

# Fuzzy Control of pH Using NAL

Mosé Galluzzo, Vincenzo Cappellani, and  
Umberto Garofalo

*Università di Palermo, Italy*

---

## ABSTRACT

*A fuzzy controller for a neutralization process is described. The controller was set up for a laboratory pilot plant. The approach is shown to be effective and can be extended to highly nonlinear and nonstationary processes. The "operator" knowledge encoded in the rules was obtained by several experimental runs of the system using manual control. Rules are composed using the max-min compositional rule of inference. The use of metarules, which depends on controller performance and on active disturbances, makes the controller behave like an adaptive controller. The control program is encoded in NAL, a new experimental logic programming language that was first used in this work in a real-time application.*

**KEYWORDS:** *fuzzy logic, adaptive control, pH control, process control, expert systems, logic programming*

---

## INTRODUCTION

The pH of a solution is defined, for diluted solutions, as the negative decimal logarithm of the hydrogen ion concentration. The control of pH, which usually consists in maintaining it at  $\text{pH} = 7$ , is required in wastewater treatment plants, in food processing plants, and in many plants where biological processes are carried out. The control of pH is considered (Waller and Gustafsson [1]) one of the most difficult problems in process control. This is due to the nonlinearity and the nonstationarity of neutralization processes.

The neutralization curve, which is the relationship between the neutralizer flowrate and the corresponding pH value of the neutralized stream, is used to characterize a neutralization process. The curve shown in Figure 1a refers to the case of the neutralization of a strong acid solution by the use of a strong base. This nonlinear characteristic may vary widely as a consequence of small

---

*Address correspondence to Mosé Galluzzo, Dipartimento di Ingegneria Chimica dei Processi e dei Materiali, Università di Palermo, Viale delle Scienze, 90128 Palermo, Italy.*

Received March 1, 1989; accepted January 1, 1991.

International Journal of Approximate Reasoning 1991; 5:505-519

© 1991 Elsevier Science Publishing Co., Inc.

655 Avenue of the Americas, New York, NY 10010 0888-613X/91/\$3.50

amounts of secondary species that determine the buffer effect, which is due to the incomplete dissociation of weak electrolytes. The neutralization curve may therefore assume a form similar to one of the shapes shown in Figure 1b depending on the electrolytes present in the stream and on their concentrations. This is why traditional feedback control very often can give unsatisfactory performance: in fact, controller parameters must be tuned assuming that the neutralization process is represented by one point of the neutralization curve. Even feedforward control, which might seem to be a solution, does not usually give good results because knowledge of all species in the input stream and their concentrations would be required.

To overcome these difficulties, both modifications in process design and advanced control strategies have been proposed. Process modifications are often good solutions, but they are not always feasible. Adaptive control based on approximate linear or nonlinear models is probably the most studied advanced control technique in pH control (Pajunen [2], Jacobs et al. [3]). Nevertheless, the application of adaptive control requires the availability of a good model, in this case the knowledge of the main species and their concentrations.

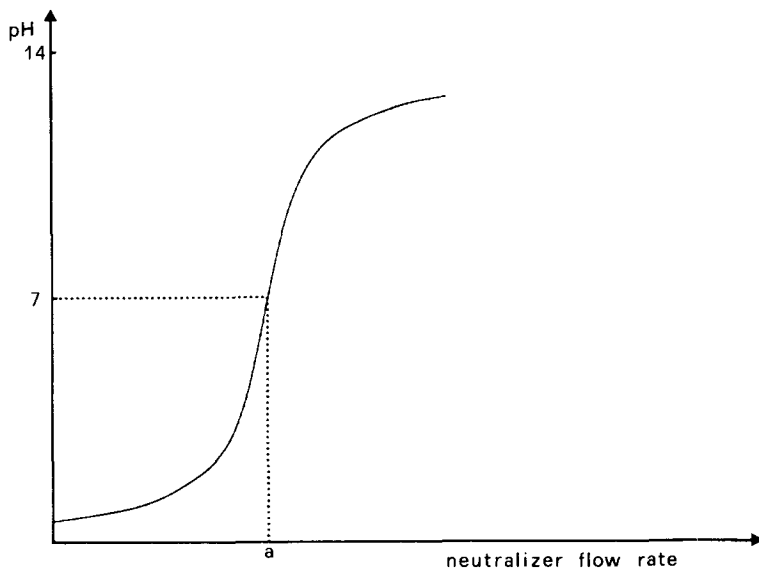
In order to avoid the modeling problem, the use of reaction invariants (Waller and Makila [4], Gustafsson and Waller [5]) or the use of hypothetical species (Jutila and Orava [6], Jutila [7]) has been proposed. Both methods are complex and still require a knowledge of the neutralization process that is not always available. For instance, for the method based on hypothetical species it is necessary to know the concentration of at least one species for each equilibrium reaction in which a hypothetical species is involved.

Fuzzy control has been indicated (Sugeno [8], Galluzzo and Giarratano [9]) as a possible solution for cases in which it is difficult or impossible to derive a reliable mathematical model.

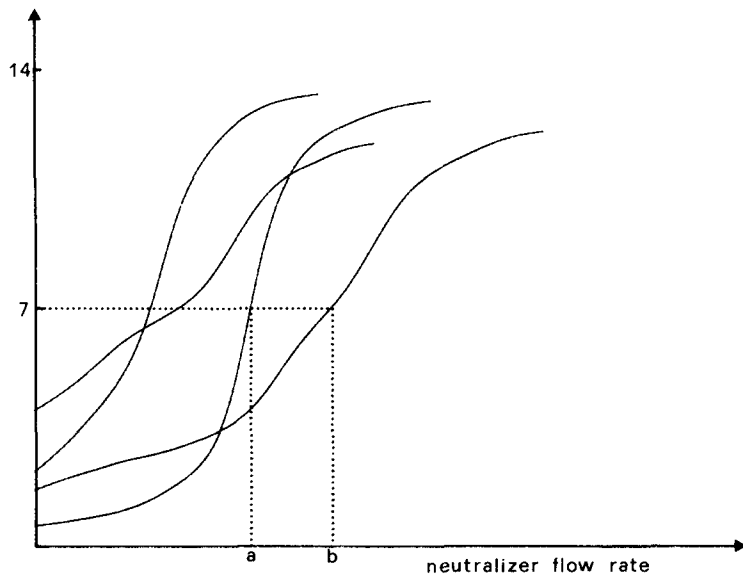
The application of fuzzy control to a simple neutralization process with a single input stream was studied earlier by Galluzzo and Giarratano [10]. A feedforward control scheme was implemented, which makes use of pH, conductivity, and osmotic pressure measurements of the input stream as disturbance measurements. The process model was expressed by heuristic rules, obtained by runs of the system under manual control; rules were composed using the max-min compositional rule of inference.

A more complex neutralization process and a different fuzzy control algorithm are used in the present work. The block diagram chart of the neutralization process is shown in Figure 2. Both the pH and the flow rate of input streams can be considered as disturbances. Various control schemes have been implemented; for the sake of simplicity a mixed feedback-feedforward control scheme using the pH of one input stream as disturbance measurement will be described.

The main characteristics of the control program are its adaptability and the



(a)



(b)

**Figure 1.** Neutralization curves (a) for an unbuffered solution and (b) for a buffered solution.

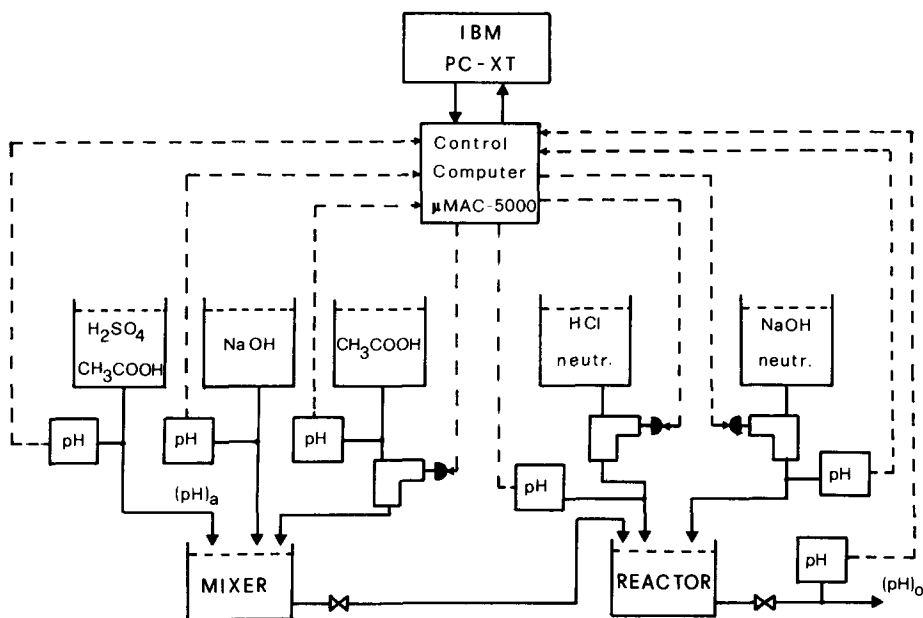


Figure 2. Experimental system.

language used. The fuzzy control algorithm, after an initial setting, can modify its internal structure in an "expert way" to adapt itself to the changing conditions of the system. The control program is written in NAL, a new experimental programming language that is very suitable for control applications.

## EXPERIMENTAL WORK

The solution to be neutralized is obtained in a tank by mixing three feed streams, consisting respectively of a sodium hydroxide solution, an acetic acid solution, and a mixed solution of sulfuric acid and acetic acid. A buffer solution is formed if the concentration of hydrogen ions produced by the dissociation of sulfuric acid is less than the concentration of acetate ions; in this case the acid-base couple acetic acid/acetate ion, which is responsible for the buffer effect, is present in the stream to be neutralized. The higher the acid and base concentrations are, and the closer their rate is to 1, the stronger the buffer effect is. By changing the concentration of one stream, acetic acid and acetate ion concentrations will also change, producing a variable buffer effect.

The buffered solution is then sent to a neutralization reactor where a strong sodium hydroxide solution is used as a neutralizer. Any change of the pH or the flow rate of input streams modifies the neutralization curve.

A data acquisition and control system is used as interface between the process and the IBM PC XT, where the main part of the control program is active.

### The Control Program

The control program CONTESP, used for the mixed feedback–feedforward control configuration as described, is formed of two different parts. One, written in  $\mu$ MACBASIC, runs on the  $\mu$ MAC 5000; the second, written in NAL (see next section), runs on the IBM PC XT.

The control program uses the pH of the solution of sulfuric acid and acetic acid,  $(\text{pH})_a$ , the pH of the output stream,  $(\text{pH})_o$ , and its rate of change as measured variables. This corresponds to the use of a simple gain feedforward control and a proportional derivative feedback control. The flow rate of a strong solution of sodium hydroxide is used as control variable.

The full range of variation of variables was divided into several intervals, and fuzzy subsets were defined for them using linear membership functions for the terminal subsets and triangular membership functions given by

$$f(1) = 2(x - a), \quad a < x < a + (a + b)/2$$

$$f(2) = 2(b - x), \quad a + (a + b)/2 < x < b$$

for intermediate subsets.

Fuzzy subsets and intervals of definition are reported in Table 1.

Control rules were built, making use of the results of several experimental runs of the neutralization system. Each run consisted in the determination, by “manual control”, of the flow rate of the neutralization stream for a different set of values of input variables. This allowed the setup of the “operator” knowledge of the system needed to implement the control action. Dead times of the system were taken into account in writing the control rules.

An example of control rules is the following:

IF  $(\text{pH})_a$  is very low  
 and  $(\text{pH})_o$  is normal  
 and the rate of change of  $(\text{pH})_o$  is negative small  
 THEN after 25 seconds make the neutralizer flow rate high

The knowledge gained by several experimental runs using the fuzzy control algorithm was coded as well in metarules. Metarules are used to automatically

**Table 1.** Fuzzy Subsets and Interval Limits

Variable	Subset	Interval Limits	
		<i>a</i>	<i>b</i>
$(\text{pH})_a$	Very low	1	2
	Low	1.75	2.75
	High	2.5	3.5
	Very high	3.25	4.25
$(\text{pH})_o$	Very low	4	5
	Low	4.8	6.2
	Slightly low	6	6.8
	Normal	6.7	7.3
	Slightly high	7.2	8
	High	7.8	9.2
	Very high	9	10
$d(\text{pH})_o / dt$	Negative big	$< -0.08$	
	Negative small	$> -0.08$	$< 0$
	Positive small	$> 0$	$< 0.08$
	Positive big	$> 0.08$	
Neutralizer flow rate	Very low	0	10
	Low	8	22
	Slightly low	18	32
	Medium low	28	42
	Medium high	38	52
	Slightly high	48	62
	High	58	72
	Very high	68	80

modify rules if the performance of the control system is not satisfactory. The modification of a rule is carried out in the action part by changing the fuzzy subset associated to it.

The conditions that may lead to a modification of control rules are the following.

1. The actual deviation of pH is greater than a fixed maximum deviation.
2. The pH deviation measured at a given time interval before the current time is greater than a fixed maximum deviation.
3. The control action value is outside preset limits that depend on the proximity of the system to the neutrality condition.

An example of a metarule is

IF            the actual  $(\text{pH})_o$  deviation is greater than the allowed deviation  
and    15 seconds ago the  $(\text{pH})_o$  deviation was greater than the allowed deviation

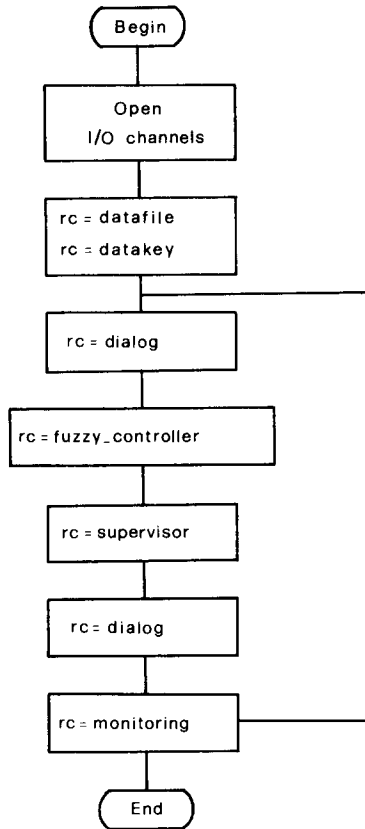


Figure 3. Program flow chart.

and  $(pH)_o$  is very low  
 and the neutralizer flow rate is less than 80% of full scale  
**THEN** use the fuzzy subset of higher level for the control  
 variable in the action part of fired rules.

The fuzzy controller and the metarule modification algorithm are included in a control program shell; its flow diagram is shown in Figure 3.

The phase of rules elicitation was long and difficult, because all the knowledge of the dynamic behavior of the neutralization system had to be acquired by suitable experimental runs and any modification required a new compilation of the program, a time-consuming task even using a suitable language like NAL. In a real situation this phase would be limited to the strict elicitation part because the operator (the expert) would already have the

knowledge. Nevertheless, the burden of writing rules and associating fuzzy subsets to variables would be relieved by the use of a suitable fuzzy programming environment (Chiu and Togai [11]).

Initially the number of subsets used was more limited, but the resulting control performance was not acceptable because of the oscillating behavior. In particular, severe difficulties were found during the startup of the experimental system or with large input changes. An increased number of subsets and some changes to rules and subset definitions resulted in better performance, but it was the introduction of metarules that gave the control system, together with adaptive behavior, a smoother dynamics.

### **NAL Programming Language**

The NAL language was used in this application in order to check its applicability to real-time problems and develop its real-time instructions. Obviously, other languages could have been used in this application; note that in the previous work (Galluzzo and Giarratano [9]) BASIC was used. NAL (Non Algorithm Language) is a new experimental programming language based on a "data flow architecture." At present, NAL is a precompiler written in C that generates a C source code. Because of its logical structure, NAL is well suited for implementing rule-based algorithms. Another advantage in using NAL in real-time applications is its debugging speed in spite of the fact that it is a precompiler.

The main characteristic of NAL is that an instruction is processed as soon as the necessary data are available. This allows the available data to be controlled separately from the logic of the problem. As a result, programming is simplified because it is possible to build separate software components, the execution of which does not need to be planned but is automatically determined by the available data. For this reason NAL has two different levels for the analysis of a program.

In the first level, the availability of data and the executability of instructions are checked. The significant features of this level are

1. Physical control on available data eliminates the errors caused by not initializing data.
2. There is no end-of-file check, and instructions for which data are not available are inhibited.
3. NAL automatically decides when to stop the execution of a program and recognizes whether a program is sequential or cyclic.
4. Instructions are not executed in a procedural way; if an instruction cannot be executed because of lack of data, NAL will execute it when data become available.

In the second level NAL analyzes the logical structure of the program. NAL has been given a syntax similar to logic programming languages, basically



founded on a single structure: the implication between functional propositions

$$y = f(x) \quad ? \quad z = g(w)$$

where  $x$  and  $w$  may also be functions. The meaning of the implication is the following: First, if  $x$  is available, execute  $f(x)$  and assign  $y$ ; then, if  $w$  is available, execute  $g(w)$  and assign  $z$ .

Furthermore, NAL makes use of three operators similar to the control structures of traditional programming languages. The *is* operator checks the truth value of a logic condition; the *all* and *any* operators are equivalent to the quantifier operators of predicate logic.

---

## RESULTS

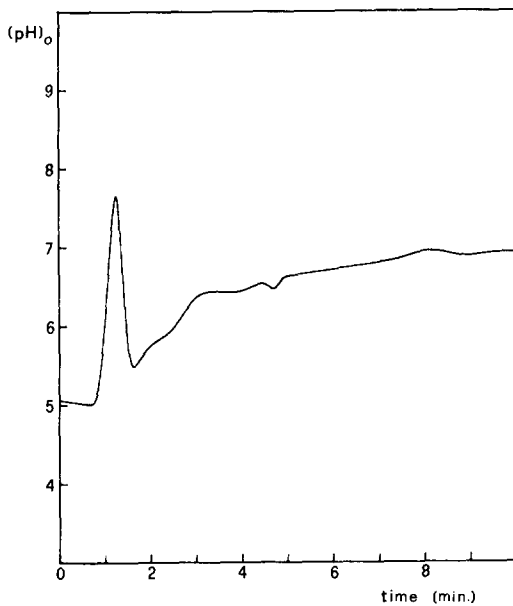
---

The results of some experimental runs using the described control program are reported in Figures 4–6.

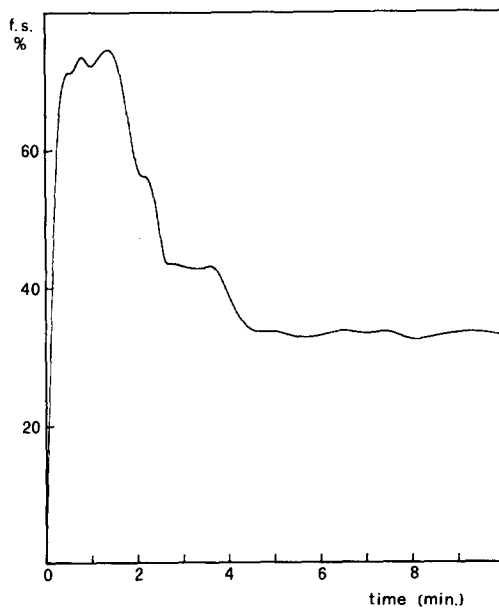
Figure 4 refers to a set point variation from pH = 5 to pH = 7; in this case only the feedback part of the fuzzy control algorithm is active, because no disturbance is introduced. Figure 4a shows the controlled variable (pH)<sub>o</sub>, while Figure 4b shows the control variable, the neutralizer flow rate, in percent of the available control range. Let us note that there is an initial overcompensation, due to the use of the rate of change of the controlled variable, that is nevertheless useful in reducing the response time.

Figure 5 refers to the response of the system to a step disturbance in the input flow rate. Even in this case only the feedback part of the fuzzy control algorithm is active because the input flow rate is not a measured variable. The responses of the uncontrolled and controlled systems are reported in Figures 5a and 5b, while the change in the control variable for the controlled system is reported in Figure 5c. The fuzzy control system is able to bring the pH of the output stream back to 7 within a few minutes; a faster response can be obtained by a different choice of the fuzzy subsets for the control variable, but with the drawback of an increased oscillatory character.

In Figure 6 the response of the system to a step disturbance in the pH of the acid feed is shown. In this case both feedforward and feedback actions are active; Figure 6a refers to a pH variation caused by a flow rate variation of acetic acid, while in Figure 6b the same pH variation was caused by a sulfuric acid flow rate variation; note that in both cases the effect of the disturbance on the uncontrolled system was a deviation of pH from 7 to 3.8. For comparison, a traditional feedforward–feedback control configuration, using a fixed-gain feedforward controller with a time delay and a PID feedback controller, was implemented for the same experimental system.

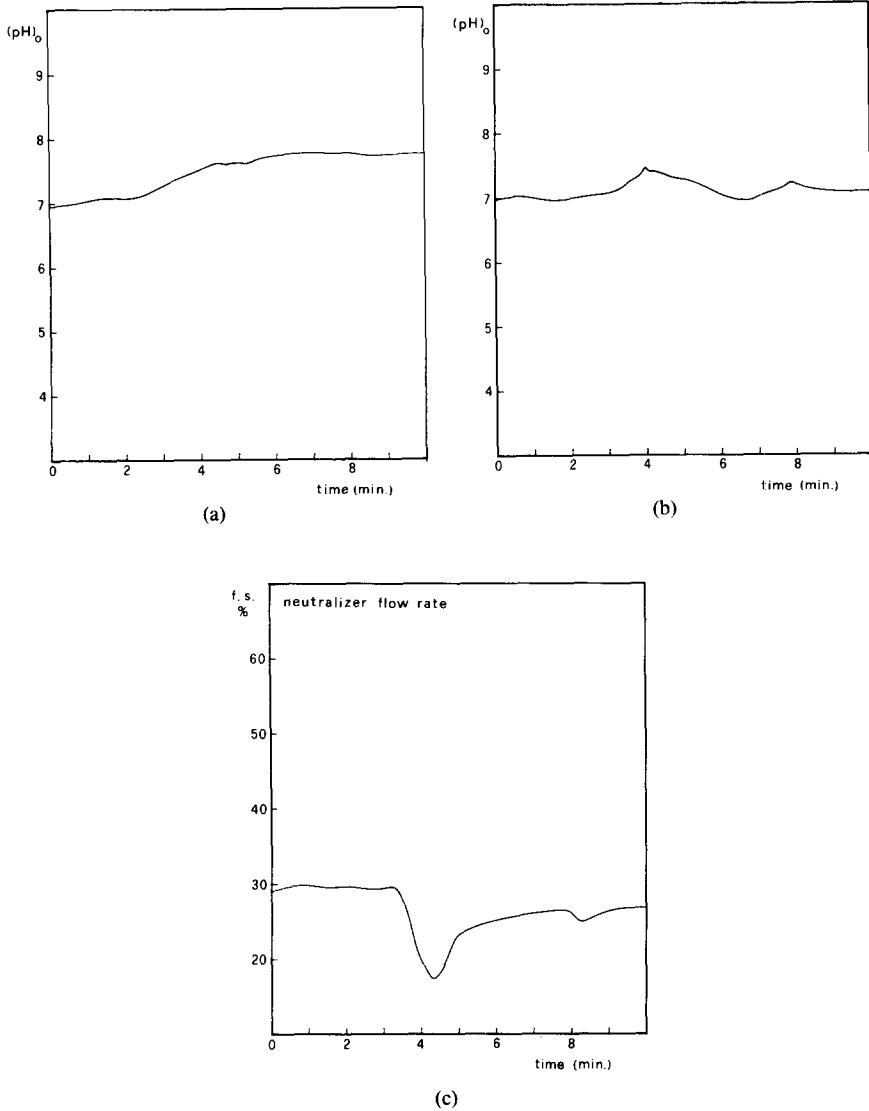


(a)



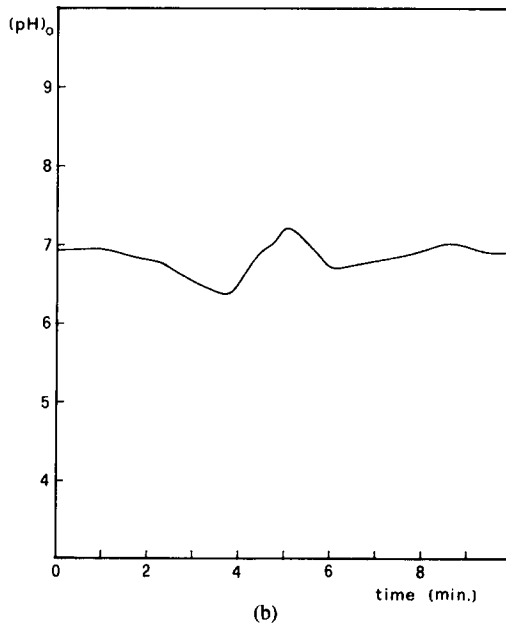
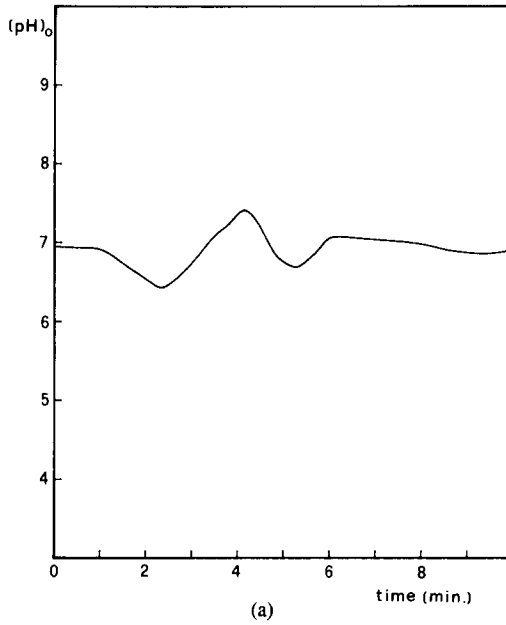
(b)

Figure 4. Expert control—set point response. (a)  $(\text{pH})_o$ , (b) neutralizer flow rate.

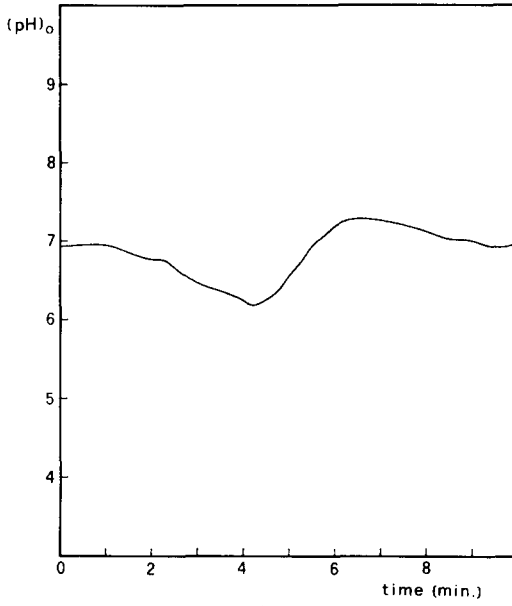


**Figure 5.** Responses to a flow rate disturbance. (a) Uncontrolled system, (b) Expert control,  $(\text{pH})_0$ ; (c) Expert control, neutralizer flow rate.

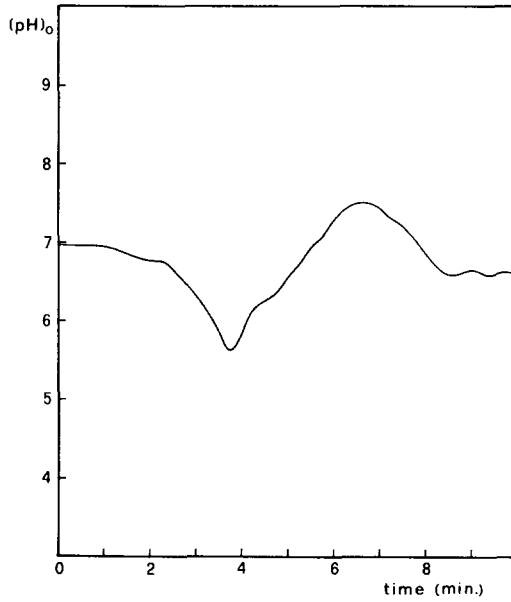
Controller parameters were chosen using a tuning procedure for the feedback controller and some trial-and-error experiments for the gain and the time delay of the feedforward controller. The results of an experimental run using the adopted parameters are shown in Figure 7. They refer to a step disturbance in the pH of the acetic acid-sulfuric acid solution, of the same size as the



**Figure 6.** Expert control—responses to pH disturbances in the input stream caused by variation in (a) acetic acid flow rate, (b) sulfuric acid flow rate.



(a)



(b)

**Figure 7.** Feedforward and PID control—responses to pH disturbances in the input stream caused by variation in (a) acetic acid flow rate, (b) sulfuric acid flow rate.

disturbance used in the experiments of Figure 6. Figure 7a refers to a variation of the input pH obtained by changing the acetic acid flow rate to the mixer, while Figure 7b refers to a pH variation introduced by changing the sulfuric acid flow rate to the mixer. Note that the PID controller tuning was based on step disturbances in the input pH obtained by changing the flow rate of sulfuric acid.

A comparison of Figures 6 and 7 shows that an improvement is obtained by using the fuzzy controller; the improvement consists in a behavior that does not depend on the specific cause of the input pH variation and that is satisfactory in both cases. This implies that only one measurement, the pH of the input stream, is necessary for the implementation of the feedforward fuzzy control system.

---

## CONCLUSIONS

---

The implemented fuzzy controller seems to be a promising answer to the problem of control of nonlinear and nonstationary systems. The robustness of the system response to the input pH variations determined by different causes is a very interesting result that cannot be achieved by the use of traditional controllers. The limitations of the method are related to the phase of rules elicitation and the computer memory and speed requirements.

The determination of controller rules is a complex process requiring a long and difficult trial-and-error adjustment procedure that can, however, be made easier by the availability of a fuzzy logic programming environment (Chiu and Togai [11]). Furthermore, the adaptive characteristics of the algorithm employed makes the problem of determining a very good initial set of rules less important.

The controller program used almost all of the 640 KB of available memory with a cycle execution time of about 5 s. There are two reasons for the large memory requirement and the long execution time: the use of a not-yet-optimized version of the experimental language and the fuzzy logic computations. The latter could prohibit the use of the approach for more complex systems; even in this case, however, the availability of a fuzzy logic programming and chip designing environment [11] could help to overcome the problem.

---

## References

---

1. Waller, K. V., and Gustafsson, T. K., Fundamental properties of continuous pH control, *ISA Trans.* 22(1), 25–34, 1983.
2. Pajunen, G. A., Comparison of linear and nonlinear adaptive control of a pH process, *IEEE Control Syst. Mag.* 39–44, February 1987.

3. Jacobs, O. L. R., Badran, W. A., Proudfoot, C. G., and While, C., On controlling pH, *IEE Proc. Pt. D* **134**(3), 196–200, 1987.
4. Waller, K. V., and Makila, P. M., Chemical reaction invariants and variants and their use in reactor modeling, simulation, and control, *Ind. Eng. Chem. Proc. Des. Dev.* **20**, 1–11, 1981.
5. Gustafsson, T. K., and Waller, K. V., Dynamic modeling and reaction invariant control of pH, *Chem. Eng. Sci.* **38**(3), 389–398, 1983.
6. Jutila, P., and Orava, P. J., Control and estimation algorithms for physicochemical models of pH processes in stirred tank reactors, *Int. J. Syst. Sci.* **12**(7), 855–875, 1981.
7. Jutila, P., An application of adaptive pH-control algorithms based on physicochemical modelling in a chemical waste-water treatment plant, *Int. J. Control* **38**(3), 639–655, 1983.
8. Sugeno, M. (Ed.), *Industrial Applications of Fuzzy Control*, North-Holland, Amsterdam, 1985.
9. Galluzzo, M., and Giarratano, A., An expert fuzzy controller of a pH neutralization process, *Achema 88*, Frankfurt, W. Germany, June 1988.
10. Galluzzo, M., and Giarratano, A., An expert fuzzy controller for pH control, *Chim. Ind.* **71**(5), 52–56, 1989.
11. Chiu, S., and Togai, M., A fuzzy logic programming environment for real-time control, *Int. J. Approx. Reasoning* **2**, 163–175, 1988.