

Design of Gain Scheduled Fuzzy PID Controller

Leehter Yao and Chin-Chin Lin

Abstract—An adaptive fuzzy PID controller with gain scheduling is proposed in this paper. The structure of the proposed gain scheduled fuzzy PID (GS_FPID) controller consists of both fuzzy PI-like controller and fuzzy PD-like controller. Both of fuzzy PI-like and PD-like controllers are weighted through adaptive gain scheduling, which are also determined by fuzzy logic inference. A modified genetic algorithm called accumulated genetic algorithm is designed to learn the parameters of fuzzy inference system. In order to learn the number of fuzzy rules required for the TSK model, the fuzzy rules are learned in an accumulated way. In other words, the parameters learned in the previous rules are accumulated and updated along with the parameters in the current rule. It will be shown that the proposed GS_FPID controllers learned by the accumulated GA perform well for not only the regular linear systems but also the higher order and time-delayed systems.

Keywords—gain scheduling, fuzzy PID controller, adaptive control, genetic algorithm.

I. INTRODUCTION

PID controllers have been widely used in the industry due to the facts that they have simple structures and they assure acceptable performance for the majority of industrial processes. Because of their simple structures, PID controllers are easy to design, operate and maintain. Consequently, PID controllers earn their popularity among practitioners in the industry. Beginning with Ziegler and Nichols's works[1], various parameter tuning methods for conventional PID controllers have been proposed[2]. Ever since fuzzy theories are proposed in [3], fuzzy logic has gradually adopted as one of major approaches for controller design. One of the prominent and efficient ways is applying fuzzy logic to the design of PID controllers. The well-known pioneered and successful example in early stage was the design of a fuzzy PI controller and its practical engineering applications [4-5]. There have been numerous articles investigating different schemes of applying fuzzy logic to the design of PID controllers, which are generally termed as fuzzy PID controllers. Fuzzy PID controllers can be further classified into two types: the gain

scheduling type and the direct action type [6-7].

For the gain scheduling type of fuzzy PID controllers, the PID gains are tuned based on a fuzzy inference system rather than the conventional Ziegler and Nichols's approach. In [8-9], three PID gains K_p , K_i and K_d were respectively calculated through fuzzy logic based on error and error rate. In [10], the PID gains were still calculated based on Ziegler and Nichols's formula, yet the formula was parameterized by a single parameter. A fuzzy inference system was then designed to calculate this single parameter. In order to preserve good load disturbance attenuation, [11] calculated the set-point weight by a fuzzy inference system but calculate three PID gains by conventional Ziegler and Nichols's approach. For the direct action type of fuzzy PID controllers, the control actions are determined directly by means of a fuzzy inference system. This type of fuzzy controllers is in essence like the conventional gain-tuning PID controller. Therefore, they are also called the PID-like controllers. The majority of fuzzy controllers such as [12-18] belong to this class of controllers. Different fuzzy PID structures have been proposed based on all possible contexts of knowledge representations. The performance evaluations and comparison between gain-scheduling type and direct action type fuzzy PID controllers were investigated in [19].

In spite of the significant performance of fuzzy PID controllers over their conventional counterparts, the parameters of fuzzy inference systems for either gain scheduling or direct action type of controllers are generally tuned manually. In order to achieve optimal performance, genetic algorithm (GA) is one of common approaches applied to learn fuzzy PID controller's parameters of fuzzy inference systems. A simple GA and a multi-objective GA are respectively applied in [20] and [21] to find the optimal tuning of parameters in a hybrid fuzzy PI+D controller's fuzzy inference. Similarly, a multi-objective GA is also applied to learn the parameters of fuzzy inference system of a hybrid PD+I controller[22]. In [23], a fuzzy PID controller consisting of one input fuzzy inference system with three rules and six tuning parameters is proposed. A GA-based optimization scheme is applied to learn these parameters.

In this paper, a gain scheduled fuzzy PID (GS_PID) controller, which consists of both PI-like and PD-like fuzzy controllers, is proposed. Both PI-like and PD-like fuzzy controllers are weighted through gain scheduling, which is determined by another fuzzy inference system. The fuzzy inference model employed in this paper is the Takagi-Sugeno-Kang (TSK) model. A modified GA called accumulated GA is proposed to learn both the parameters of membership functions in the antecedent parts and the coefficients of the linear functions in the consequent parts. The

Manuscript received Nov. 5, 2004.

Leehter Yao is with Dept. of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan.(phone: +886-2-2751-0623; fax: +886-2-2751-3892; e-mail: ltyao@ntut.edu.tw).

Chin-chin Lin is with Dept. of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan.(e-mail: s2618019@ntut.edu.tw).

basic idea of the accumulated GA is to learn the fuzzy inference system rule by rule until the learning error is reduced to a range of tolerance. The parameters learned in the previous rules are accumulated and updated along with the parameters in the current rule.

II. GAIN SCHEDULED FUZZY PID CONTROLLER

The structure of proposed GS_FPID controller consists of both fuzzy PI controller and fuzzy PD controller. Both of fuzzy controllers are weighted through scheduled gains, which are also determined by fuzzy logic inference. At each sampling instance k , the model of proposed GS_FPID controller is defined as

$$u(k) = \alpha(k)\Delta u(k) + \beta(k)u(k-1) \quad (1)$$

where $\alpha(k)$, $\beta(k)$ and $\Delta u(k)$ are defined by fuzzy logic inference. As shown in Fig. 1, given that the reference input is denoted by $r(k)$, the error is defined as $e(k) = r(k) - c(k)$ and the error rate $\Delta e(k) = e(k) - e(k-1)$. The gains $\alpha(k)$, $\beta(k)$ and the increments of controller output $\Delta u(k)$ can be respectively determined by the following fuzzy rules:

$$R_1: \text{ if } e(k) \text{ is } A_{i1} \text{ and } \Delta e(k) \text{ is } A_{i2} \\ \text{ then } \alpha(k) = a_{i1}e(k) + a_{i2}\Delta e(k); \quad (2)$$

$$R_2: \text{ if } e(k) \text{ is } A_{i3} \text{ and } \Delta e(k) \text{ is } A_{i4} \\ \text{ then } \beta(k) = a_{i3}e(k) + a_{i4}\Delta e(k); \quad (3)$$

$$R_3: \text{ if } e(k) \text{ is } A_{i5} \text{ and } \Delta e(k) \text{ is } A_{i6} \\ \text{ then } \Delta u(k) = a_{i5}e(k) + a_{i6}\Delta e(k); \quad (4)$$

$i = 1 \dots m$. The Takagi-Sugeno-Kang (TSK) models are used in the fuzzy rules (2)-(4). In the rule sets R_1 , R_2 and R_3 , the sets A_{ij} , $j = 1 \dots 6$, are fuzzy sets for $e(k)$ and $\Delta e(k)$, respectively. For a fuzzy PD controller, the controller output is defined as

$$u(k) = \Delta u_{pd}(k) \quad (5)$$

where $\Delta u_{pd}(k)$ is inferred by fuzzy logic as in (4) based on $e(k)$ and $\Delta e(k)$. The output of a fuzzy PI controller can be defined as

$$u(k) = u(k-1) + \Delta u_{pi}(k) \quad (6)$$

where $\Delta u_{pi}(k)$ is inferred by fuzzy logic as in (4) based on $e(k)$ and $\Delta e(k)$. In addition to improving system's relative stability, fuzzy PD controller is utilized for increasing the speed of response while fuzzy PI controller aims to reduce steady state error. Therefore, if the GS_FPID controller is designed to be a composite of both fuzzy PD and PI controllers, it has both functionalities of these two fuzzy controllers. The proposed fuzzy PID controller is essentially the weighting adjusted fuzzy PD and PI controller. Therefore,

$$u(k) = \alpha_1(k)\Delta u_{pd}(k) + \beta(k)(u(k-1) + \Delta u_{pi}(k)) \\ = (\alpha_1(k)\Delta u_{pd}(k) + \beta(k)\Delta u_{pi}(k)) + \beta(k)u(k-1) \quad (7)$$

where $\alpha_1(k)$ and $\beta(k)$ are respectively the weightings of PD and PI controllers. Since $\Delta u_{pd}(k)$ and $\Delta u_{pi}(k)$ are determined based on $e(k)$ and $\Delta e(k)$ by fuzzy logic inference as in (4), they can be simplified as one variable. Therefore, the first part in (7) is simplified as

$$\alpha_1(k)\Delta u_{pd}(k) + \beta(k)\Delta u_{pi}(k) = \alpha(k)\Delta u(k) \quad (8)$$

where $\alpha(k) = \alpha_1(k) + \beta(k)$. With the simplification in (8), the composite of fuzzy PD and PI controllers in (7) is thus given as in (1).

Within GS_FPID controller, the weightings of fuzzy PD and PI controller, i.e., $\alpha(k)$ and $\beta(k)$, are continuously changed

based on the error $e(k)$ and the error rate $\Delta e(k)$. GS_FPID controller can thus be considered as an adaptive controller with gain scheduling. It continuously adjusts the gains of fuzzy PD and PI controller based on error and error rate. The gain scheduling mechanism is designed to be the one with fuzzy inference scheme.

For TSK model, the output of FLC in (2)-(4) is calculated as:

$$\alpha(k) = \sum_{i=1}^m r_{\alpha i} (a_{i1}e(k) + a_{i2}\Delta e(k)), \quad (9)$$

$$\beta(k) = \sum_{i=1}^m r_{\beta i} (a_{i3}e(k) + a_{i4}\Delta e(k)), \quad (10)$$

$$\Delta u(k) = \sum_{i=1}^m r_{ui} (a_{i5}e(k) + a_{i6}\Delta e(k)), \quad (11)$$

where

$$r_{\alpha i} = \frac{\tau_{\alpha i}}{\sum_{i=1}^m \tau_{\alpha i}}, \quad (12)$$

$$r_{\beta i} = \frac{\tau_{\beta i}}{\sum_{i=1}^m \tau_{\beta i}}, \quad (13)$$

$$r_{ui} = \frac{\tau_{ui}}{\sum_{i=1}^m \tau_{ui}}. \quad (14)$$

The degrees of firing (DOF) in (12)-(14) are respectively defined as:

$$\tau_{\alpha i} = A_{i1}(e(k)) \wedge A_{i2}(\Delta e(k)), \quad (15)$$

$$\tau_{\beta i} = A_{i3}(e(k)) \wedge A_{i4}(\Delta e(k)), \quad (16)$$

$$\tau_{ui} = A_{i5}(e(k)) \wedge A_{i6}(\Delta e(k)), \quad (17)$$

where the notation \wedge denotes the operation of minimization.

The FLCs in (9)-(11) can be viewed as PI-like controllers since

$$\alpha(k) = k_{P\alpha}e(k) + k_{I\alpha}\Delta e(k), \quad (18)$$

$$\beta(k) = k_{P\beta}e(k) + k_{I\beta}\Delta e(k), \quad (19)$$

$$\Delta u(k) = k_{Pu}e(k) + k_{Iu}\Delta e(k), \quad (20)$$

$$\text{with } k_{P\alpha} = \sum_{i=1}^m r_{\alpha i} a_{i1}, \quad (21)$$

$$k_{I\alpha} = \sum_{i=1}^m r_{\alpha i} a_{i2}, \quad (22)$$

$$k_{P\beta} = \sum_{i=1}^m r_{\beta i} a_{i3}, \quad (23)$$

$$k_{I\beta} = \sum_{i=1}^m r_{\beta i} a_{i4}. \quad (24)$$

$$k_{Pu} = \sum_{i=1}^m r_{ui} a_{i5}, \quad (25)$$

$$k_{Iu} = \sum_{i=1}^m r_{ui} a_{i6}. \quad (26)$$

Substituting (18)-(20) into (1) yields

$$\begin{aligned} u(k) &= (k_{P\alpha}e(k) + k_{I\alpha}\Delta e(k))(k_{Pu}e(k) + k_{Iu}\Delta e(k)) \\ &\quad + (k_{P\beta}e(k) + k_{I\beta}\Delta e(k))u(k-1) \\ &= k_{P\alpha}k_{Pu}e^2(k) + (k_{P\alpha}k_{Iu} + k_{Pu}k_{I\alpha})e(k)\Delta e(k) \quad (27) \\ &\quad + k_{I\alpha}k_{Iu}(\Delta e(k))^2 + k_{P\beta}e(k)u(k-1) \\ &\quad + k_{I\beta}\Delta e(k)u(k-1). \end{aligned}$$

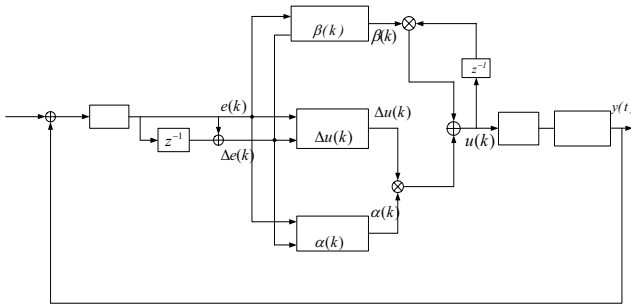


Fig. 1 Structure of GS_FPID controller

It is shown in (27) that the output of GS_FPID controller, $u(k)$, is no longer like a traditional PID controller, which is a linear function of $e(k)$, $\Delta e(k)$ and $u(k-1)$. Instead, it is a complex recurrent function of $e^2(k)$, $e(k)\Delta e(k)$, $(\Delta e(k))^2$, $e(k)u(k-1)$ and $\Delta e(k)u(k-1)$. Referring to (21)-(26), GS_FPID thus has more degrees of freedom adapting with these influences to the controller output. With the additional degrees of freedom, it will be shown in this paper that GS_FPID performs well for controlling the higher order systems as well as the systems with time delays.

III. LEARNING OF GS_FPID CONTROLLER

The membership functions of antecedent fuzzy set A_{ij} , $i = 1 \dots m$, $j = 1 \dots 6$, for input variables $e(k)$ or $\Delta e(k)$ of FLC in (2)-(4) are defined as symmetric triangular functions $\mu_{A_{ij}}(x_j; L_{ij}, C_{ij})$, of which L_{ij} and C_{ij} respectively denotes the width and center of the triangular membership function. For GS_FPID controller, GA is applied to learn the parameters of input membership functions as well as the coefficients of the first order FIR filter in the consequent of each fuzzy rule. The parameters to be learned are expressed as binary strings and cascaded as a chromosome. The number of fuzzy rules in FLC for α , β and Δu are assumed to be the same in GS_FPID controller. However, the number of fuzzy rules is not known a priori. In order to learn the number of fuzzy rules as well as the parameterizations of each fuzzy rule, the fuzzy rule is iteratively learned and accumulated. In other words, a single fuzzy rule for α , β and Δu , respectively, is first utilized. GA is then applied to learn the parameterizations in both antecedent and consequent parts of the rule. As the learning of GA converges, a second rule is added based on the previously learned fuzzy rule continuing the learning process. The second learning process is not restricted to simply learn the second

fuzzy rule. In fact, the first and second fuzzy rules are both learned by GA in the same learning process. In order not to waste the efforts that has been spent on learning the previous fuzzy rule, the parameters of previously learned fuzzy rule are taken as the initial conditions for the GA so that the learning process can be greatly improved. Similarly, in the third learning process, the parameters to be learned include the ones in the first, second and third rules. As the learning processes go on, the learning error in each learning process is a decreasing sequence given that the GA in each learning process converges.

Denote the GA applied in the i -th learning process by GA_i and assume that it takes G_i generations for GA_i to converge. Furthermore, assume that L chromosomes are employed in every generation of GA. Let d_{ijg} be the parameters of the i -th fuzzy rule to be learned, which is implemented as part of the g -th chromosome of GA's gene pool in the j -th generation, $g = 1 \dots L$, $j = 1 \dots G_i$. That is to say, d_{ijg} is the additional set of parameters to be learned in GA_i . If the set D_{ijg} denotes GA_i 's whole set of parameters implemented in the g -th chromosome of the j -th generation, then the total number of parameters to be learned is accumulated process by process, i.e.,

$$D_{ijg} = [D_{(i-1)jg}, d_{ijg}], \quad g = 1 \dots L, j = 1 \dots G_i. \quad (28)$$

Note that for the initial learning process, $D_{1jg} = d_{1jg}$. Referring to (2)-(4), let L_{ijg}^p and C_{ijg}^p respectively be the width and center of triangular membership functions of fuzzy sets A_{ip} , a_{ijg}^p be the p -th parameter in the consequent of fuzzy rule, which are implemented in the g -th chromosome of GA_i 's j -th generation, $p = 1 \dots 6$. Then, d_{ijg} can be defined as

$$d_{ijg} = [L_{ijg}^1, \dots, L_{ijg}^6, C_{ijg}^1, \dots, C_{ijg}^6, a_{ijg}^1, \dots, a_{ijg}^6]. \quad (29)$$

Let $y(k; D_{ijg})$ be the sampled system output due to the fuzzy PID controller associated with D_{ijg} . For GA_i , the fitness function associated with the chromosome D_{ijg} is defined as the integration of square error (ISE):

$$f_{ijg} = \sum_{k=1}^{N_1} \rho_1 (r(k) - y(k; D_{ijg}))^2 + \sum_{k=N_1+1}^N \rho_2 (r(k) - y(k; D_{ijg}))^2 \quad (30)$$

where N denotes the total number of samples to be evaluated. In order to make the designed GS_FPID controller hold good transient response, (30) allows the flexibility of assigning the first N_1 samples different weightings from the rest of the samples. Within N samples in total, N_1 samples are given the weighting ρ_1 while the rest of them are given the weighting ρ_2 . The accumulated GA tends to adjust the parameters of GS_FPID controller minimizing the ISE. The best chromosome learned by GA is determined by searching all possible chromosomes minimizing f_{ijg} . The elitist method is utilized to choose parent chromosomes to mate and generate children chromosomes. In order to increase GA's searching efficiency, the parameters learned in the previous learning process are given as the initial conditions. If D_{i-1}^* denotes the

best chromosome learned by GA_{i-1} , then the best chromosome in the j -th generation of GA_i can be determined as

$$\mathbf{D}_{ij}^* = \underset{\mathbf{D}_{ijg}, g=1..L}{\text{Arg min}} (f_{ijg}^*) \Big|_{\mathbf{D}_{i-1}^*} \quad (31)$$

In order to guarantee the fitness function can be reduced generation by generation, the best chromosome in the current generation is passed to the next generation. With the elitist method for choosing parent chromosomes, the variations of fitness values for GA_i is a non-increasing sequence. If f_{ij}^* denotes the fitness function associated with \mathbf{D}_{ij}^* , the best chromosome determined by GA_i can thus be determined as

$$\mathbf{D}_i^* = \underset{\mathbf{D}_{ij}^*, j=1..G_i}{\text{Arg min}} (f_{ij}^*) \quad (32)$$

Since the sequence of f_{ij}^* is a nonincreasing sequence, \mathbf{D}_i^* is the best chromosome learned in the final generation, i.e., $\mathbf{D}_i^* = \mathbf{D}_{iG_i}^*$. In (31), although $6 \times i$ parameters are to be learned in the i -th learning process, $6 \times (i-1)$ best parameters learned in the previous learning process, denoted by \mathbf{D}_{i-1}^* , can be taken as the initial values in the initial generation of GA_i . In addition, $6 \times (i-1)$ parameters in each of the other $(L-1)$ chromosomes in the initial generation of GA_i can also be obtained by mating \mathbf{D}_{i-1}^* with the other randomly generated chromosomes.

In order to determine the appropriate number of fuzzy rules for α , β and Δu , GAs with increasing size of chromosomes are iteratively applied to learn the parameters in each fuzzy rule. For each learning process, a new fuzzy rule is added associated with the previously learned fuzzy rules. Since the sequence f_{ij}^* , $j = 1..G_i$, is a non-increasing sequence, if the set of parameters \mathbf{D}_{ij}^* cannot further reduce the fitness function, the convergence of fitness values will stagnate. Within each GA, the convergence of fitness values is monitored. If the fitness values do not change for certain number of generations, and the fitness value is still larger than a threshold, a new GA is applied.

IV. ROBUSTNESS OF GS_FPID CONTROLLER

In some applications, it is known *a priori* how the dynamics of a controlled system change with the operating conditions of the system. The controller can thus be designed to change the parameters accommodating the variations of system dynamics based on the measurements of system operating conditions. This type of feedback control scheme is called gain scheduling.[8] The proposed GS_FPID controller can also be applied to solving the gain scheduling problems. In order to design suitable scheduling parameters based on system operating conditions, traditional controller with gain scheduling requires system dynamics. However, GS_FPID requires no information of system dynamics. The operating conditions for the GS_FPID controller simply come from the measurements of error $e(\cdot)$ and error rate $\Delta e(\cdot)$. On the basis of

$e(\cdot)$ and $\Delta e(\cdot)$, GS_FPID controller adjusts controller parameters by fuzzy inference.

Assume that the variations of system dynamics with operating conditions are known. If q samples of variations are taken, then, with $n = 1..q$, the system outputs can be represented as

$$y_n(t) = \bar{d}(\mathbf{x}(t), u(t), t) + \Delta d_n(\mathbf{x}(t), u(t), t), \quad (33)$$

where $\mathbf{x}(\cdot)$ is the vector of state variables, $u(\cdot)$ is the input, $\bar{d}(\cdot)$ is a fixed system function and $\Delta d_n(\cdot)$ is due to the n -th variation of system dynamics. With self tuning of fuzzy gains $\alpha(k)$ and $\beta(k)$, robustness is embedded in the GS_FPID controller. As the dynamics of controlled system varies, GS_FPID controller is able to accommodate the variations of system dynamics with self tuning of fuzzy gains $\alpha(k)$ and $\beta(k)$. In order to learn the GS_FPID controller for gain scheduling, the fitness function in (30) can be modified accommodating the variations of system dynamics. Let $y_n(k; \mathbf{D}_{ijg})$ be the sampled system output due to n -th variation of system dynamics in association with the parameters of GS_FPID controller, \mathbf{D}_{ijg} . Similar to (30), the fitness function associated with the chromosome \mathbf{D}_{ijg} can be modified as

$$f_{ijg} = \sum_{n=1}^q \rho_1 \left(\sum_{k=1}^{N_1} (r(k) - y_n(k; \mathbf{D}_{ijg}))^2 + \rho_2 \sum_{k=N_1+1}^N (r(k) - y_n(k; \mathbf{D}_{ijg}))^2 \right) \quad (34)$$

where q denotes the number of known system variations. The best chromosome minimizing (34) by GA_i is obtained as in (31) and (32). Referring to (34), the best GS_FPID controller learned by the accumulated GA is able to adaptively change the gains adjusting the weightings of fuzzy PI and fuzzy PD controllers to accommodate the variations of system dynamics.

V. COMPUTER SIMULATION

In this section, the accumulated GA are iteratively applied to design the GS_FPID controller for the higher order, variable structure or time delayed system. The fitness function is designed in the form of ISE. For each GA, 80 chromosomes are utilized in each generation. For the fitness function in (30), $N_1 = 50$, $\rho_1 = 5$, $\rho_2 = 1$;

Example 1. Assume that the system to be controlled is with variable system dynamics as following:

$$M(s) = \frac{2s+1}{s^2+as+1}, \quad (36)$$

where $a = 0.9, 1.2$, and 1.5 is known *a priori*. The sampling interval is set to be 0.01 second and 1500 samples are taken for evaluation in the fitness function. The step responses due to GS_FPID controller with 2 fuzzy rules are shown in Fig. 2. It is shown in Fig. 2 that the GS_FPID controller is able to accommodate the variations of system dynamics.

Example 2. Assume that the system to be controlled is with high order time delay variable system dynamics as following:

$$M(s) = \frac{e^{-Ls}}{(s+1)(s+0.5)(s+0.25)(s+0.125)} \quad (37)$$

where $L = 1, 2,$ and 3 is known *a priori*. The sampling interval is set to be 0.1 second and 200 samples are taken for evaluation in the fitness function. The step responses due to GS_FPID controller with simply one fuzzy rule are respectively shown in Fig. 3. It is shown in Fig. 3 that the GS_FPID controller is able to accommodate time delay variations.

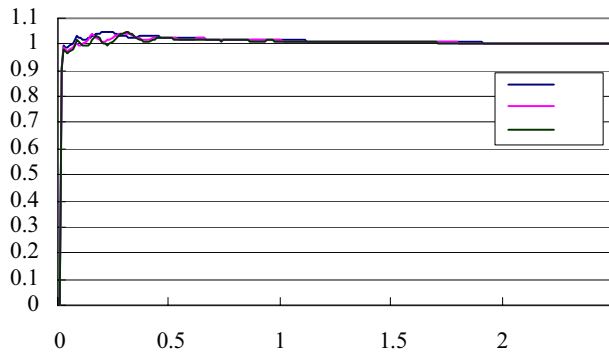


Fig. 2. Step response of Example 1.

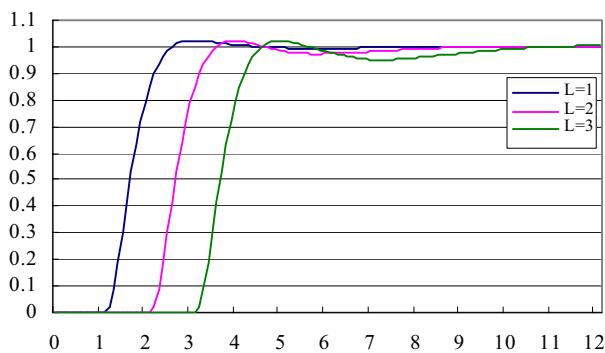


Fig. 3. Step response of Example 2.

VI. CONCLUSIONS

A GS_FPID controller based on the accumulated GA is proposed in this paper. The parameters as well as the required number of fuzzy rules in the fuzzy inference system are simultaneously learned by the accumulated GA. It is shown that the proposed GS_FPID controller performs well for variable structure or time delayed system. In order to design suitable scheduling parameters based on system operating conditions, traditional controller with gain scheduling requires system dynamics. However, GS_FPID requires no information of system dynamics. The operating conditions for the GS_FPID controller simply come from the measurements of error $e(\cdot)$ and error rate $\Delta e(\cdot)$.

The proposed fuzzy PID controller can not only be considered as an adaptive fuzzy PID controller adapting with varying system dynamics, but also can be considered as a regular fuzzy PID controller with more flexibility. The proposed gain scheduled fuzzy PID controller performs well when controlling the system without varying dynamics. In fact,

since the proposed gain scheduled fuzzy PID controller is with more parameterization degree of freedom, it can be utilized to control more complex systems which generally cannot be controlled well by the regular fuzzy PID controllers.

REFERENCES

- [1] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controller," *ASME Trans.*, Vol. 64, pp. 756-768, Jun. 1942.
- [2] A. Datta, M. T. Ho and S. P. Bhattacharyya, *Structure and Synthesis of PID Controllers*, London, U.K.:Springer-Verlag, 2000.
- [3] L. A. Zadeh, "Fuzzy sets," *Inform. Contr.*, Vol. 8, pp. 338-353, June. 1965.
- [4] E. H. Mamdani, "Application of fuzzy algorithms for simple dynamic plant," *Proc. Inst. Elect. Eng.*, Vol. D-122, pp. 1585-1588, 1974.
- [5] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Math. Stud.*, Vol. 7, pp. 1-13, 1975.
- [6] G. K. I. Mann, B. G. Hu and R. G. Gosine, "Analysis of direct action fuzzy PID controller structures," *IEEE Trans. SMC - Pt. B*, Vol. 29, pp. 371-388, Jun. 1999.
- [7] G. K. I. Mann, B. G. Hu and R. G. Gosine, "A systematic study of fuzzy PID controllers - function-based evaluation approach," *IEEE Trans. Fuzzy Syst.*, Vol. 9, pp. 699-712, Oct. 2001.
- [8] Z.-Y. Zhao, M. Tomizuka and S. Isaka, "Fuzzy gain scheduling of PID controllers," *IEEE Trans. Syst., Man, Cybern.*, Vol. 23, pp. 1392-1398, 1993.
- [9] S. G. Tzafestas and N. P. Papanikolopoulos, "Incremental fuzzy expert PID control," *IEEE Trans. Ind. Electron.*, Vol. 37, No. 5, pp. 365-371, 1990.
- [10] S.-Z. He, S. Tan and F.-L. Xu, "Fuzzy self-tuning of PID controllers," *Fuzzy Sets Syst.*, pp. 37-46, 1993.
- [11] A. Visioli, "Fuzzy logic based set-point weighting for PID controllers," *IEEE Trans. Syst. Man, Cybern. - Pt. A*, Vol. 29, pp. 587-592, 1999.
- [12] M. Maeda and S. Murakami, "Fuzzy gain scheduling of PID controllers," *Fuzzy Sets Syst.*, Vol. 51, pp. 29-40, 1992.
- [13] C.-L. Chen, P.-C. Chen and C. K. Chen, "Analysis and design of fuzzy control systems," *Fuzzy Sets Syst.*, Vol. 57, pp. 125-140, 1993.
- [14] W. Li, "Design of a hybrid fuzzy logic proportional plus conventional integral-derivative controller," *IEEE Trans. Fuzzy Syst.*, Vol. 6, No. 4, pp. 449-463, 1998.
- [15] B. Armstrong, "FLC design for bounded separable functions with linear input-output relations as a special case," *IEEE Trans. Fuzzy Syst.*, Vol. 4, pp. 72-79, 1996.
- [16] H. A. Malki, D. Misir, D. Feigenspan and G. Chen, "Fuzzy PID control of a flexible-joint robot arm with uncertainties from time-varying loads," *IEEE Trans. Contr. Syst. Technol.*, Vol. 5, pp. 371-378, 1997.
- [17] D. Misir, H. A. Malki and G. Chen, "Design and analysis of a fuzzy proportional integral derivative controller," *Fuzzy Sets Syst.*, Vol. 79, pp. 297-314, 1996.
- [18] S. J. Qin and G. Borders, "A multiregion fuzzy logic controller for nonlinear process control," *IEEE Trans. Fuzzy Syst.*, Vol. 2, pp. 74-81, 1994.
- [19] A. Visioli, "Tuning of PID controllers with fuzzy logic," *IEE Proc. - Control Theory Appl.*, Vol. 148, No. 1, pp. 1-8, 2001.
- [20] M. J. Er and Y. L. Sun, "Hybrid fuzzy proportional-integral plus conventional derivative control of linear and nonlinear systems," *IEEE Trans. Ind. Electron.*, Vol. 48, No. 6, pp. 1109-1117, Dec. 2001.
- [21] K. S. Tang, K. F. Man, G. Chen and S. Kwong, "An optimal fuzzy PID controller," *IEEE Trans. Ind. Electron.*, Vol. 48, No. 4, pp. 757-765, Aug. 2001.
- [22] K. S. Tang, K. F. Man, G. Chen and S. Kwong, "A GA-optimized fuzzy PD+I controller for nonlinear systems," in *Proc. IEEE IECON'01*, Denver, Colorado, Nov. 29-Dec. 2, 2001, pp. 718-723.
- [23] B. Hu, G. K. I. Mann and R. G. Gosine, "New methodology for analytical and optimal design of fuzzy PID controllers," *IEEE Trans. Fuzzy Syst.*, Vol. 7, No. 5, pp. 521-539, Oct. 1999.