INFORMATION SCIENCES **51**, 183–192 (1990)

Linear Fuzzy Controller: It Is a Linear Nonfuzzy Controller

J. BUCKLEY*

Mathematics Department, University of Alabama at Birmingham, Birmingham, Alabama 35294

and

H. YING

Kemp-Carraway Heart Institute, Birmingham, Alabama 35234

ABSTRACT

We consider a process controlled, by a controller described by an nth-order linear ordinary differential equation, towards its target output. As a special case, the controller is a proportional-integral-derivative (PID) controller. We show how to construct a linear fuzzy controller that gives precisely the same control as the PID controller. We speculate that nonfuzzy controllers and fuzzy controllers may coincide on an unsuspectingly large class of control problems.

1. INTRODUCTION

This paper continues the initial work in [4], where the authors defined a linear fuzzy controller and evaluated the fuzzy control rules using different kinds of fuzzy logic. They investigated the output of this linear fuzzy controller for the special case of two fuzzy controller inputs and three fuzzy numbers employed to fuzzify each controller input. The present paper generalizes their results to an arbitrary number of fuzzy controller inputs and an arbitrary number of fuzzy numbers employed to fuzzify each controller input, but for only one type of fuzzy logic used to evaluate the fuzzy control rules.

Consider a process with one output $y(t)$, one input $u(t)$, and a set of point s which is the desired process output. For our observations on $y(t)$ the controller computes (approximates) $y^{(i)}(t)$, which is the ith derivative of error $= y(t)-s$, for $0 \leqslant i \leqslant n$. We agree that the 0th derivative of error is equal to error. We

*Address all correspondence to this author.

655 Avenue of the Americas, New York, NY 10010 0020-0255/90/$03.50

choose constants c_i, $0 \leqslant i \leqslant n$, so that $c_i y^{(i)}(t) \in [-1,1]$ for $t \geqslant 0$. Let $r^{(i)}(t) = c_i y^{(i)}(t)$, $0 \leqslant i \leqslant n$. The linear nonfuzzy controller is

$$\frac{du}{dt} = -\sum_{i=0}^{n} r^{(i)}(t), \tag{1}$$

which means that the rate of change of the process input is directly proportional to error, rate of change of error, etc. If we now let $du^*/dt = (n+1)^{-1}\, du/dt$, then

$$\frac{du^*}{dt} = -\frac{1}{n+1} \sum_{i=0}^{n} r^{(i)}(t). \tag{2}$$

The controller computes the $r^{(i)}(t)$ from the process output $y(t)$, producing du^*/dt from Equation (2). Then $du/dt = (n+1)\, du^*/dt$, and we obtain du as du/dt times the time increment of the observations, dt. The new process input is the previous process input plus du. We call this controller the linear nonfuzzy PID$^{(n)}$ controller, where P stands for proportional, I for integral, and D$^{(n)}$ for the nth derivative. Special cases of the PID$^{(n)}$ controller are: (1) the P controller when $n = 0$; (2) the PI controller when $n = 1$; and (3) the PID controller when $n = 2$. In general we call this linear nonfuzzy controller the PID$^{(n)}$ controller for all $n \geqslant 0$. Notice that we have scaled all the variables $r^{(i)}(t)$, $0 \leqslant i \leqslant n$, and du^*/dt so that their values always lie in $[-1,1]$.

In the next section we describe a linear fuzzy controller whose input is the $r^{(i)}(t)$, $0 \leqslant i \leqslant n$, and whose output is exactly equal to the value of du^*/dt in Equation (2). This proves that a fuzzy controller can always be constructed to give precisely the same control as the linear nonfuzzy PID$^{(n)}$ controller. Since the PID$^{(n)}$ controller has enjoyed wide use and success in engineering, this result may explain some of the success obtained with the linear fuzzy controller.

2. LINEAR FUZZY CONTROLLER

We assume the output from the process is $r^{(i)}(t)$, $0 \leqslant i \leqslant n$, which will be input to the fuzzy controller. The fuzzy controller is called linear because: (1) the fuzzification of the $r^{(i)}$ is a linear operation; (2) the fuzzy control rules are linear; and (3) the defuzzification of the fuzzy controller output is also a linear operation. We now discuss each of these parts of the linear fuzzy controller in detail.

2.1. FUZZIFICATION

We first define $2N+1$, $N \geqslant 1$, triangular fuzzy numbers for each of the $n+1$ fuzzy controller inputs $r^{(i)}$. The fuzzy numbers for $r^{(i)}$ will be denoted by $R_j^{(i)}$, $1 \leqslant j \leqslant 2N+1$. Let $v(j) = [j-(N+1)]/N$, which is the central value of $R_j^{(i)}$. The graph of the membership function $y = \mu(x|R_j^{(i)})$ of $R_j^{(i)}$, $2 \leqslant j \leqslant 2N$, is: (1) zero outside $(v(j-1), v(j+1))$; (2) one at $v(j)$; (3) a straight line segment from $(v(j-1),0)$ to $(v(j),1)$ on $[v(j-1), v(j)]$; and (4) a straight line segment from $(v(j),1)$ to $(v(j+1),0)$ on $[v(j), v(j+1)]$. The graph of $y = \mu(x|R_j^{(i)})$ is a triangle over $[v(j-1), v(j+1)]$ for $2 \leqslant j \leqslant 2N$. The graph of $y = \mu(x|R_1^{(i)})$ is simply a straight line segment from $(-1,1)$ to $(-1+N^{-1},0)$ on $[-1,-1+N^{-1}]$, and zero otherwise. The graph of $y = \mu(x|R_{2N+1}^{(i)})$ is a straight line segment from $(1-N^{-1},0)$ to $(1,1)$ on $[1-N^{-1},1]$, and zero otherwise. All fuzzy numbers have their support in $[-1,1]$, and we have the same set of fuzzy numbers for each controller input $r^{(i)}$.

Given a value for $r^{(i)}$ in $[-1,1]$, there is a unique value of j in $\{1,\ldots,2N\}$, say $j(i)$, such that $r^{(i)}$ is in $[v(j(i)), v(j(i)+1))$ if $1 \leqslant j(i) < 2N$, or $r^{(i)}$ is in $[1-N^{-1},1]$ if $j(i) = 2N$. Let $\mu_j^{(i)} = \mu(r^{(i)}|R_j^{(i)})$. We therefore see that, for $0 \leqslant i \leqslant n$, we have

$$\mu_j^{(i)} = \begin{cases} \alpha_{j(i)}^{(i)} & \text{if } j = j(i), \\ 1-\alpha_{j(i)}^{(i)} & \text{if } j = j(i)+1, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

where

$$\alpha_{j(i)}^{(i)} = j(i) - N(r^{(i)}+1). \tag{4}$$

This shows that each $\mu_j^{(i)}$ is a linear function[1] of the input $r^{(i)}$, and we say that we have a linear fuzzification operation.

2.2. RULES

We have fuzzy control rules $\mathcal{R}_k$ for $1 \leqslant k \leqslant K = (n+1)(2N)+1$. Let $\mathcal{S}^{(i)}$ denote some statement about $r^{(i)}$, $0 \leqslant i \leqslant n$. For example $\mathcal{S}^{(0)}$ could be

[1]For $r^{(i)}$ in $[v(j(i)), v(j(i)+1))$ if $1 \leqslant j(i) < 2N$, or for $r^{(i)}$ in $[1-N^{-1},1]$ if $j(i) = 2N$.

"error" and $\mathscr{S}^{(1)}$ could be "rate," which means rate of change of error. Then

$$\mathscr{R}_1 : \text{If } \left[\mathscr{S}^{(0)} = R_1^{(0)}\right] \text{ AND } \cdots \text{ AND } \left[\mathscr{S}^{(n)} = R_1^{(n)}\right],$$

$$\text{then } O = (\text{member } K),$$

and

$$\mathscr{R}_K : \text{If } \left[\mathscr{S}^{(0)} = R_{2N+1}^{(0)}\right] \text{ AND } \cdots \text{ AND } \left[\mathscr{S}^{(n)} = R_{2N+1}^{(n)}\right],$$

$$\text{then } O = (\text{member } 1),$$

where O denotes output. The output will be a discrete fuzzy set whose elements are numbered $l = 1,2,\ldots,K$. Traditionally, one uses linguistic variables like "negative—large," "positive—very small," etc. for the elements in O. However, since O may be very large, we have decided to use numbers for the elements in O instead of linguistic variables. Our defuzzification algorithm employs the central value of each element in O, so it will be convenient in the rest of this paper to assign to each element in O its central value. Let

$$v(l) = \frac{l-[(n+1)N+1]}{(n+1)N}. \tag{5}$$

The value of $v(l)$ produces the central value of the element in O numbered l. For example: (1) $O = -1 = v(1)$ is the first element in O numbered $l = 1$; (2) $O = 0.00 = v(l)$ is an element in O numbered $l = (n+1)N+1$; and (3) $O = 1 = v(K)$ is the last element in O numbered $l = K$. For any fuzzy control rule $\mathscr{R}_k$ whose conclusion is "$O = (\text{member } l)$" we will now write the conclusion as "$O = v(l)$." That is, we will use the numerical value (central value) of the element in O in place of its name (number).

If $\mathscr{C}_j$, $1 \leqslant j \leqslant J$, denotes a clause like $[\mathscr{S}^{(i)} = R_j^{(i)}]$, then

$$\text{OR}\left\{\mathscr{C}_j \middle| 1 \leqslant j \leqslant J\right\}, \tag{6}$$

stands for

$$\mathscr{C}_1 \text{ OR } \mathscr{C}_2 \text{ OR } \cdots \text{ OR } \mathscr{C}_J. \tag{7}$$

We may now define all the other rules as

$$\mathscr{R}_k : \text{If } \text{OR}\left\{ \left[\mathscr{S}^{(0)} = R_{i_0}^{(0)} \right] \text{AND} \cdots \text{AND} \left[\mathscr{S}^{(n)} = R_{i_n}^{(n)} \right] \middle| i_0 + \cdots + i_n = n + k \right\},$$

$$\text{then } O = \frac{(n+1)N + 1 - k}{(n+1)N},$$

for $1 < k < K$. We say the rules numbered k, $1 < k < K$, are linear because the OR is taken over all subscripts with constant sum $n + k$, where k is the rule number. All rules have prior rule confidence one, and there is no tresholding. Therefore, all rules fire given input $r^{(i)}$.

To evaluate a rule we must first evaluate all the clauses $[\mathscr{S}^{(i)} = R_j^{(i)}]$. The value of $[\mathscr{S}^{(i)} = R_j^{(i)}]$ is $\mu_j^{(i)} = \mu(r^{(i)} | R_j^{(i)})$. Let T be any t-norm extended (by associativity) to $n + 1$ arguments, and let C be any co-t-norm extended to $n + 1$ arguments. For example

$$\text{LOR}(x_0, \ldots, x_n) = \min\left(\sum_{i=0}^{n} x_i, 1 \right), \tag{8}$$

is the extended Lukasiewicz OR, and

$$\text{PAND}(x_0, \ldots, x_n) = \prod_{i=0}^{n} x_i \tag{9}$$

is the extended probabilistic AND. We have a special case when $n = 0$, and then we set $T(x_0) = x_0$ and $C(x_0) = x_0$ for the t-norm and co-t-norm, respectively, of only one argument

When $\mathscr{R}_k$ fires, the value of its left-hand side is Δ_k, which is given by

$$\Delta_1 = T\left(\mu_1^{(0)}, \ldots, \mu_1^{(n)} \right), \tag{10}$$

$$\Delta_K = T\left(\mu_{2N+1}^{(0)}, \ldots, \mu_{2N+1}^{(n)} \right), \tag{11}$$

and

$$\Delta_k = C\left\{ T\left(\mu_{i_0}^{(0)}, \ldots, \mu_{i_n}^{(n)} \right) \middle| i_0 + \cdots + i_n = n + k \right\} \tag{12}$$

for $1 < k < K$. We have used the notation $C\{ z_j | 1 \leqslant j \leqslant J \}$ for $C(z_1, \ldots, z_J)$.

The numbers Δ_k produce the discrete fuzzy set O whose elements are $v(l)$, $1 \leqslant l \leqslant K$. The membership value of $v(l)$ is Δ_{K-l+1}, $1 \leqslant l \leqslant K$.

2.3. DEFUZZIFY

The discrete fuzzy set O is defuzzified into the real number DERIV as

$$\mathrm{DERIV} = \sum_{l=1}^{K} \Delta_l v(K-l+1) = \sum_{l=1}^{K} \Delta_l \left[\frac{(n+1)N+1-l}{(n+1)N} \right]. \tag{13}$$

This is a linear operation, like expected value in probability, where we take an element in O times its membership value and sum over all the members of O.

2.4. MAIN RESULT

We now show that with the correct T and C the fuzzy controller output DERIV is equal to du^*/dt in Equation (2).

THEOREM. *If $T =$ PAND and $C =$ LOR, then*

$$\mathrm{DERIV} = -\frac{1}{n+1} \sum_{i=0}^{n} r^{(i)}.$$

Proof. Given $r^{(i)} \in [-1,1]$, $0 \leqslant i \leqslant n$, they determine the $j(i)$ in $\{1,\ldots,2N\}$. Let $m = [\sum_{i=0}^{n} j(i)] - n$. We notice that all the $\Delta_l = 0$ except for $l = m, m+1, \ldots, m+n+1$ because only two $\mu_j^{(i)}$ can be nonzero for each i. Therefore

$$\Delta_m = \prod_{i=0}^{n} \alpha_{j(i)}^{(i)} \tag{14}$$

and

$$\Delta_{m+n+1} = \prod_{i=0}^{n} \left(1 - \alpha_{j(i)}^{(i)}\right). \tag{15}$$

We must now consider $n = 0$ as a special case. If $n = 0$, then $m = j(0)$ and we only have Δ_m and Δ_{m+1} given by Equations (14) and (15), respectively, for $n = 0$. Putting these values into Equation (13), one easily sees that DERIV equals $-r^{(0)}$.

We will now assume that $n \geqslant 1$. For $1 \leqslant j \leqslant n$ let

$$\Omega_j = \sum_{\Gamma_j} \prod \left\{ \alpha_{j(i)}^{(i)} \middle| i \in I - \Gamma_j \right\} \prod \left\{ 1 - \alpha_{j(i)}^{(i)} \middle| i \in \Gamma_j \right\}, \tag{16}$$

where the sum is over all $\Gamma_j \subset \{0,1,\ldots,n\}$ with $|\Gamma_j| = j$. Then

$$\Delta_{m+j} = \mathrm{LOR}(\Omega_j, 1) \tag{17}$$

for $1 \leqslant j \leqslant n$. By Lemma 1 (Appendix) we have

$$\Delta_{m+j} = \Omega_j, \tag{18}$$

since $\Omega_j \leqslant 1$. Hence

$$\mathrm{DERIV} = \left[\frac{(n+1)N+1-m}{(n+1)N}\right] \sum_{j=0}^{n+1} \Delta_{m+j} - \frac{1}{(n+1)N} \sum_{j=0}^{n+1} j\Delta_{m+j}. \tag{19}$$

Lemma 2 (Appendix) shows that

$$\sum_{j=0}^{n+1} \Delta_{m+j} = 1. \tag{20}$$

Also, by Lemma 3 (Appendix) we have

$$\sum_{j=0}^{n+1} j\Delta_{m+j} = (n+1) - \sum_{j=0}^{n} \alpha^{(i)}_{j(i)}. \tag{21}$$

We now substitute Equations (20) and (21) into Equation (19), and also Equation (4) for $\alpha^{(i)}_{j(i)}$, and recalling that the sum of the $j(i)$ equals $m+n$, we obtain the desired result.

3. DISCUSSION

We first notice that the results presented above are true for any $N \geqslant 1$, so we may use only three $(N-1)$ fuzzy members for each input $r^{(i)}$. This reduces the number of rules to only $K = 2n+3$. However, no rules are needed, because we derived a closed-form expression for the defuzzified output DERIV in terms of the fuzzy controller inputs $r^{(i)}$. That is, we can compute DERIV without firing any rules.

The linear fuzzy controller described above may be called a probabilistic fuzzy controller because the rules are evaluated as if we were dealing with probabilities (see the proof of Lemma 1 in the Appendix).

One may wonder what will happen if we use other t-norms and co-t-norms to evaluate the rules. In [4] the authors show, for the case where $N = n = 1$, that for certain T and C DERIV equals du^*/dt [Equation (2)], while for other choices of T and C the two are not always equal. In particular, if $T = \min$ and $C = \max$, then the two controllers are not necessarily equal for all inputs $r^{(i)}$. We conclude that the success of the fuzzy controller may be very sensitive to the choice of the t-norm and co-t-norm used to evaluate the fuzzy control rules.

Some authors [1–3] use other types of fuzzy numbers (nonlinear, trapezoidal) for each input instead of our triangular fuzzy numbers. We have worked out a few simple examples which show that DERIV will not always equal du^*/dt [Equation (2)] when trapezoidal fuzzy numbers are employed. Therefore, the use of triangular fuzzy numbers was essential for the main result in this paper.

The important parts of the linear fuzzy controller needed to obtain that DERIV equals du^*/dt are: (1) triangular fuzzy numbers for fuzzification; (2) linear fuzzy control rules, a separate conclusion for each rule, and $T =$ PAND, $C =$ LOR; and (3) a linear defuzzification algorithm. Future research is needed to see if we can construct fuzzy controllers equal to other well-known nonfuzzy controllers.

APPENDIX

We prove the three lemmas used in the proof of the theorem above.

LEMMA 1. *$\Omega_j \leqslant 1$ for $1 \leqslant j \leqslant n$.*

Proof. Given $r^{(i)}$, define a probability P_i on the fuzzy number set $\mathscr{A}^{(i)} = \{R_1^{(i)}, \dots, R_{2N+1}^{(i)}\}$ as follows:

$$P_i\left[R_j^{(i)}\right] = \begin{cases} \alpha_{j(i)}^{(i)} & \text{if } j = j(i), \\ 1 - \alpha_{j(i)}^{(i)} & \text{if } j = j(i) + 1, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Let $P = \prod_{i=0}^{n} P_i$ be the product probability on $\mathscr{A} = \prod_{i=0}^{n} \mathscr{A}^{(i)}$. The sum of $P(a)$ over all $a \in \mathscr{A}$ is equal to one. Therefore $\Omega_j \leqslant 1$, since Ω_j is a sum of $P(a)$ over some of the $a \in \mathscr{A}$.

LEMMA 2. $\sum_{j=0}^{n+1} \Delta_{m+j} = 1$.

Proof. From Lemma 1 this sum is just the sum of $P(a)$ over all $a \in \mathscr{A}$ where $P(a)$ can be positive; all the other $P(a)$ are necessarily zero. Therefore this sum equals one.

LEMMA 3.

$$\sum_{j=0}^{n+1} j\Delta_{m+j} = (n+1) - \sum_{i=0}^{n} \alpha_{j(i)}^{(i)}.$$

Proof. The proof is by induction on $n \geqslant 0$. We first show the equality holds for $n = 0$. This is obvious, since the sum on the left is equal to $\Delta_{m+1} = \Omega_1 = 1 - \alpha_{j(0)}^{(0)}$, which is the expression on the right when $n = 0$.

Now assume the equality holds for some $n \geqslant 0$, and we show it also holds for $n+1$. To simplify the notation we write $x_i = \alpha_{j(i)}^{(i)}$, and then

$$\Omega_j = \sum_{\Gamma_j} \prod \{x_i | i \in I - \Gamma_j\} \prod \{1 - x_i | i \in \Gamma_j\}, \tag{2}$$

where $\Gamma_j \subset I = \{0, \ldots, n\}$ with $|\Gamma_j| = j$. We see that

$$\sum_{j=0}^{n+2} j\Delta_{m+j} = (1 - x_{n+1}) \sum_{j=1}^{n+2} j\Omega_{j-1} + x_{n+1} \sum_{j=0}^{n+1} j\Omega_j. \tag{3}$$

By the induction hypothesis we have

$$x_{n+1} \sum_{j=0}^{n+1} j\Omega_j = x_{n+1}\left[(n+1) - \sum_{i=0}^{n} x_i\right]. \tag{4}$$

We now observe that

$$\sum_{j=1}^{n+2} j\Omega_{j-1} = \sum_{j=1}^{n+2} (j-1)\Omega_{j-1} + \sum_{j=1}^{n+2} \Omega_{j-1} = \left[(n+1) - \sum_{i=0}^{n} x_i\right] + 1, \tag{5}$$

by the induction hypothesis and Lemma 2, respectively.

If we substitute Equations (4) and (5) into Equation (3) we obtain

$$\sum_{j=0}^{n+2} j\Delta_{m+j} = (n+2) - \sum_{i=0}^{n+1} x_i, \tag{6}$$

and the equation is true for $n+1$. Therefore, the result is true for $n = 0, 1, 2, \ldots$.

REFERENCES

1. J. J. Buckley and H. Ying, Expert fuzzy controller I: Theory, *Fuzzy Sets and Systems*, to appear.
2. L. P. Holmblad and J. J. Ostergaard, Control of a cement kiln by fuzzy logic, in *Fuzzy Information and Decision Processes* (M. M. Gupta and E. Sanchez, Eds.), North-Holland, Amsterdam, 1982.
3. E. H. Mamdani, J. J. Ostergaard, and E. Lembessis, Use of fuzzy logic for implementing rule-based control of industrial processes, in *Fuzzy Sets and Decision Analysis* (H.-J. Zimmerman, L. A. Zadeh, and B. R. Gaines, Eds.), North-Holland, Amsterdam, 1984.
4. W. Siler and H. Ying, Fuzzy control theory: The linear case, *Fuzzy Sets and Systems*, to appear.

Received 29 March 1988; revised 24 June 1988