

# Fuzzy Control Theory: A Nonlinear Case\*

HAO YING,<sup>†</sup> WILLIAM SILER<sup>‡</sup> and JAMES J. BUCKLEY<sup>§</sup>

*A simplest possible fuzzy controller is proved analytically to be equivalent to a proportional-integral controller when a linear defuzzification algorithm is used or to a nonlinear proportional-integral controller when a nonlinear defuzzification algorithm is used.*

**Key Words**—Fuzzy control; nonlinear control systems; PID control.

**Abstract**—Sources of nonlinearity in a fuzzy controller include the fuzzification algorithm used for the controller inputs; fuzzy control rules; type of fuzzy logic used for evaluating the fuzzy control rules; and the defuzzification algorithm used for the controller output. We analyze the performance of a simple fuzzy controller with linear and nonlinear defuzzification algorithms. We prove theoretically that such a fuzzy controller, the smallest possible, with two inputs (error and rate change of error) and a nonlinear defuzzification algorithm is equivalent to a nonlinear proportional-integral (PI) controller with proportional-gain and integral-gain changing with error and rate change of error about a setpoint. Furthermore, this fuzzy controller is precisely equivalent to a conventional linear PI controller if a linear defuzzification algorithm is employed. Computer simulation showed that the performance of the fuzzy controller was almost the same as that of the PI controller when first-order and second-order linear processes were used. Furthermore, the fuzzy controller was significantly better when a first-order with a time delay model was used. More importantly, the simulated result illustrated that the fuzzy controller was stable when a nonlinear process model was controlled, but the PI controller was unstable.

## 1. INTRODUCTION

AS AN IMPORTANT branch of fuzzy set theory (Zadeh, 1965), fuzzy controllers have been developed and several applications have been achieved (Mamdani and Gaines, 1981; Sugeno, 1985) since the invention of the first fuzzy controller in 1974 (Mamdani, 1974). However, a theory for the fuzzy controller has been badly needed.

In this paper we establish the relationship between the PI controller and a smallest possible fuzzy controller, consisting of two inputs, error and rate of change of error ("rate" for short),

\* Received 9 July 1988; revised 2 February 1989; revised 23 May 1989; received in final form 31 August, 1989. The original version of this paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Editor A. P. Sage.

<sup>†</sup>Kemp-Carraway Heart Institute, Birmingham, AL 35234, U.S.A. and Department of Biomedical Engineering, School of Engineering, University of Alabama at Birmingham, Birmingham, AL 35294, U.S.A. Author to whom all correspondence should be addressed.

<sup>‡</sup>Mote Marine Laboratory, Sarasota, FL 33577, U.S.A.

<sup>§</sup>Mathematics Department, University of Alabama at Birmingham, Birmingham, AL 35294, U.S.A.

one output and four fuzzy control rules. The fuzzy sets "error" and "rate" have two fuzzy members, and the fuzzy set for controller output has three fuzzy members. We will focus on the nonlinearity introduced by the defuzzification algorithm. We will prove theoretically that a fuzzy controller with a nonlinear defuzzification algorithm, but otherwise linear, is precisely equivalent to a nonlinear PI controller where proportional-gain and integral-gain change with error and rate. Here we define that a fuzzy controller is a linear fuzzy controller if the defuzzified output of the fuzzy controller is a linear function of its inputs.

## 2. THEORETICAL ANALYSIS OF THE FUZZY CONTROLLER WITH TWO INPUTS

### 2.1. Description of a typical fuzzy controller

A typical fuzzy controller can be described by the block diagram in Fig. 1.

Most fuzzy controllers developed so far employ error and rate about a setpoint as their inputs. In this paper, we employ the following notation;

$$\begin{aligned}e(nT) &= y(nT) - \text{setpoint} \\e \sim (nT) &= F[GE * e(nT)] \\r(nT) &= [e(nT) - e(nT - T)]/T \\r \sim (nT) &= F[GR * r(nT)] \\u(nT) &= du(nT) + u(nT - T) \\&= GU * dU(nT) + u(nT - T)\end{aligned}$$

where  $n$  is a positive integer and  $T$  is a sampling period. The  $e(nT)$ ,  $r(nT)$ ,  $y(nT)$  and  $u(nT)$  denote error, rate, process output and output from the fuzzy controller (which is also process input) at sampling time  $nT$ . The  $e(nT - T)$  and  $u(nT - T)$  are the error and the output from the fuzzy controller at sampling time  $nT - T$ , respectively.  $GE$  (gain for error) is the input

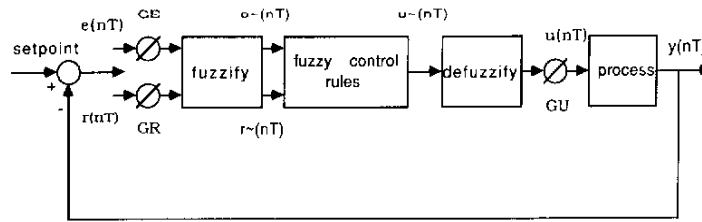


FIG. 1. Block diagram of a typical fuzzy controller.

scaler for error, GR (gain for rate) is the input scaler for rate and GU (gain for controller output) is the output scaler of the fuzzy controller.  $F[\ ]$  means fuzzification. The  $du(nT)$  designates the incremental output of fuzzy controller (to process input) at sampling time  $nT$ . The  $dU(nT)$  denotes the incremental output of the fuzzy controller from defuzzifying the fuzzy set, "output", at sampling time  $nT$ .  $u \sim (nT)$  denotes the fuzzy set corresponding to  $u(nT)$ .  $e \sim (nT)$  [  $r \sim (nT)$  ] denotes the fuzzy set corresponding to scaled error  $GE * e(nT)$  [scaled rate  $GR * r(nT)$ ].

The components of a typical fuzzy controller include:

- (1) the input scalers GE, GR and the output scaler GU;
- (2) a fuzzification algorithm for scaled error, scaled rate and output;
- (3) the fuzzy control rules;
- (4) fuzzy logic used for evaluating the fuzzy control rules; and
- (5) a defuzzification algorithm used for getting crisp output from the fuzzy set "output".

The sources of nonlinearity of a fuzzy controller include:

- (1) the fuzzification algorithm used for the controller inputs;
- (2) the fuzzy control rules;
- (3) type of fuzzy logic used for evaluating the fuzzy control rules; and
- (4) the defuzzification algorithm used for the controller output.

2.2. Fuzzification algorithm and fuzzy control rules

The linear fuzzification algorithm, for scaled error and rate, is shown in Fig. 2.

The fuzzy set "error" has two members, e.p

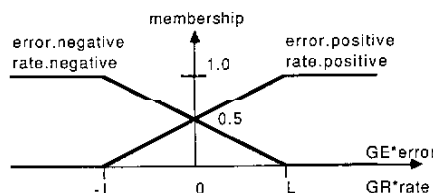


FIG. 2. Fuzzification algorithm for the inputs of the fuzzy controller, scaled error and rate change of error.

(denotes error.positive) and e.n (denotes error.negative); and the fuzzy set "rate" has two members, r.p (denotes rate.positive) and r.n (denotes rate.negative). The definition of the fuzzy set "output" is shown in Fig. 3. The fuzzy set "output" has three members, o.p (denotes output.positive), o.z (denotes output.zero) and o.n (denotes output.negative). Actually, since the grades of membership of the members of fuzzy set "output" are computed from the fuzzy control rules, no fuzzification of controller output is needed. However, the definition of "output" is necessary for defuzzification, as we will show below.

There are four linear fuzzy control rules used in this study:

if error = e.p and rate = r.p then output = o.n (r1)

if error = e.p and rate = r.n then output = o.z (r2)

if error = e.n and rate = r.p then output = o.z (r3)

if error = e.n and rate = r.n then output = o.p. (r4)

2.3. Fuzzy logic for evaluation of the fuzzy control rules

The fuzzy logics with which we are concerned are those of Zadeh and of Lukasiewicz. If  $\mu_A$  and  $\mu_B$  represent the grades of membership of an object in fuzzy sets A and B, respectively, then these logics are defined as:

Zadeh logic:

$AND(\mu_A, \mu_B) = \min(\mu_A, \mu_B)$

$OR(\mu_A, \mu_B) = \max(\mu_A, \mu_B)$ ;

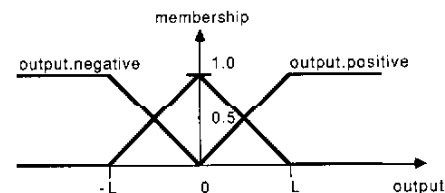


FIG. 3. Fuzzification algorithm for the incremental output of the fuzzy controller.

Lukasiewicz logic:

$$\text{AND}(\mu_A, \mu_B) = \max(0, \mu_A + \mu_B - 1)$$

$$\text{OR}(\mu_A, \mu_B) = \min(1, \mu_A + \mu_B)$$

In evaluating the control rules, it is proper to use the Zadeh AND to evaluate the individual control rules, and the Lukasiewicz OR to evaluate the implied OR between control rules r2 and r3. The Lukasiewicz OR should be used since the conditions being ORd are maximally negatively correlated (Siler and Ying, 1988). However, for the smallest fuzzy controller here, with the linear defuzzification below, exactly the same controller output will be obtained by using the Zadeh AND and either the Zadeh or Lukasiewicz OR, since the (different) grades of membership of o.z are multiplied by zero in the defuzzification process. But when the nonlinear defuzzification process described below is used, different results will be obtained from the Zadeh and Lukasiewicz ORs. In the interest of brevity, we will give the equations for Lukasiewicz OR, since it is the correct logic to use. However, the simulation results will be shown for both logics.

2.4. Defuzzification algorithms

The linear defuzzification algorithm used is

$$d_L = \frac{\sum[(\text{value of member}) * (\text{membership of member})]}{\sum(\text{memberships})} \quad (1)$$

where  $d_L$  is the defuzzified equivalent of the fuzzy set. The value, used in the defuzzification algorithm, is the value for the members of the fuzzy set "output" which are chosen, in our study, as those values for which the grade of membership in  $[-L, L]$  is unity. These values are 1 for the fuzzy member o.p, 0 for the fuzzy member o.z and  $-L$  for the fuzzy member o.n as shown in Fig. 3. As we have shown in Siler and Ying (1988) and Buckley and Ying (1988a), the resulting fuzzy controller is precisely equivalent to a linear nonfuzzy controller.

A nonlinear defuzzification procedure was also used, which amounted to a normalization of the grades of membership of the members of the fuzzy set being defuzzified to a sum of one. The defuzzified output of a fuzzy set is defined as

$$d_{NL} = \frac{\sum[(\text{value of member}) * (\text{membership of member})]}{\sum(\text{memberships})} \quad (2)$$

where  $d_{NL}$  is the defuzzified equivalent of the fuzzy set. The nonlinearity is introduced into the defuzzification algorithm by the denominator.

2.5. Proof that a fuzzy controller with the nonlinear defuzzification is equivalent to a nonfuzzy nonlinear PI controller

The control rules r1-r4 all employ the Zadeh AND of two conditions, one on the scaled rate, and the other on the scaled error itself. Since the Zadeh AND is the minimum of two values, two different conditions arise for each rule, one when the scaled error is less than the scaled rate, and one when the scaled rate is less than the scaled error. The eight different combinations of scaled error and scaled rate constituting input to the control rules are shown graphically in Fig. 4.

To prove that the fuzzy controller constructed above is a conventional nonlinear PI controller with changing proportional-gain and integral-gain, all combinations of the inputs, scaled error and rate shown in Fig. 4, must be considered. The results of evaluating the fuzzy control rules r1, r2, r3 and r4, when scaled error and rate are in  $[-L, L]$ , are given in Table 1. E.p and E.n (R.p and R.n) in Table 1 mean the membership values of e.p and e.g (r.p and r.n) in the fuzzy sets error (rate). For example, for given values of scaled error  $GE * e(nT)$  and rate  $GR * r(nT)$ , the membership values obtained by using the fuzzification algorithm shown in Fig. 2 are E.p and R.p. Then, say in rule r1, the membership value associated with the member, o.n, of the fuzzy set "output" is the  $\text{Min}(E.p, R.p)$ . In this way, the membership values listed in Table 1 can be obtained. Notice that

$$E.p = [GE * e(nT) + L]/2L \quad (3)$$

$$E.n = [L - GE * e(nT)]/2L \quad (4)$$

$$R.p = [GR * r(nT) + L]/2L \quad (5)$$

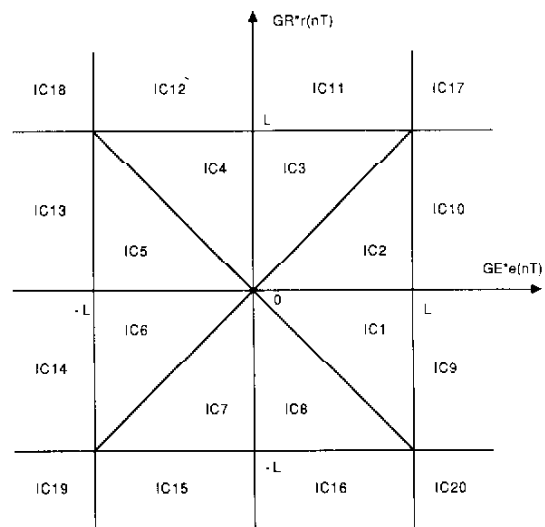


FIG. 4. Possible input combinations (IC) of scaled error and rate change of error which must be considered when the fuzzy controller rules are evaluated as shown in Table 1.

TABLE 1. RESULTS OF EVALUATING THE FUZZY CONTROL RULES r1, r2, r3 AND r4 FOR ALL COMBINATIONS OF INPUTS USING ZADEH AND (min) FUZZY LOGIC WHEN SCALED ERROR AND RATE OF PROCESS OUTPUT ARE WITHIN THE INTERVAL  $[-L, L]$  OF THE FUZZIFICATION ALGORITHM. THE INPUT COMBINATIONS OF SCALED ERROR AND RATE ARE SHOWN GRAPHICALLY IN FIG. 4

Input Combinations (IC) of scaled error and rate as shown in Fig. 4	Memberships obtained by evaluating fuzzy control rules r1, r2, r3 and r4 using Zadeh AND (min) fuzzy logic			
	r1	r2	r3	r4
IC1	R.p	R.n	E.n	E.n
IC2	R.p	R.n	E.n	E.n
IC3	E.p	R.n	E.n	R.n
IC4	E.p	R.n	E.n	R.n
IC5	E.p	E.p	R.p	R.n
IC6	E.p	E.p	R.p	R.n
IC7	R.p	E.p	R.p	E.n
IC8	R.p	E.p	R.p	E.n

$$R.n = [L - GR * r(nT)] / 2L \quad (6)$$

$$E.p + E.n = 1 \quad (7)$$

$$R.p + R.n = 1. \quad (8)$$

Also, be aware that in our case the Lukasiewicz OR reduces to the sum of the grades of membership being ORd, since this sum can never be greater than one for the simple fuzzy controller used in this paper.

The main theoretical results follow:

(A) When the Lukasiewicz OR is used for the membership of the member  $o.z$  of the fuzzy set "output", the incremental output of the fuzzy controller at sampling time  $nT$ ,  $du_{NL}(nT)$ , can be described by following two equations obtained from the nonlinear defuzzification algorithm ( $d_{NL}$ ):

$$\text{If } GR * |r(nT)| \leq GE * |e(nT)| \leq L,$$

$$du_{NL}(nT) = - \frac{0.5 * L * GU}{2L - GE * |e(nT)|} [GF * e(nT) + GR * r(nT)]. \quad (9)$$

$$\text{If } GE * |e(nT)| \leq GR * |r(nT)| \leq L,$$

$$du_{NL}(nT) = - \frac{0.5 * L * GU}{2L - GR * |r(nT)|} [GE * e(nT) + GR * r(nT)]. \quad (10)$$

Equations (9) and (10) may be verified by substituting the values for E.p through E.n [equations (3) through (6) into Table 1] and evaluating  $d_{NL}$  for each row in Table 1. Since the incremental output of the fuzzy controller using the nonlinear defuzzification algorithm  $d_{NL}$  is a nonlinear function of the inputs, error and rate, the fuzzy controller is a nonlinear fuzzy controller.

(B) If scaled error and/or rate of process output are not within the interval  $[-L, L]$  of the fuzzification algorithm shown in Fig. 2, the incremental output of the fuzzy controller is as listed in Table 2 for the nonlinear defuzzification algorithm  $d_{NL}$ .

(C) The fuzzy controller is precisely equivalent to the nonfuzzy linear PI controller

$$du_{PI}(nT) = - [Ki * e(nT) + Kp * r(nT)] \quad (11)$$

TABLE 2. THE INCREMENTAL OUTPUT OF THE FUZZY CONTROLLER WHEN SCALED ERROR AND/OR RATE OF PROCESS OUTPUT ARE NOT WITHIN THE INTERVAL  $[-L, L]$  OF THE FUZZIFICATION ALGORITHM. THE INPUT COMBINATIONS OF SCALED ERROR AND RATE ARE SHOWN GRAPHICALLY IN FIG. 4

Input Combinations (IC) of scaled error and rate as shown in Fig. 4	Incremental output of the controller, $du_{NL}(nT)$
IC9 and IC10	$- [GR * r(nT) + L] * GU / 2$
IC11 and IC12	$- [GE * e(nT) + L] * GU / 2$
IC13 and IC14	$- [GR * r(nT) - L] * GU / 2$
IC15 and IC16	$- [GE * e(nT) - L] * GU / 2$
IC17	$- L * GU$
IC18	0
IC19	$L * GU$
IC20	0

if the linear defuzzification algorithm  $d_L$  is employed. Therefore, the fuzzy controller becomes the PI controller with the proportional-gain  $K_p = 0.5 * GU * GR / L$  and the integral-gain  $K_i = 0.5 * GU * GE / L$ .

Comparing equation (11) with equations (9) and (10), it is obvious that the fuzzy controller is a nonlinear PI controller with  $K_p$  and  $K_i$  changing with error and rate. We call  $K_p$  and  $K_i$  dynamic proportional-gain  $K_{p_d}$  and dynamic integral-gain  $K_{i_d}$ , respectively, when error and rate are not zero. According to equations (9) and (10), the dynamic proportional-gain  $K_{p_d}$  and integral-gain  $K_{i_d}$  for the fuzzy controller using the nonlinear defuzzification algorithm  $d_{NL}$  are:

$$K_{p_d} = \frac{0.5 * L * GU * GR}{2L - GE * |e(nT)|} \quad (12)$$

$$K_{i_d} = \frac{0.5 * L * GU * GE}{2L - GE * |e(nT)|}$$

when  $GR * |r(nT)| \leq GE * |e(nT)| \leq L$ , and

$$K_{p_d} = \frac{0.5 * L * GU * GR}{2L - GR * |r(nT)|} \quad (13)$$

$$K_{i_d} = \frac{0.5 * L * GU * GE}{2L - GR * |r(nT)|}$$

when  $GE * |e(nT)| \leq GR * |r(nT)| \leq L$ .

Also, we define the static proportional-gain  $K_{p_s}$  and static integral-gain  $K_{i_s}$  as the proportional-gain and the integral gain when both error and rate are zero. According to equations (12) and (13), the static proportional-gain  $K_{p_s}$  and static integral-gain  $K_{i_s}$  for the fuzzy controller using the nonlinear defuzzification algorithm  $d_{NL}$  are

$$K_{p_s} = GU * GR / 4, \quad K_{i_s} = GU * GE / 4. \quad (14)$$

### 2.6. Computer simulation results

Computer simulation was used to determine the control performance of the fuzzy controller. The results of the fuzzy controller using the nonlinear defuzzification algorithm  $d_{NL}$  were compared to that of the PI controller. As a matter of interest, because of its wide use, simulated results were included for the nonlinear fuzzy controller using the Zadeh OR, as well as the theoretically correct Lukasiewicz OR.

To make a fair comparison between the fuzzy and PI controllers, the static proportional-gain  $K_{p_s}$  and the static integral-gain  $K_{i_s}$  of the fuzzy controller were set to equal the proportional-gain  $K_p$  and the integral-gain  $K_i$ , respectively, of the PI controller. That is, for the nonlinear fuzzy

controller,

$$K_p = \frac{GU * GR}{4} \quad \text{and} \quad K_i = \frac{GU * GE}{4}. \quad (15)$$

There are infinitely many combinations of  $GE$ ,  $GR$  and  $GU$  so that expressions (15) hold true. We can introduce desired degrees of nonlinearity of the fuzzy controller by using different combinations of  $GE$  and  $GR$ . Once  $GE$  and  $GR$  are selected,  $GU$  can be uniquely determined to satisfy expression (15).

Linear first-order and second-order process models were first employed and the simulated control results are shown in Fig. 5 and Fig. 6, respectively. It appears that there was little difference in control performance between the PI controller and the fuzzy controller using the nonlinear defuzzification algorithm  $d_{NL}$ .

Then, a first-order with a time delay process model was controlled and the results are given in Fig. 7. The fuzzy controller using the nonlinear defuzzification algorithm  $d_{NL}$  produced considerably less overshoot than the PI controller.

Finally, a nonlinear process model was used. Simulated control results, shown in Fig. 8, indicated that there was significant difference between the control performance of the PI controller and those of the fuzzy controller using the nonlinear defuzzification algorithm  $d_{NL}$ . According to Fig. 8, the PI control result was not stable, but the nonlinear fuzzy controller was stable.

There was little difference between the versions of the nonlinear fuzzy controller using the Zadeh or Lukasiewicz OR, although the Lukasiewicz OR appeared slightly superior. For the above examples, the interval  $[-16, 16]$  of the fuzzification algorithm was used. We shall now explore the reasons why the fuzzy controller using the nonlinear defuzzification was better than the PI controller.

### 2.7. Reasons for improved performance of the fuzzy controller over PI controller.

To compare the PI controller to the fuzzy controller using the nonlinear defuzzification  $d_{NL}$ , the PI controller can be expressed by the following equation, if  $K_p$  and  $K_i$  used by the PI controller satisfy expression (15)

$$du_{PI}(nT) = -\frac{GU}{4} [GE * e(nT) + GR * r(nT)]. \quad (16)$$

By comparing equation (16) with equations (9) and (10), we obtain

$$\frac{1}{4 - \frac{2 * GE * |e(nT)|}{L}} \geq \frac{1}{4} \quad (17)$$

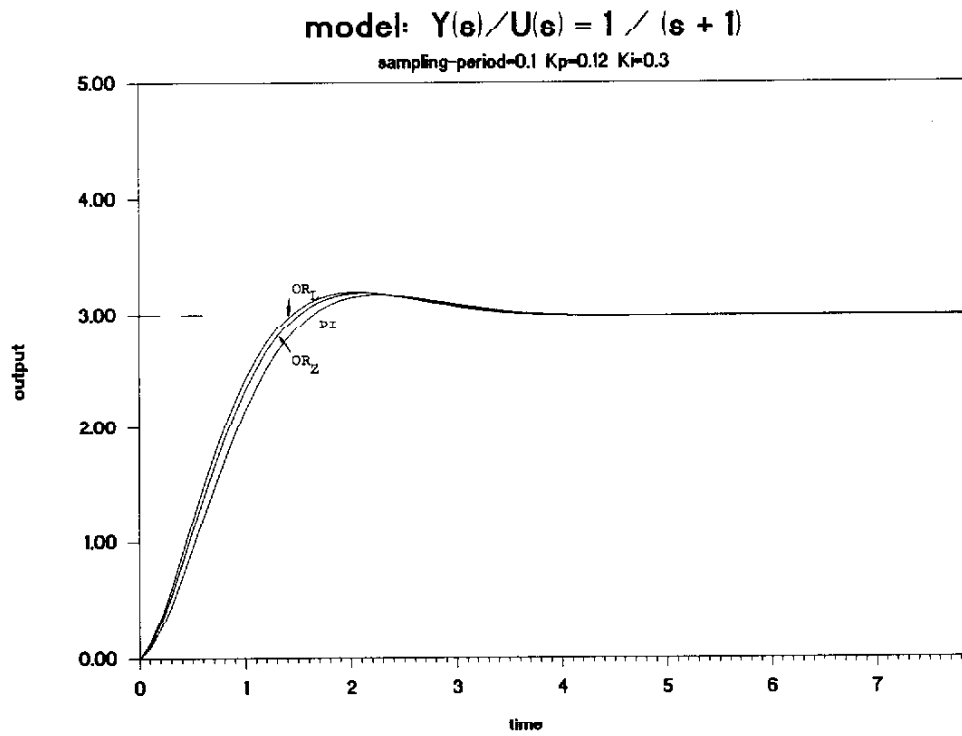


FIG. 5. Comparison of the performances between the fuzzy controller and the PI controller when a linear first-order model  $Y(s)/U(s) = 1/(s + 1)$  is used.  $OR_Z$  denotes the fuzzy controller using the Zadeh OR;  $OR_L$  denotes the fuzzy controller using the Lukasiewicz OR.

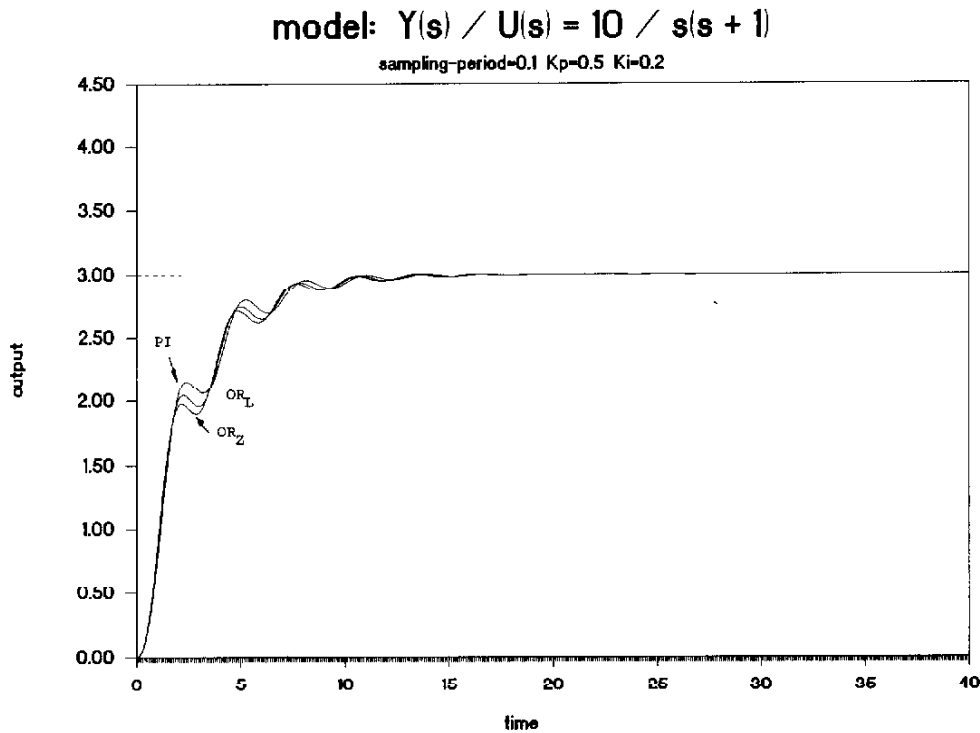


FIG. 6. Comparison of the performances between the fuzzy controller and the PI controller when a linear second-order model  $Y(s)/U(s) = 10/s(s + 1)$  is used.  $OR_Z$  denotes the fuzzy controller using the Zadeh OR;  $OR_L$  denotes the fuzzy controller using the Lukasiewicz OR.

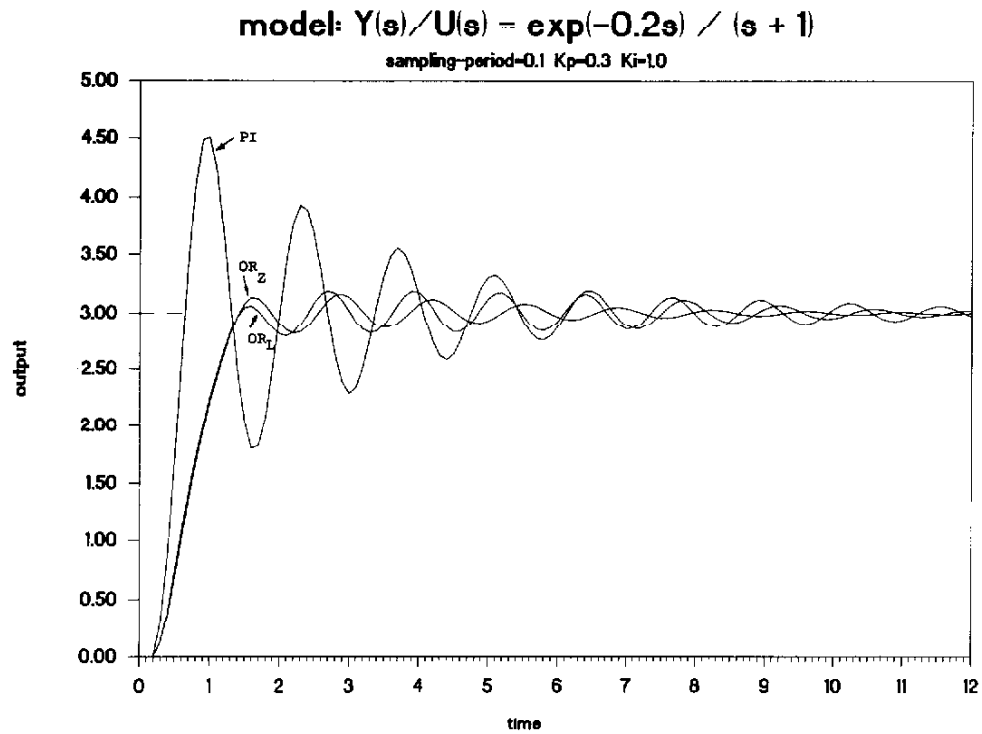


FIG. 7. Comparison of the performances between the fuzzy controller and the PI controller when a first-order with a time delay model

$$Y(s)/U(s) = e^{-0.2s}/(s + 1)$$

is used. OR<sub>Z</sub> denotes the fuzzy controller using the Zadeh OR; OR<sub>L</sub> denotes the fuzzy controller using the Lukasiewicz OR.

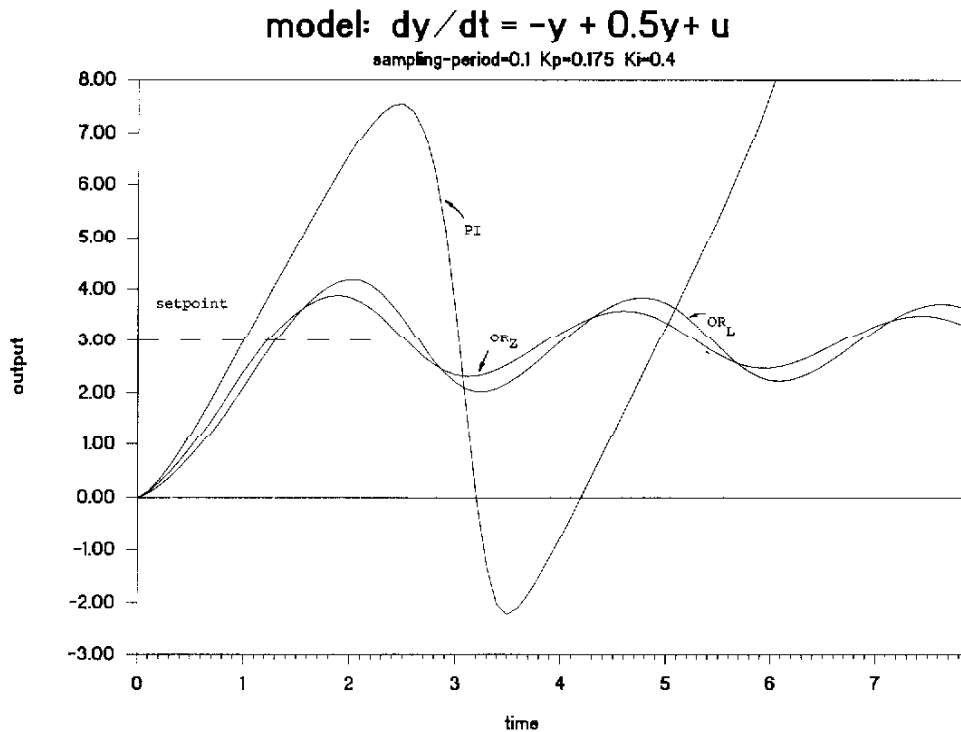


FIG. 8. Comparison of the performances between the fuzzy controller and the PI controller when a nonlinear model

$$dy/dt = -y + 0.5y^2 + u$$

is used. OR<sub>Z</sub> denotes the fuzzy controller using the Zadeh OR; OR<sub>L</sub> denotes the fuzzy controller using the Lukasiewicz OR.

when  $GR * |r(nT)| \leq GE * |e(nT)| \leq L$ , and

$$\frac{1}{4 - \frac{2 * GR * |r(nT)|}{L}} \geq \frac{1}{4} \quad (18)$$

when  $GE * |e(nT)| \leq GR * |r(nT)| \leq L$ .

This means

$$|du_{NL}(nT)| \geq |du_{PI}(nT)| \quad (19)$$

when  $GE * |e(nT)| \leq L$  and  $GR * |r(nT)| \leq L$ .

According to equations (16) to (19) as well as equations (9) and (10), we can conclude:

(1) When  $e(nT) = 0$  and/or  $r(nT) = 0$ ,

$$|du_{NL}(nT)| = |du_{PI}(nT)|.$$

Otherwise,

$$|du_{NL}(nT)| > |du_{PI}(nT)|.$$

(2) The larger  $|e(nT)|$  ( $|r(nT)|$ ) is, the larger the difference between  $du_{NL}(nT)$  and  $du_{PI}(nT)$ . This nonlinearity may be desirable in order to control a process with less rise-time and less overshoot. On the other hand, when the process output is near the setpoint and the rate is near zero, the control action of the fuzzy controller using the nonlinear defuzzification is about the same as that of the PI controller. That ensures zero steady-state error of the fuzzy control system and probably makes the fuzzy control system more stable.

(3) By adjusting the input scalers GE and/or GR, we can change the degree of the nonlinearity of the fuzzy controller employing the nonlinear defuzzification  $d_{NL}$ . The larger the GE (GR) is, the more nonlinear the fuzzy controller will be.

### 3. CONCLUSION

The fuzzy controller discussed above is precisely equivalent to the nonfuzzy linear PI controller if a linear defuzzification algorithm is employed.

Based on the above theoretical analysis and computer simulation, we conclude that the control performances of the fuzzy controller and the nonfuzzy linear PI controller were almost the same if the linear process models were

controlled. However, the fuzzy controllers could control the time-delay process model and nonlinear process model significantly better than the nonfuzzy linear PI controller, due to the nonlinearities of the fuzzy controller introduced by the nonlinear defuzzification algorithm, no matter whether the Zadeh or Lukasiewicz OR was used. This conclusion may be also hold true for other time-delay and nonlinear processes and for other fuzzy controllers with more input, more output and more fuzzy control rules. A fuzzy controller with a large number of linear fuzzy control rules has been studied (Buckley and Ying, 1988b).

The degree of nonlinearities of the fuzzy controller can be changed by adjusting the input scalers GE and GR.

It is known that a linear controller is not necessarily good for controlling a nonlinear process. Generally speaking, a nonlinear controller may control a nonlinear process better. But, designing a nonlinear controller is much more difficult than designing a linear controller, though some nonlinear control theories exist. Therefore, the fuzzy controller constructed in this paper provides an alternative method to control nonlinear processes, even if the mathematical models of the nonlinear processes are available.

Future research will focus on the nonlinearities introduced by the other components of the fuzzy controller for the smallest possible fuzzy controller and other fuzzy controllers with more input, more output and more fuzzy control rules.

### REFERENCES

- Buckley, J. J. and H. Ying (1988a). Linear fuzzy controller: It is a linear non-fuzzy controller. *Inf. Sci.*, to appear.
- Buckley, J. J. and H. Ying (1988b). Fuzzy controller theory: limit theorems for linear fuzzy control rules. *Automatica*, **25**, 469-472.
- Mamdani, E. H. (1974). Application of fuzzy algorithms for control of simple dynamics plant. *Proc. IEE*, **121**, 1585-1588.
- Mamdani, E. H. and B. R. Gaines (Eds) (1981). *Fuzzy Reasoning and its Applications*. Academic Press, London.
- Siler, W. and H. Ying (1988). Fuzzy control theory: the linear case. *Fuzzy Sets Sys.*, **33**, 275-290.
- Sugeno, M. (Ed.) (1985). *Industrial Applications of Fuzzy Control*. Elsevier (North-Holland), Amsterdam.
- Zadeh, L. A. (1965). Fuzzy Sets. *Inf. Control*, **8**, 338.