changes. This operation is applied to the chosen computer program(s) with a probability based on their fitness in the population structure;
3. the best computer program that appeared in any generation, is designated as the result of genetic programming. This result may be a solution (or an approximate solution) to the problem.

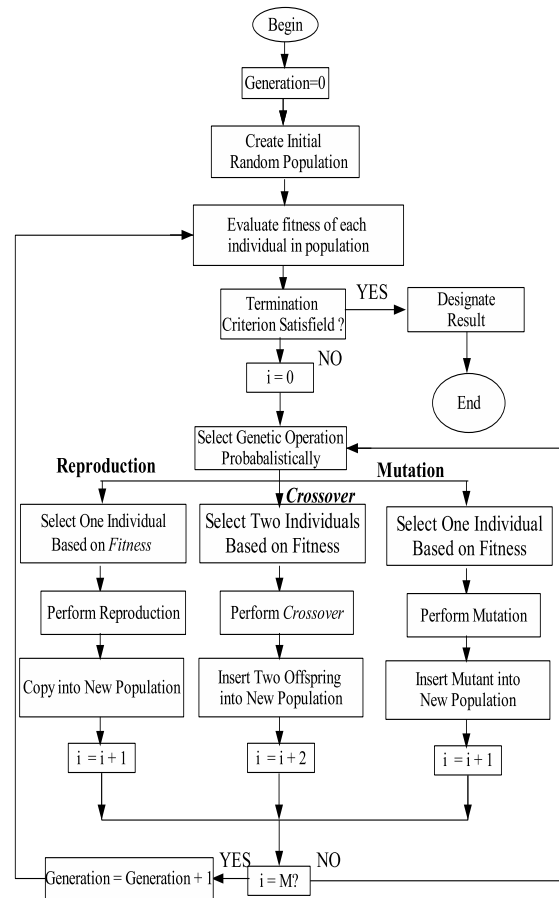


Fig. 5. Flowchart of a generic GP algorithm.

The block diagram of the proposed method PID controller parameters tuning is shown in Fig 6. Since K_p , T_i , and T_d are

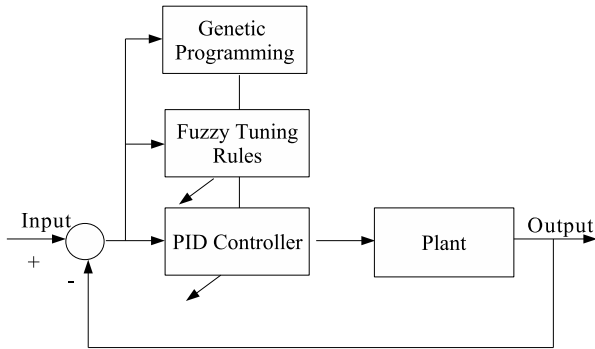


Fig. 6. Block diagram of the proposed method for PID controllers parameters tuning.

expressed in terms of e , the PID parameters tuned by the proposed method should have more flexibility and capability than those tuned by traditional techniques. However, to apply the fuzzy tuning rules, the values of K_p , T_i , and T_d , corresponding to each rule must be determined in an optimal way first. Therefore, a performance index will be defined and an algorithm to search for the optimal values of the PID parameters.

Searching for the optimal value of K_p , T_i , and T_d that maximize the fitness function, it is expected that the proposed algorithm will meet at least the following three requirements: the ability to handle nonlinearities; the capability of solving large-scale problems since the number of variables is high, and the algorithm should generate an optimal solution rapidly without being stuck at a local optima. GP meets all the above requirements and, therefore, can be applied to search for the optimal value of K_p , T_i , and T_d .

The parameters tuning procedure are summarized as follows:

- Step 1: Given the membership functions of e ;
- Step 2: Define the fitness function;
- Step 3: Determine the population size, the crossover rate and the mutation rate;
- Step 4: Create an initial generation in a random way;
- Step 5: For each member in the generation, compute the values of K_p , T_i , and T_d ;
- Step 6: Evaluate the fitness of the each PID controller set;
- Step 7: Generate offspring through the genetic operators;
- Step 8: Repeat steps 5-8 iteratively until the number of generation reaches a pre-fixed value.

VI. RESULTS

The hybrid algorithm scheme has been tested on a variety of processes. In this work, PID controllers will be tuned by the proposed method and compared to other tuning methods for three different plants. The population size, the crossover rate, the mutation rate and the generation number are chosen to be 500, 0.95, 0.05 and 30, respectively. Table I shows the comparison between the Ziegler-Nichols (ZN) method, the Iterative Feedback Tuning (IFT) method, the internal model control

(IMC) method, the integral square error (ISE) method and the proposed method, in which e_{ss} , M_p , T_s and ise denote the percentage of steady-state error, percentage maximum overshoot, the 5% settling time, and the mean of the integral of square error, respectively.

The parameters K_p , T_i and T_d used for tuning the controller PID through ZN, IFT, IMC and ISE have been shown in [8].

A. Example 1:

The plant model is given by Equation 4.

$$G_1(s) = \frac{e^{-5s}}{1 + 20s}. \quad (4)$$

The closed loop step responses obtained for the five PID tuning methods for system G_1 are shown in Fig. 7. The set-

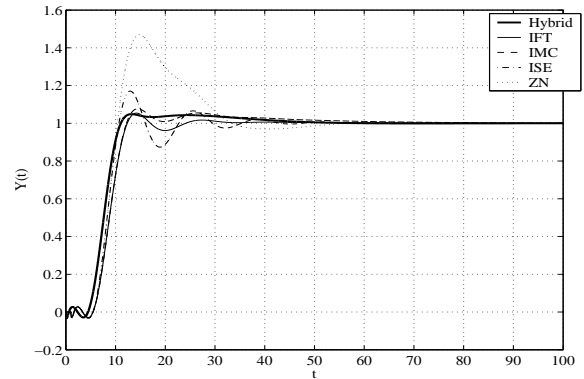


Fig. 7. Step responses of the closed-loop system for $G_1(s)$.

ling time and ise for the hybrid tuning method is significantly smaller than the obtained by other methods.

B. Example 2:

The plant model is given by Equation 5.

$$G_2(s) = \frac{1}{(1 + 10s)^8}. \quad (5)$$

The closed loop step responses obtained for the five PID tuning methods for system G_2 are shown in Fig. 8. The responses for hybrid method, IFT and IMC are almost indistinguishable, but superior than those obtained for the ISE and ZN.

C. Example 3:

The plant model is given by Equation 6.

$$G_3(s) = \frac{(1 - 5s)}{(1 + 10s)(1 + 20s)}. \quad (6)$$

The closed loop step responses obtained for the five PID tuning methods for system G_3 are shown in Fig. 9. The settling time and the ise obtained by applying the hybrid method is significantly shorter than those achieved by the four other schemes.

These results show that a variety of processes can be satisfactorily controlled by the hybrid method proposed here. Generally, it yielded better control performance than other method used in the literature, for the chosen processes.

TABLE I
SUMMARY OF SIMULATION RESULTS.

Process		Hybrid	IFT	IMC	ISE	ZN
$G_1(s)$	e_{ss}	0.00	0.00	0.00	0.00	0.00
	M_p	4.851%	5.403%	7.702%	17.064%	46.945%
	t_s	10.402 s	14.791 s	28.846 s	27.197 s	30.565 s
	ise	0.0681	0.0787	0.0792	0.0768	0.0932
$G_2(s)$	e_{ss}	0.00	0.00	0.00	0.00	0.00
	M_p	4.227%	1.000%	1.075%	17.747%	13.965%
	t_s	103.274 s	125.348 s	237.371 s	358.824 s	213.256 s
	ise	0.1227	0.1372	0.1335	0.1435	0.1458
$G_3(s)$	e_{ss}	0.00	0.00	0.00	0.00	0.00
	M_p	4.035%	0.533%	23.564%	25.456%	53.163%
	t_s	18.215 s	26.754 s	42.837 s	44.718 s	66.561 s
	ise	0.0610	0.0704	0.0722	0.0729	0.0903

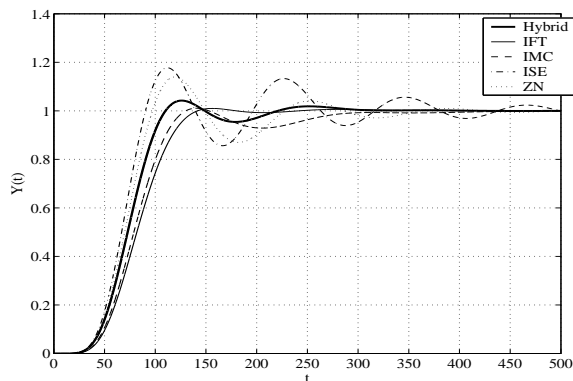


Fig. 8. Step responses of the closed-loop system for $G_2(s)$.

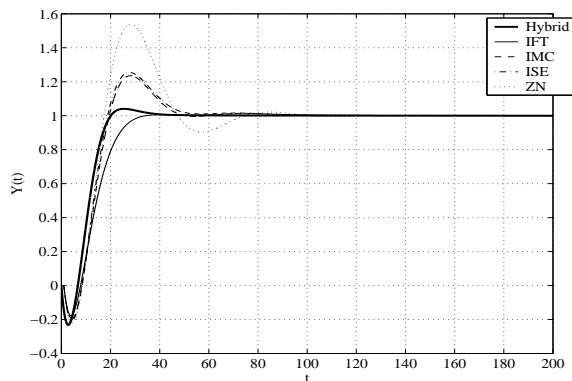


Fig. 9. Step responses of the closed-loop system for $G_3(s)$.

VII. CONCLUSIONS

It is proposed an approach that uses fuzzy rules and genetic programming to determine the PID optimal controller parameters. Human knowledge and experience in control system design are also exploited for this purpose. The performance of this optimal tuning method for PID controller was illustrated, through three representative examples of different situations: high order process, a process with large time delay and a highly non-minimum phase process. The results for the three examples were compared with the obtained for four classical PID tuning methods.

In traditional PID tuning technique, the PID parameters are fixed after being selected in an optimal way. To increase the flexibility and capability of PID controllers, a GP based fuzzy method is proposed in this work to tune the PID controller. Since the PID parameters generated by the proposed approach are expressed by fuzzy rules, they are adaptive and determined by the error signal. A GP is then used to search for the optimal PID parameters corresponding to each fuzzy tuning rules.

REFERENCES

[1] G. M. Almeida, V. V. R. Silva, and E. G. Nepomuceno *PID controller tuning using genetic programming*, Submitted to Computers and Chemical Engineering 2004.

[2] G. M. Almeida, V. V. R. Silva, and E. G. Nepomuceno *Programação Genética em Matlab, Uma aplicação na aproximação de funções matemáticas.*, Anais do Congresso Brasileiro de Automática 2004.

[3] K. J. Astrom, and T. Hagglund *PID Controllers: Theory, Design, and Tuning*, Instruments Society of America, 2nd edition, 1995.

[4] B. Grosman, and D. R. Lewin *Automated nonlinear model predictive control using genetic programming*, Computers and Chemical Engineering, vol 26, pp. 631-640 2002.

[5] G. Hill, and H. Hellendoorn, and J. Teichrow *Fuzzy-C development system - users manual*, , Togai Infraclogic, 30 Corporated Park, Irvine, CA 92714, USA, 1990.

[6] M. P. Hinchliffe, and M. J. Willis *Dynamic systems modelling using genetic programming*, Computers and Chemical Engineering, vol 27, pp. 1841-1854 2003.

[7] B. G. Hu, and G. K. I. Mann, and R. G. Gosine *A Systematic study of fuzzy PID controllers - Function-based evaluation approach*, IEEE Transactions on Fuzzy Systems, vol 9, pp. 699-712 2001.

[8] O. Lequin, and M. G. M. Mossberg, and E. Bosmans, and L. Triest *Iterative feedback tuning of PID parameters: comparison with classical tuning rules*, Control Engineering Practice, vol 11, pp. 1023-1033 2003.

[9] J. R. Koza *An Genetic Programming: On the Programming of Computers by Natural Selection*, MIT Press, Cambridge, MA, 1992.

[10] J. R. Koza, and F. H. Bennett III, and D. Andre, and M. A. Keane *A Synthesis of topology and sizing of analog electrical circuits by means of genetic programming*, Computer Methods in Applied Mechanics and Engineering, vol 186, pp. 459-482 2000.

[11] E. H. Mamdani *Application of fuzzy algorithms for control of simple dynamic plant*, Proc. Inst. Elect. Eng. Contr. Sci., vol 121, pp. 1585-1588 1974.

[12] G. K. I. Mann, and B. G. Hu, and R. G. Gosine *Analysis of direct action*

- fuzzy PID controller structures*, IEEE Transactions on Systems, Man and Cybernetics, vol 29, pp. 371-388 1999.
- [13] C. C. Yu *Auto-tuning of PID Controllers*, Berlin: Springer 1999.
- [14] Z. Y. Zhao, and M. Tomizuka, and S. Isaka *Fuzzy gain scheduling of PID controllers*, IEEE Transactions on Systems, Man and Cybernetics, vol 23, pp. 1392-1398 1993.
- [15] J. G. Ziegler, and N. B. Nichols *Optimum settings for automatic controllers*, Transactions ASME, vol 62, pp. 759-768 1942.