RI 9563

REPORT OF INVESTIGATIONS/1995

An Adaptive System for Process Control

UNITED STATES DEPARTMENT OF THE INTERIOR

UNITED STATES BUREAU OF MINES

U.S. Department of the Interior Mission Statement

As the Nation's principal conservation agency, the Department of the Interior has responsibility for most of our nationally-owned public lands and natural resources. This includes fostering sound use of our land and water resources; protecting our fish, wildlife, and biological diversity; preserving the environmental and cultural values of our national parks and historical places; and providing for the enjoyment of life through outdoor recreation. The Department assesses our energy and mineral resources and works to ensure that their development is in the best interests of all our people by encouraging stewardship and citizen participation in their care. The Department also has a major responsibility for American Indian reservation communities and for people who live in island territories under U.S. administration.

Report of Investigations 9563

An Adaptive System for Process Control

By C. L. Karr, E. J. Gentry, and D. A. Stanley

UNITED STATES DEPARTMENT OF THE INTERIOR
Bruce Babbitt, Secretary

BUREAU OF MINES
Rhea Lydia Graham, Director

This report is based upon research conducted under an agreement between the University of Alabama and the U.S. Bureau of Mines.

00
11

International Standard Serial Number
ISSN 1066-5552

000
111

CONTENTS

Page

ILLUSTRATIONS

UNIT OF MEASURE ABBREVIATIONS USED IN THIS REPORT

M	molar	(mL/s)/s	milliliter per second per second
MHz	megahertz	mol/L	mole per liter
mL	milliliter	pct	percent
mL/s	milliliter per second	s	second

OTHER ABBREVIATIONS AND ACRONYMS USED IN THIS REPORT

A	acidic	MB	mildly basic
B	basic	*N*	neutral
COA	center of area	Q_{ACID}	volumetric flow rate of acid
FLC(s)	fuzzy logic controller(s)	Q_{BASE}	volumetric flow rate of base
GA(s)	genetic algorithm(s)	S	small
L	large	VA	very acidic
MA	mildly acidic	VB	very basic

AN ADAPTIVE SYSTEM FOR PROCESS CONTROL

By C. L. Karr,[1] E. J. Gentry,[2] and D. A. Stanley[3]

ABSTRACT

Researchers at the U.S. Bureau of Mines (USBM) have developed adaptive process control systems in which genetic algorithms (GA's) are used to augment fuzzy logic controllers (FLC's). GA's are search algorithms that rapidly locate near-optimum solutions to a wide spectrum of problems by loosely modeling the search procedures of natural genetics. FLC's are rule-based systems that efficiently manipulate a problem environment by modeling the "rule-of-thumb" strategy used in human decision-making. Together, GA's and FLC's include all of the capabilities necessary to produce powerful, efficient, and robust adaptive control systems. To perform efficiently, such control systems require a ***control element*** to manipulate the problem environment, an ***analysis element*** to recognize changes in the problem environment, and an ***adaptive element*** to adjust to the changes in the problem environment. The control system also employs a computer simulation of the problem environment. Details of an overall adaptive control system are discussed. A specific laboratory acid-base pH system is used to demonstrate the ideas presented; all results are from the physical laboratory system and not from a computer simulation.

[1]Mechanical engineer.
[2]Computer clerk (now with SEER Technology, Cary, NC).
[3]Supervisory research chemist.
Tuscaloosa Research Center, U.S. Bureau of Mines, Tuscaloosa, AL.

INTRODUCTION

The need for efficient process control has never been more important than it is today because of economic stresses forced on industry by processes of increased complexity and by intense competition in a world market. No industry is immune to the cost savings necessary to remain competitive; even traditional industries such as mineral processing (*1*),[4] chemical engineering (*2*), and wastewater treatment (*3*) have been forced to implement cost-cutting measures. Cost cutting generally requires the implementation of emerging technology that is often more application complex than established techniques. The processes that result from the new technology often experience rapidly changing process dynamics. Such processes prove difficult to control with conventional strategies, because these strategies lack an effective means of adapting to changes in the problem environment. Furthermore, the mathematical tools employed for process control can be unduly complex even for simple systems.

Years of research have gone into the development of both open- and closed-loop controllers, and this research has yielded an entire field of study, the field of process control. This field has produced a number of classical process control techniques that result in very efficient nonadaptive controllers, including industry standards such as proportional-integral (PI), proportional-derivative (PD), and proportional-integral-derivative (PID) controllers. Details of these control systems have been well documented and can be found in classical control theory texts such as the one by Coughanowr and Koppel (*4*).

When used to manipulate systems characterized by rapidly changing process dynamics, conventional nonadaptive feedback controllers have traditionally been tuned for the worst case scenario to provide satisfactory performance over all operating conditions (*5*). This approach often yields sluggish response times or produces controllers that generally perform poorly. To accommodate changing process dynamics yet avoid sluggish response times, i.e., to perform at an acceptable level, adaptive control systems are needed that are capable of altering their approach to process control according to the current state of the process.

Early attempts at developing adaptive controllers merely altered conventional nonadaptive control systems. For instance, conventional PID controllers were made adaptive using various strategies. Details of two such strategies appear in articles by Astrom, and others (*6*) and by Clarke and Gawthrop (*7*). Unfortunately, the severe demands placed on control systems by industrial processes with rapidly changing dynamics tax adaptive PID controllers to their limit, thereby pointing to the need for innovative techniques.

Modern technology in the form of high-speed computers and artificial intelligence (AI) has opened the door for the development of control systems that adopt the approach to adaptive control used by humans, and perform more efficiently and with more flexibility than other systems designed to date. Two important tools that have emerged from the field of AI are expert systems and genetic algorithms. These tools can be used to augment conventional control systems, but more significantly they can be used to develop entirely new adaptive control system designs.

Expert systems have become increasingly popular as practical applications of AI. These rule-based systems have performed as well as humans in several problem domains (*8*); however, their lack of flexibility in representing the subjective nature of human decision-making limits their performance in process control problems. Expert systems can be provided with the means to model the uncertainty inherent in human decision-making via fuzzy set theory (*9*). Zadeh developed fuzzy set theory in an attempt to circumvent the complexity associated with more traditional mathematical tools required in control theory. In fuzzy set theory, abstract or subjective concepts are represented with ***fuzzy linguistic variables***, terms like "very high" and "not quite low." Fuzzy linguistic variables have been incorporated into expert systems to form fuzzy logic controllers (FLC's) which are being used successfully in an increasing number of application areas (*10-11*). Like expert systems, FLC's include rules to direct the decision-making process, but they also include ***membership functions*** that convert linguistic variables into the precise numeric values computers require for the implementation of a control strategy. The rule set is composed of production rules (rules of the form IF <*condition*> THEN <*action*>) and can be gleaned from a human expert's knowledge, which has been gained through the personal experience of working with the problem environment. The membership functions are defined to represent the expert's understanding of the fuzzy linguistic variables and to provide these fuzzy variables with concrete meaning. It is often difficult to define suitable membership functions for control purposes. Genetic algorithms (GA's) offer one method of easily producing precise membership functions.

GA's are search techniques based on natural genetics; they use operations found in natural genetics to guide their trek through a search space. GA's search large spaces quickly, requiring only objective function values to guide

[4] Italic numbers in parentheses refer to items in the list of references at the end of this report.

their search, an inviting characteristic since most commonly used search techniques require derivative information, continuity of the search space, or complete knowledge of the objective function to guide their search. Furthermore, GA's take a more global view of the search space than many methods encountered in engineering optimization (12). The immense potential of GA's lies in their ability to perform efficiently across a broad spectrum of search problems (*12-13*). They are of interest to researchers striving to design adaptive control systems because of their proven ability to adjust to environmental changes, much as living organisms adjust to changes in their own environment.

Both expert systems and GA's have been used successfully to produce efficient process control systems. These AI-based tools have in fact been used to provide conventional nonadaptive control systems with adaptive capabilities. The resulting systems have outperformed their nonadaptive counterparts in some applications.

The virtual explosion in the popularity of expert systems has seen them utilized in a number of application areas, and the area of process control has been no exception. There have been several instances in which an expert system was successfully used to improve the performance of a controller. However, the most straightforward and perhaps the most effective way to utilize an expert system to produce an adaptive controller is to use a rule-based system to alter the gain constant associated with a PID controller. Control systems that adopt this approach have been developed by Krauss and Myron (*14-15*), who used an expert system to alter the gain constant in response to changes in the problem environment.

GA's have also been introduced into the design of adaptive control systems. Odetayo and McGregor (*16*) used a GA to select rules for a control system that was based on a conventional expert system. Furthermore, Valenzuela-Rendon (*17*) developed a fuzzy classifier system that, in essence, used a GA to learn a rule set for a controller. Also, researchers like Procyk and Mamdani (*18*) have used a derivative-based approach to alter the rules of an FLC. Galluzzo and others (*19*) used an independent set of rules, called metarules, to alter the rule set of an FLC. All of the above are examples in which the control system receives substantial feedback concerning the changes in the problem environment. Often, because of inadequate sensors, systems must cope with inadequate feedback.

Researchers at the U.S. Bureau of Mines (USBM) have addressed the issue of inadequate feedback. This problem occurs rather frequently in the mineral processing industry, in which it is often impractical or too costly to measure all of the variables involved in a process. Thus, USBM researchers must address the problem of inadequate feedback. Thus, researchers have developed an overall approach to the design of adaptive control systems, based on GA's, that is effective in systems in which minimal feedback concerning the state or condition of the problem environment is available. Since the control systems that result must make accommodations for this lack of feedback, an overall structure is used that is more suitable to the tasks of recognizing, quantifying, and adapting to changes in the problem environment than control systems of the past have been.

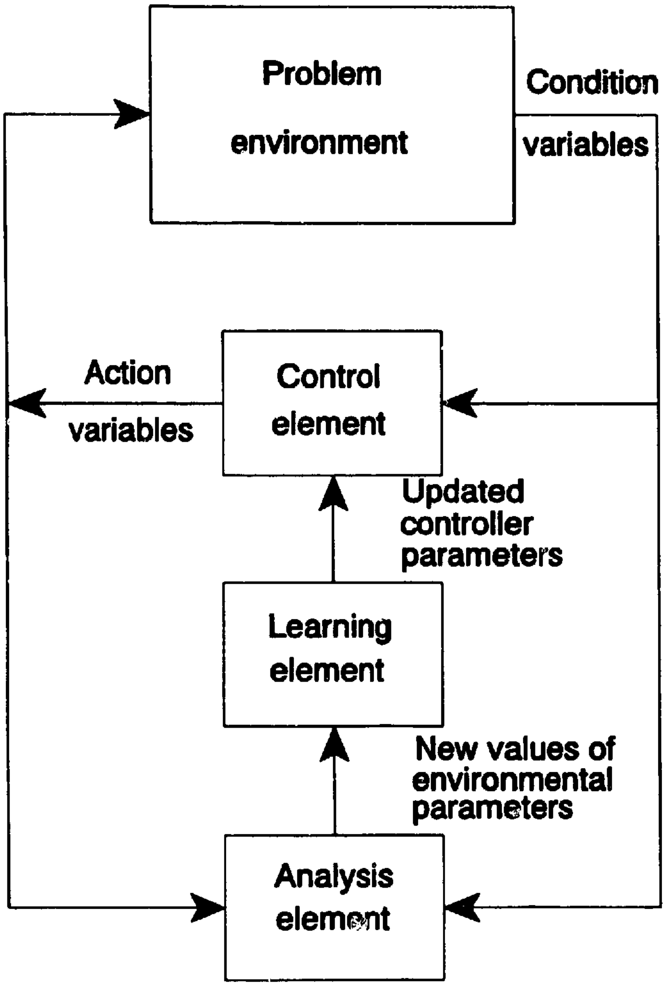
The adaptive control systems developed at the USBM include three components: a ***control element*** to manipulate the problem environment, an ***analysis element*** to recognize changes in the problem environment, and an ***adaptive element*** to adjust to the changes in the problem environment. Each of these components employs either a GA or an FLC. When both are employed, they are used in a unique fashion. Unlike the work reported by Odetayo and McGregor (*16*) wherein a GA was used to alter the rules associated with an FLC, the USBM-developed approach uses a GA to alter the membership functions associated with an FLC. This approach has been shown to be effective in a number of problem environments (*20*).

Each of the three components mentioned above are discussed. Furthermore, a particular problem environment, a laboratory pH system, is introduced to serve as a forum for the details of the USBM-developed adaptive controller. This pH system includes nonlinearities due to the logarithmic scale of pH, and changing process dynamics due to buffering and to alterations in the concentrations of the acid and base used to manipulate the system pH. Although the complete control system is still being developed, results are presented here to demonstrate the effectiveness of using GA's and FLC's for adaptive process control.

STRUCTURE OF AN ADAPTIVE CONTROLLER

Figure 1 shows a schematic of the USBM's adaptive control system. At the heart of this control system is the loop consisting of the control element and the problem environment. The control element receives information from the problem environment concerning the status of the condition variables (those variables on which proper control actions are based). It then computes a desirable state for a set of action variables (those variables that can

Figure 1



Schematic showing requisite elements of stand-alone, comprehensive adaptive control system.

be changed by the controller to alter the state of the problem environment), which force the problem environment toward a setpoint (the desired state). This is the basic approach adopted for the design of virtually any closed-loop control system, yet such a system includes no mechanism for adaptive control.

The adaptive capabilities of the system shown in figure 1 occur in the lower loop and are associated with information exchange between each of the three individual elements of the loop. This information exchange includes several different facets. The analysis element receives information concerning the condition variables from the environment and receives information concerning the action variables from the control element. The analysis element uses the information it receives to compute the changes that have occurred in the problem environment. It then passes information concerning the computed changes to the learning element, which uses the information to prescribe alterations to the control element. This information exchange allows for the completion of some necessary tasks in well defined steps.

In general, the analysis element must recognize when a change in the problem environment has occurred. A "change," as it is used here, is an alteration to a parameter in the problem environment other than one of the condition or action variables of the rule set. Changes in parameters other than the condition and action variables cannot be accounted for by the control element (see figure 1). Also, the change must affect the response of the problem environment; otherwise it has no effect on the way in which the control element must act to efficiently manipulate the problem environment. The analysis element requires information concerning the condition and action variables over some finite time period to recognize changes in the environment and to compute the new performance characteristics associated with these changes. The new environment (the problem environment with the altered parameters) can pose all kinds of difficulties for the control element, because the control element is no longer manipulating the environment for which it was designed. Therefore, the algorithm that drives the control element must be in some way altered. As shown in the schematic of figure 1, this task is accomplished by the adaptive element. The most efficient approach for altering the control element is to utilize information concerning the past performance of the control system.

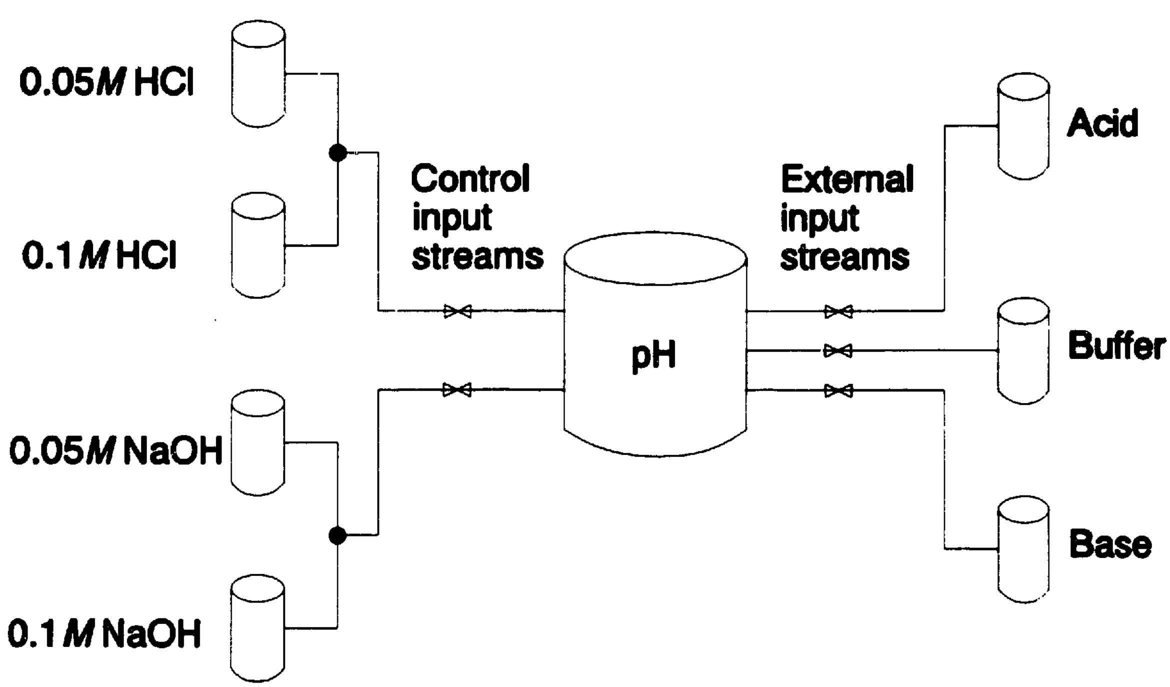
This section has described, in abstract terms, the basic structure of an adaptive controller. In the next section, a particular problem environment, a pH system, is introduced to serve as a forum for presenting the details of a stand-alone, comprehensive adaptive controller being developed by USBM researchers.

PROBLEM ENVIRONMENT

The problem environment is a laboratory pH system representative of pH systems present in a number of mineral and chemical industries (*1*, *21*). The fundamental goal of the control system is to drive the pH to some setpoint. The pH system contains nonlinearities and changing process dynamics, and it is an extension of a system studied by Galluzzo and others (*19*). The nonlinearities occur because the output of pH sensors is proportional to the logarithm of hydrogen ion concentration. The changing process dynamics has three separate causes. First, there is a mechanism for introducing a buffer to the system that significantly alters the manner in which the pH responds as acid or base is added. Second, the concentrations of the acid and base that the controller uses to manipulate the pH of the system can be altered. Third, the setpoint of the system, the desired value to which the system pH is to be driven, can be altered. The system studied by Galluzzo and others (*19*) included a mechanism for the addition of a buffer. However, the current problem environment is more difficult to control because of changing concentrations of the input acid and base, and because of possible changes in the system setpoint.

A schematic of the pH system under consideration is shown in figure 2. The system includes a beaker initially containing a given volume of a solution having some known pH. There are five valved input streams into the beaker. Only the valves controlling the two ***input control streams*** can be adjusted by the controller. The hydrogen ion concentration of these two control input streams can be changed by some "random agent" to be either 0.1*M* HCl or 0.05*M* HCl and 0.1*M* NaOH or 0.05*M* NaOH. The control element has no knowledge of the changes made in these concentrations by the random agent; it is left up to the analysis element to recognize that the concentrations have changed and to determine what the new concentrations are. The valves on the other three input streams are used to manipulate ***external streams***, which are altered by the same random agent that manipulates the concentration of the control input streams. Thus, the problem environment can be manipulated by the control system, or by an external agent. Certainly, the control system must be able to control the problem environment despite the changes made by the external agent. Furthermore, the control system has no knowledge of the changes being made by the external agent. The three external streams include (1) 0.05*M* HCl, (2) 0.05*M* CH_3COONa, and (3) a buffer (a combination of 0.1*M* CH_3COOH and 0.1*M* CH_3COONa). Additionally, the random agent is capable of changing the desired setpoint to which the system pH is to be driven. The existence of the random agent allows for alterations in the system parameters that dramatically alter the way in which the problem environment reacts to adjustments made by the controller to the valves on the control streams. Like the changes in the concentrations of the control input streams, the magnitude of the changes to the external streams must be recognized by the analysis element.

Figure 2



Schematic of problem environment, which is a laboratory pH system that includes nonlinearities and changing process dynamics.

In light of the above description of the pH system, the goal of the control problem is to drive the system pH to the desired setpoint in the shortest time possible by adjusting the valves on the two control input streams. Furthermore, the valves on the input streams are to be fully closed when the target pH value has been achieved. As a constraint on the control problem, the valves can only be adjusted a limited amount (0.5 (mL/s)/s, which is 20 pct of the maximum flow rate of 2.5 mL/s), thereby, restricting pressure transients in the associated pumping systems.

The pH system was designed on a small scale so that experiments can be performed in limited laboratory space. Titrations were performed in a 1,000-mL beaker using a magnetic bar to stir the solution. Computer-driven peristaltic pumps were used for the five input streams. An industrial pH electrode and transmitter sent signals through an analog-to-digital board to a 33-MHz 386 personal computer, which implemented the control system.

CONTROL ELEMENT

In this section, an FLC control element for the pH system is described. The step-by-step details concerning the development of the FLC are presented in such a way as to make them easily extensible to other problem environments.

Like conventional expert systems, FLC's use a set of production rules that are of the form:

IF <*condition*> THEN <*action*>

to arrive at appropriate control operations. The left-hand side of the rules (the *condition* side) consists of combinations of the controlled variables; the right-hand side of the rules (the *action* side) consists of combinations of the manipulated variables. Unlike conventional expert systems, FLC's use rules that utilize fuzzy terms like those appearing in human rules-of-thumb. For example, a valid rule for an FLC used to manipulate the pH system is

IF <ph is VERY ACIDIC and ΔpH is SMALL> THEN <Q_{BASE} is LARGE and Q_{ACID} is ZERO>.

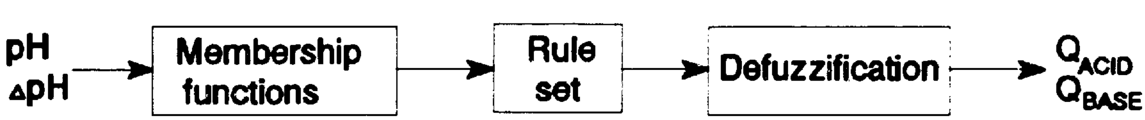
This rule says that if the solution is very acidic and is not changing rapidly, the flow rate of the base should be made to be large and the flow rate of the acid should be made to be zero.

The fact that FLC's use fuzzy terms gives rise to another fundamental difference with conventional expert systems: FLC's provide a mechanism for a particular value of a condition variable to be described by more than one fuzzy term. For instance, a system pH of 5 can be described by both the fuzzy terms "very acidic" and "mildly acidic." This is appropriate because the line between these two descriptive terms is not definite; a subjective decision is required. FLC's use fuzzy membership functions to allow particular values of the condition variables to be described, to some degree, by each of the fuzzy terms. Therefore, more than one rule is qualified, or eligible, to enact its action at any given time, i.e., to "fire."

Since more than one rule can have its condition met at any given time, FLC's must include a mechanism for determining a single control action. Typically, a weighted average of the actions prescribed by the appropriate rules is calculated. The emphasis placed on each rule's action is based on the confidence that exists in the condition portion of each of the rules (the degree to which a concrete value is described by a fuzzy term). The weights used in this averaging technique come directly from the fuzzy membership functions that provide the fuzzy terms with meaning.

Figure 3 shows a schematic of the structure of a control element composed of an FLC. The FLC receives definite values of the condition variables, uses fuzzy membership functions to characterize the definite values with fuzzy terms (it "fuzzifies" or makes less precise the variable values), employs a rule set, and computes one definitive action to be taken on the problem environment by calculating a weighted average (it "defuzzifies" or defines the prescribed actions). This process is clarified in the following paragraphs as the membership functions and the rule set used in the pH system are set forth.

Figure 3



Structure of control element. The control element receives values of pH and ΔpH from the problem environment, "fuzzifies" these values with membership functions, employs a rule set, and uses a COA method for "defuzzification," thereby resulting in a single value for the manipulated variables.

The initial phase of FLC development is common to the development of any control system: the appropriate condition and action variables must be determined. There are numerous condition variables that could be considered in the pH system (pH of solution in the tank, flow rates of the input streams, concentrations of input solutions, volume in the tank, and many others). However, it is important to limit the number of condition variables used because the size of a rule set increases multiplicatively with the number of condition variables.

Fortunately, the effect of some of the potential condition variables can be accounted for by the adaptive capabilities of the control system. After a period of experimentation (an inevitable requirement for the development of a quality FLC), two condition variables were selected: the current value of pH in the beaker and the absolute value of the current time-rate-of-change of the pH in the tank (ΔpH). The fact that this particular pH system with all of its changing dynamics can be controlled when but two condition variables are considered demonstrates the power of an adaptive control system.

The determination of the action variables is relatively straightforward, because there are basically only two things that can be altered by the control element: the valve setting (and thus the flow rate) associated with each of the two control input streams. Therefore, the two action variables of the input streams were the flow rates for the acid (Q_{ACID}) and the base (Q_{BASE}). The selection of the action variables differs from the selection of the condition variables in that the number of action variables has no effect on the number of rules required by an FLC. Therefore, no improvements in computational efficiency are achieved by limiting the number of action variables considered.

Next, fuzzy linguistic variables are selected to represent the condition and action variables. After further experimentation, seven terms were selected to describe pH, two terms were selected to describe ΔpH, and five terms were selected to describe both Q_{ACID} and Q_{BASE}. The specific linguistic terms used to describe the pertinent variables in the pH system follow:

pH

- Very Acidic (VA)
- Acidic (A)
- Mildly Acidic (MA)
- Neutral (N)
- Mildly Basic (MB)
- Basic (B)
- Very Basic (VB)

ΔpH

- Small (S)
- Medium (M)
- Large (L)

Q_{ACID} and Q_{BASE}

- Zero (Z)
- Very Small (VS)
- Small (S)
- Medium (M)
- Large (L)

These fuzzy terms are subjective, but the developer of the pH FLC has some concept of what they mean in the context of the physical system to be controlled.

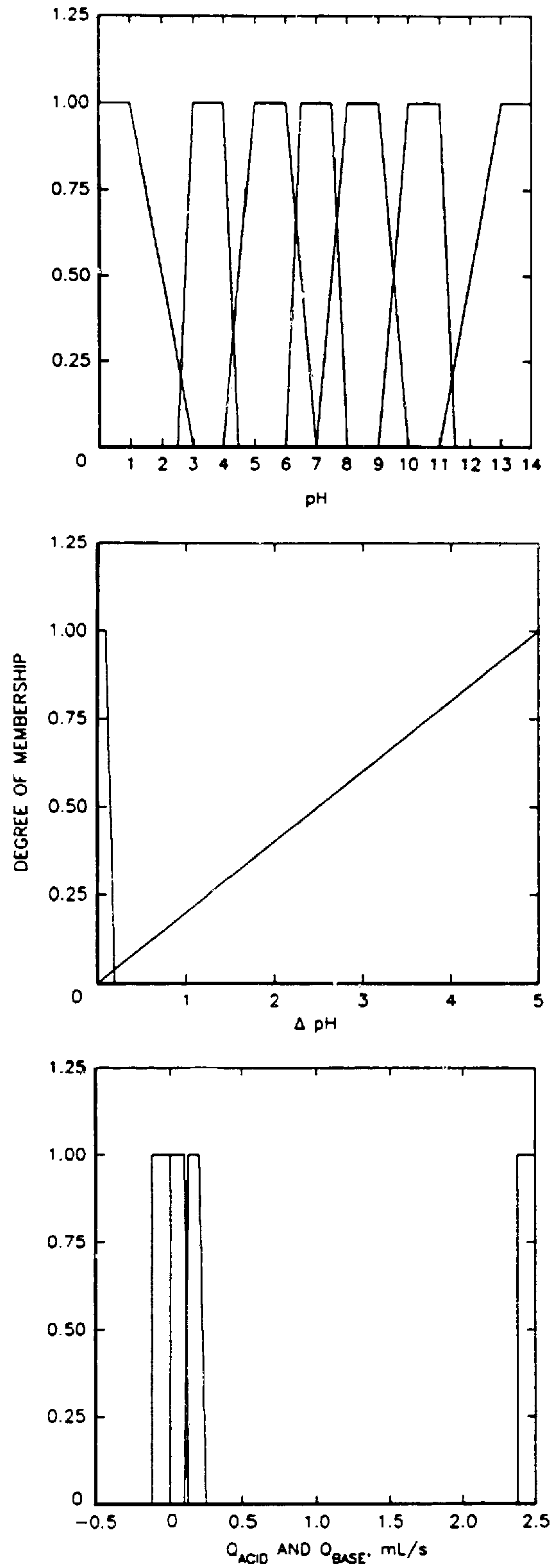
The developer's conception of the linguistic terms is described by the membership functions that must be defined to give the terms meaning. The initial membership functions used in the FLC appear in figure 4. These membership functions are later altered by the adaptive element, via a GA, in response to changes occurring in the pH system. As will be seen, alterations in these functions can dramatically change the performance of the FLC.

Although the laboratory pH system is complex, an effective pH FLC can be written that contains only 14 rules. The 14 rules are necessary because there are 7 fuzzy terms describing the pH and 2 fuzzy terms describing ΔpH ($7 \times 2 = 14$ rules to describe all possible combinations that could exist in the pH system as described by the fuzzy terms represented by the membership functions shown). The entire rule set for the pH FLC is shown in figure 5.

Now, the only aspect of the initial FLC design that is left is the technique for determining one value at which to set the flow rates of the input acid and base streams. The popular center-of-area (COA) method (*22*) is used. This method provides a convenient way to compute a weighted average of the different control actions prescribed by the rules that are eligible to fire. The COA method results in the selection of a single control action to be taken on the problem environment.

There is one detail associated with the pH system being considered that warrants special mention. There is a limit on the allowable change in the flow rates of the input streams; i.e., the flow rates cannot change by more than 0.5 (mL/s)/s. However, the membership functions describing the action variables used in the COA method (shown in figure 4) allow for values of Q_{ACID} and Q_{BASE} to range between 0.0 and 2.5 mL/s. The constraint is

Figure 4



Initial membership functions. The linguistic terms are defined by membership functions for pH, ΔpH, and Q_{ACID} and Q_{BASE}.

Figure 5

Q_{ACID}

		Δ pH S	Δ pH L
	VA	Z	Z
	A	Z	Z
	MA	Z	Z
pH	N	Z	Z
	MB	S	VS
	B	M	S
	VB	L	M

Q_{BASE}

		Δ pH S	Δ pH L
	VA	L	M
	A	M	S
	MA	S	VS
pH	N	Z	Z
	MB	Z	Z
	B	Z	Z
	VB	Z	Z

Rule set for pH FLC. The rule set must include a rule for all of the possible combinations of the condition variables as described by the chosen linguistic terms.

imposed by computing the value of the flow rates using the COA method. If this value exceeds the constrained flow rate, the flow rate is changed by the maximum allowable value of 0.5 mL/s (for either increases or decreases in flow rate).

The preceding has been a general description of the makeup of a pH FLC. The following is a step-by-step procedure for the implementation of an FLC as outlined in a paper by Karr (*20*):

1. Determine the condition variables to be considered.
2. Determine the action variables to be considered.
3. Describe the fuzzy sets for both the condition and action variables.
4. Establish a set of fuzzy production rules that cover all of the possible conditions that could exist in the problem environment.
5. Define the fuzzy membership functions.
6. Compare the set of conditions existing in the problem environment to the production rules, and use a weighted average to select a single action to be taken on the problem environment (recall that the weights are proportional to the minimum degree of membership for the conditions associated with each rule).

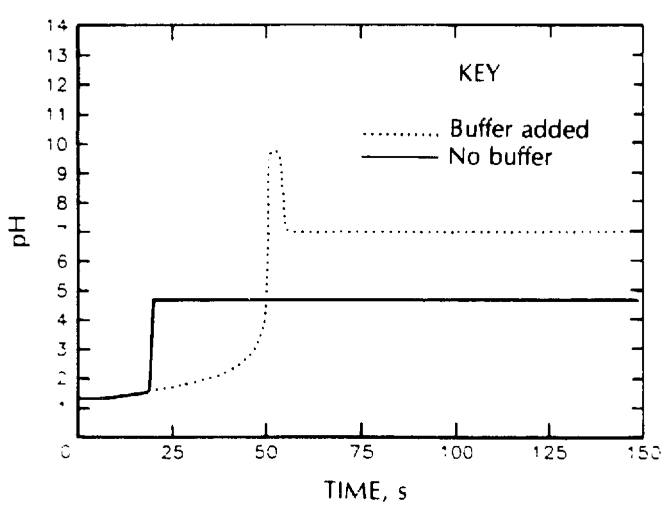
7. Continue with step 6 as long as necessary. The procedure is repeated until a specified time limit is reached, or until the system is at its setpoint. An efficient FLC will maintain equilibrium once the setpoint has been reached.

To those readers who are unfamiliar with the operation of FLC's, this approach may seem awkward. However, it allows for the development of powerful control systems. Figure 6 demonstrates the ability of the FLC to effectively drive the system pH to a setpoint of 7, as long as the process dynamics are not altered. However, it is apparent from this figure that when the process dynamics are altered (in this case, they are altered by adding a buffer) an adaptive controller becomes essential. In figure 6, it is important to realize that the nonadaptive controller would eventually drive the buffered system to the setpoint of 7, but this task can be accomplished in much less time.

ANALYSIS ELEMENT

The analysis element must recognize changes in parameters associated with the problem environment that are not taken into account by the rules of the control element. In the pH system, these parameters include (1) the concentrations of the acid and base of the input control streams, (2) the flow rates of the acid, the base, and the buffer that are altered by an external agent, and (3) the system setpoint. Changes to any of these parameters can dramatically alter the way in which the system pH reacts to additions of acid or base, thus forming a new problem environment. Recall that the FLC used for the control element includes none of these parameters in its 14 rules. Therefore, some mechanism for altering the prescribed actions must be included in the control system. However, before the control element can be altered, the control system must recognize that the problem environment has been changed and compute the nature and magnitude of the changes.

Figure 6



Effectiveness of FLC. The FLC is able to effectively drive the system pH to the setpoint of 7 as long as the process dynamics remain constant. However, when the process dynamics are altered (in this case a buffer was added), an adaptive controller is needed.

The direct way to recognize changes occurring in the problem environment is to receive feedback directly from a set of transducers, in much the same way information concerning the condition variables is received. In such a case, no analysis element is necessary. However, the requisite feedback is not always available, and it is this situation that presents the control system with a need for an analysis element.

In general, recognizing changes in the parameters associated with the problem environment requires the control system to store information concerning the past performance of the problem environment. This information is most effectively acquired through either a data base or a computer model. Storing such an extensive data base can be cumbersome and requires extensive computer memory. Therefore, the more practical approach is to use a computer model to predict the response of the problem environment and compare the predicted response with the actual response at specified times.

Fortunately, the dynamics of the pH system are well understood for buffered reactions and can be modeled using a single cubic equation (*23*) that can be solved for $[H_3O^+]$ ion concentrations, to directly yield the pH of the solution:

$$x^3 + Ax^2 + Bx + C = 0,$$

where

$$x = [H_3O^+],$$

$$A = k_a + [CH_3COONa] + [NaOH] - [HCl],$$

$$B = k_a[NaOH] - k_a[HCl] - k_a[CH_3OOH] - k_w,$$

$$C = -k_a k_w,$$

k_a = 1.8×10^{-5}, equilibrium constant for CH_3COOH,

k_w = 1.0×10^{-14}, equilibrium constant for H_2O,

and bracketed terms ([]) = molar concentrations.

Further details of the computer model appear in a paper by Karr and Gentry (*24*).

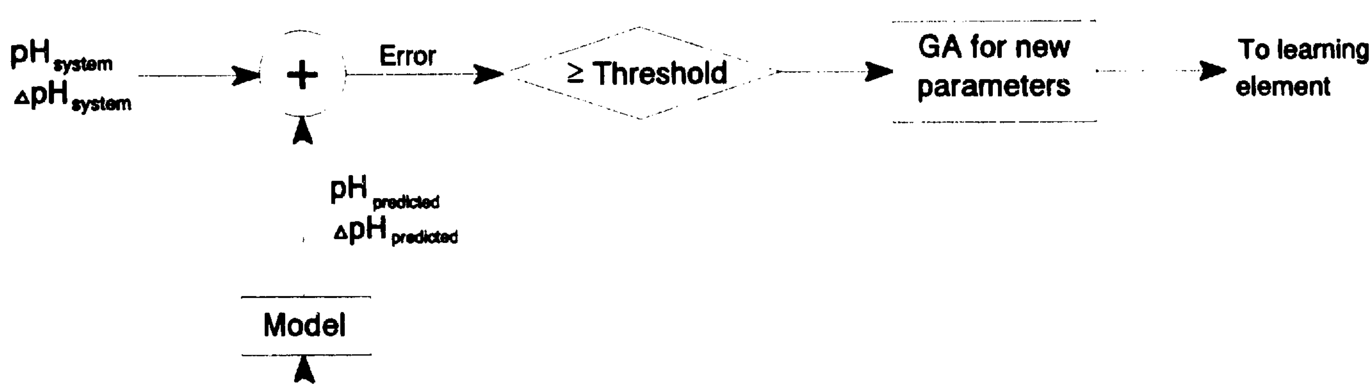
Figure 7 shows a schematic of an analysis element. In the approach represented by the schematic, a computer model represents the actual problem environment. The response of the physical pH system is compared with the response of the pH system as predicted by the computer model. When these responses differ by some threshold over a substantial period of time, the parameters of the pH system have changed and the model must be updated. Certainly, the threshold and the "substantial" period of time depend on the problem environment. For the pH system considered, when the pH predicted by the model differed from the actual pH of the physical system by 1 unit of measure or more for a period of 5 s, the pH system parameters were considered to have changed.

When the above approach is adopted, the problem of computing the new system parameters becomes a curve-fitting problem. The parameters associated with the computer model produce a particular response to changes in the action variables. The parameters must be selected so that the response of the model matches the response of the actual problem environment. As in the area of process control, curve fitting has received a great deal of attention through the years. There are a number of traditional curve-fitting techniques, and the details of these techniques can be found in texts on numerical analysis such as the one by Press and others (*25*). However, the best choice of an appropriate curve fitting technique is problem dependant, and since the objective of this report is to present AI-based techniques for process control, a novel approach to curve fitting is used.

Karr and others (*26*) have demonstrated the effectiveness of using a GA to perform curve fitting. Basically, curve fitting is a search problem: the parameters that produce a particular system response must be located. GA's are efficient search algorithms that require a minimum of information from the problem domain. Furthermore, GA's have a demonstrated capability in the area of model parameter identification (*27*).

A simple GA that has given good results in a variety of engineering problems uses three operators: reproduction, crossover, and mutation. These operators are implemented by performing the basic tasks of copying bit strings (strings made up of 1's and 0's), exchanging portions of strings, and generating random numbers. Before looking at the operators, consider the overall processing of a GA in light of a curve-fitting problem such as the problem of locating the parameters appearing in the previous cubic equation.

Figure 7



Schematic of analysis element shows that the response of the problem environment (the pH system) is compared with a model's predicted response. When the difference is larger than a threshold value, new system parameters are computed.

The operation of a GA begins with the creation of an initial generation of N strings each of length m. Each bit string represents one possible combination of the unknown parameters associated with the curve-fitting problem. Representing a parameter set as a bit string is akin to the way in which genetic information concerning an organism's composition is contained in a chromosome (the strings) composed of genes (the individual bits). These strings are then decoded, yielding the numerical value of each of the parameters. The parameters are sent to some conceptual framework that yields a measure of the quality of the solution. In the analysis element, this framework is a model of the pH system. The parameter set is then evaluated according to some objective function (fitness function), which is simply a measure of how good the solution is; i.e., how well the parameter set allows the model to predict the actual response of the pH system. Then, a new population of strings is produced via the three genetic operators. This process of producing new generations is continued until some stopping criterion is met. In the analysis element, the stopping criterion is generally based on the time available for the control element to prescribe a new control action.

As stated above, the three genetic operators can be used to produce a powerful GA. Reproduction is a process by which strings with large fitness values (parameter sets that allow for the accurate modeling of the response of the pH system) receive correspondingly large numbers of copies in the new population. For example, in "expected number control" reproduction, those strings with high fitness values f_i are given a proportionately higher probability of reproduction selection, p_{select}, according to the following distribution:

$$p_{select} = \frac{f_i}{\sum f}, \tag{2}$$

where f_i is the value of fitness function associated with an individual string and the denominator represents the summation of the fitness of all of the strings in the current population. Once the strings are reproduced for possible use in the next generation, they are placed in a mating pool (a file or location in computer memory) where they await the action of the other two operators.

The second operator is crossover, which causes a systematic exchange of information between high-quality strings. Crossover proceeds in three steps. First, two newly reproduced strings are selected from the mating pool of strings that were formed through reproduction. Second, a position along the two strings is selected at random. For example, the following binary coded strings A and B of length 10 are shown aligned for crossover:

A = 110 1010000.
B = 001 0111111.

Notice how crossing site three has been selected in this particular example through random choice, although any of the other eight positions were just as likely to have been selected. The third step is to exchange all characters following the crossing site. A' and B' are two new strings following this crossing:

A' = 110 0111111.
B' = 001 1010000.

String A' is made up of the first part of string A and the tail of string B. Likewise, string B' is made up of the first part of string B and the tail of string A. Although crossover has a random element, it should not be thought of as a random walk through the search space. When combined with reproduction, it is an effective means of exchanging information and combining portions of high-quality solutions.

Reproduction and crossover give GA's most of their search power. The third operator, mutation, enhances a GA's ability to find near-optimum solutions to the search problem. Mutation is the occasional alteration of a value at a particular string position, or more to the point, it is an insurance policy against the permanent loss of any simple bit. This loss occurs when a generation is created void of a particular character at a given string position. For example, a generation may exist that does not have a 1 in the third string position when, because of the chosen coding, a 1 in the third position may be critical to obtaining a quality solution. Under these conditions, neither reproduction nor crossover will ever allow for the production of a 1 in this third position in subsequent generations. However, mutation causes a 0 in the third position to occasionally be changed to a 1. Thus, the critical piece of information can be reinstated into the population. Although mutation can serve a vital role in a GA, it occurs with a small probability (on the order of one mutation per 1,000 string positions) and is secondary to reproduction and crossover.

At this point, an analysis element has been forged in which a GA, as described above, is used to compute the model parameters necessary to accurately predict the response of the laboratory pH system. When using a GA for a search problem, there are basically two decisions that must be made: (1) how to code the possible solutions to the search problem as bit strings and (2) how to evaluate the merit of the possible solutions. The parameters that must be coded in this instance are the concentrations of the input acid and base, and the flow rates of the three external streams. It has been reported that binary codings (the use of bit strings) produce the most efficient genetic searches (*12*). For this reason, binary coding was used for the 200-bit strings representing the appropriate model parameters. The first 40 bits of the strings were used to represent the concentration of the acid on the control input stream, the second 40 bits were used to represent the concentration of the base on the control input stream, the third 40 bits were used to represent the flow rate of the acid of the external streams, and the final 80 bits were used to represent the flow rates of the buffer and the base of the external streams, respectively. The 40 bits associated with each individual parameter were read as a binary number, converted to decimal numbers (000 = 0, 001 = 1, 010 = 2, 011 = 3, etc.), and mapped between minimum and maximum values according to the following:

$$C = C_{min} + \frac{b}{(2^m - 1)} \left(C_{max} - C_{min}\right), \quad (3)$$

where b is the binary value, m is the number of bits used to represent the particular parameter (40), and C_{min} and C_{max} are minimum and maximum values associated with each parameter that is being coded.

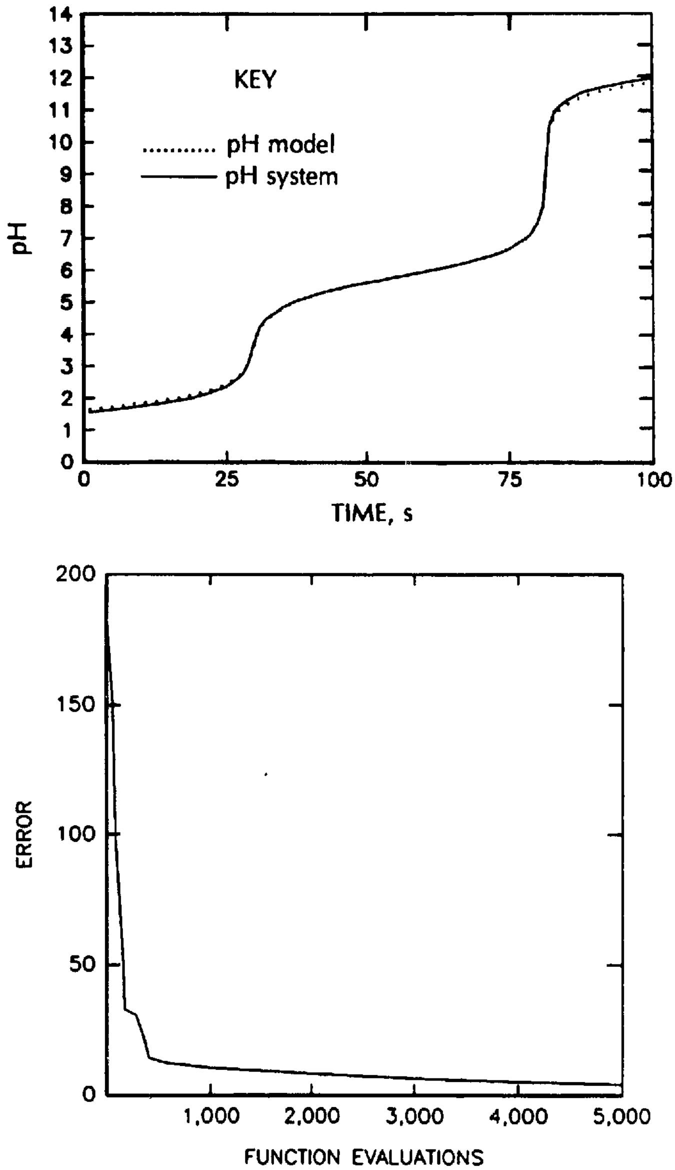
Now that an appropriate coding has been determined, the second issue must be addressed: evaluation of the merit of each string (a possible choice of the model parameters). This task of defining a fitness function to evaluate string merit is always application specific. To select model parameters that accurately mimic the response of the laboratory pH system, an effective fitness function is

$$f = \sum_{i=0s}^{i=100s} \left| pH_{model} - pH_{actual} \right|. \quad (4)$$

With this definition of the fitness function, the problem becomes a minimization problem: the GA must minimize f, which, as it has been defined, represents the difference between the response predicted by the model and the response of the laboratory system.

Figure 8 demonstrates the ability of a GA to locate the parameters needed by the computer model to compute the response of the physical pH system. This figure includes information concerning the performance of the GA in locating these parameters. The GA was able to locate the correct parameters after only 3,000 function evaluations, where a function evaluation consisted of simulating the pH system for 100 s. Locating the correct parameters took approximately 400 s on a 386 personal computer. The physical system often mandates that a control action be taken in less than 400 s. In this case, the time the GA is allotted to update the model parameters can be restricted. In such situations, the model simply must operate with inaccurate parameters until the analysis element is again employed. However, the magnitude of this problem will be diminished as computers become ever faster. And the problem as it now stands is not a major hinderance to the performance of the control system.

Figure 8



Performance of GA. A, A GA is able to locate the parameters associated with the problem environment that allow a computer model to accurately predict the response of the pH system; B, it took the GA approximately 3,000 function evaluations, where a function evaluation consisted of simulating the pH system over 100 s.

The purpose of the analysis element is to recognize changes in the parameters associated with the problem environment that are not accounted for by the control element and to compute the new values of these parameters. Once new parameters (and thus the new response characteristics of the problem environment) have been determined, the adaptive controller must alter the control element.

ADAPTIVE ELEMENT

The adaptive element is responsible for altering the control element in response to changes in the problem environment. Recall that the relevant changes occurring in the pH system include (1) changes in the concentrations of the acid and base of the control input stream, (2) random additions of acid, base, and buffer from the external streams, and (3) changes in the system setpoint. As set forth in a previous section, none of the parameters associated with the above changes are included in the rule set of the FLC that serves as the control element. Therefore, the only way to account for these conditions (outside of completely revamping the system) is to alter the membership functions employed by the FLC. However, in other control systems, there are alternative approaches to implementing adaptive capabilities.

In this report, a means for producing an adaptive FLC is adopted that is different from the approach used by other researchers who alter the rule set used by their FLC. In the approach adopted here, the membership functions (the definition of the fuzzy terms in the rule set) are altered. This approach is more consistent with the way humans control complex systems. Quite often, the rules-of-thumb humans use to manipulate a problem environment remain the same despite even dramatic changes to that environment; only the conditions under which the rules are applied are altered. This is basically the approach that is being taken when the fuzzy membership functions are altered.

The approach developed and implemented by the USBM for using a GA to alter the membership functions associated with an FLC has been well documented (*20, 24*). To implement this approach, the parameters needed to describe the fuzzy membership functions must be coded as bit strings, and the effectiveness of various FLC's must be described with an objective function. As a brief aside, realize that these are the same two issues that must be addressed in any GA application. The parameters that must be coded in the quest for efficient membership functions are the points that define the trapezoids used to describe each of the fuzzy linguistic variables (as defined by the membership functions appearing in figure 4). When the symmetry associated with the pH system (for example, **VERY ACIDIC** and **VERY BASIC** are symmetrical about the neutral point) is considered, there are 32 points that must be defined by the GA. Seven bits were allotted for the representation of each parameter thereby producing strings that are 224 bits long. Each 7-bit group was decoded using the mapping equation presented in the section on the analysis element thereby yielding the 32 parameters needed to completely define a set of membership functions.

The fitness function must indicate the objective of the control system. In the pH system, the objective is to drive the system pH to a desired setpoint in the shortest time possible, and to keep it there. The fitness function used in this application is

$$f = \sum_{i=0s}^{i=100s} | \text{ setpoint } - \text{pH} | , \qquad (5)$$

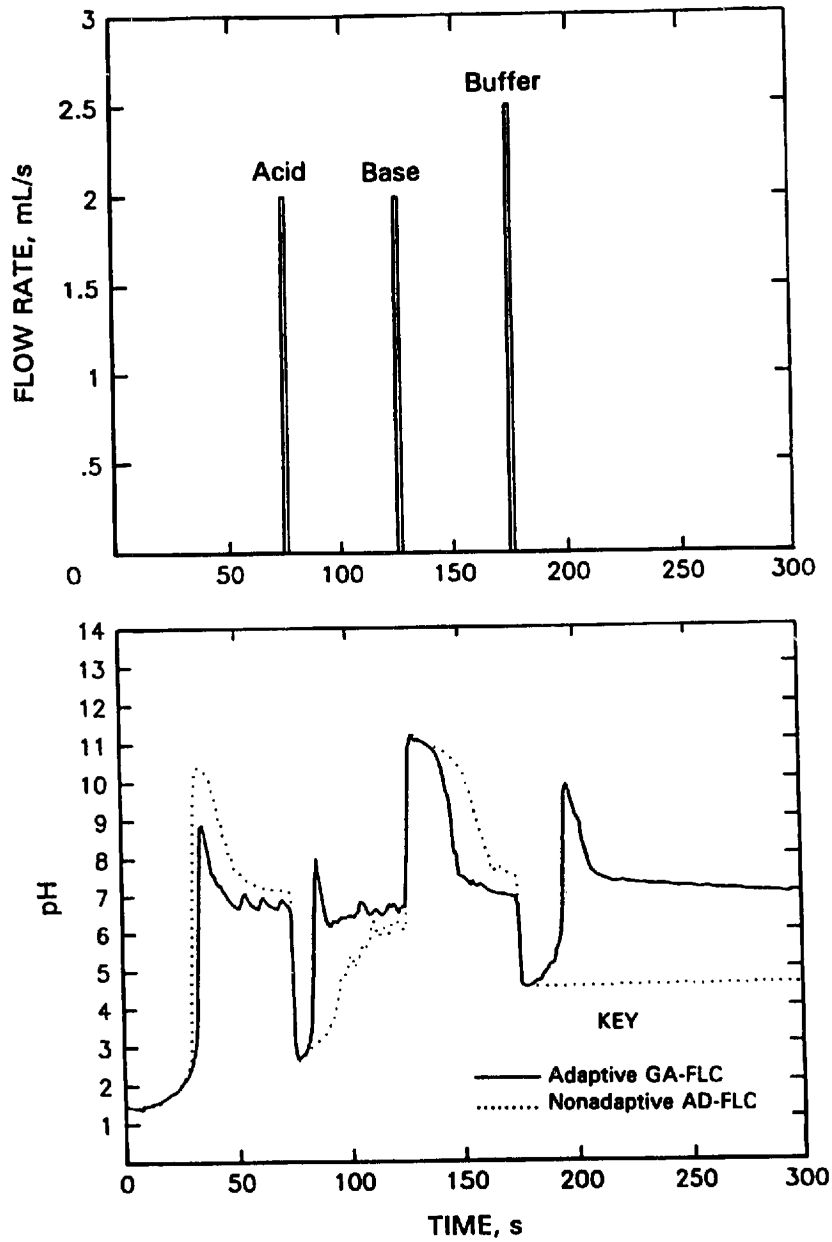
where the summation is performed over a 100-s time period as simulated using the mathematical model of the system, which has been updated by the analysis element. This simulation is initiated from the current state of the laboratory system, i.e., the current values of pH, Q_{ACID}, and Q_{BASE}.

The performance of a control system that uses a GA to alter the membership functions of an FLC is demonstrated for three different situations. First, the pH system is perturbed by the addition of an acid, a base, and a buffer. In this case, the process dynamics are dramatically altered because of the addition of the buffer. Second, the desired setpoint is altered. This actually represents a change in the objective of the controller. Third, the concentrations of the acid and base that the FLC uses to control pH are changed (those from the control input streams), which causes the system to handle differently. For example, if the 0.1*M* HCl is the control input, the pH falls a certain amount when this acid is added. However, all other factors being the same, the pH will not fall as much when the same volume of the 0.05*M* HCl is added. These three scenarios provide a challenging test bed for any control system.

Consider first a situation where a buffer is added to the pH system randomly. The adaptive pH GA-FLC alters the membership functions it uses to enact its production rules (which do not change) although the process dynamics are altered when the buffer is added. This approach is similar to the subconscious actions of a human controller; humans change their definition of the linguistic terms being used in conjunction with their informal rule-of-thumb approach. Figure 9 compares the performance of the adaptive GA-FLC with a nonadaptive FLC that does not employ a GA. The adaptive controller is able to achieve the objective much more efficiently than the nonadaptive FLC because the adaptive controller is flexible enough to accommodate the changing process dynamics.

Next, consider a situation where the setpoint is changed by a random agent. An example of such a change appears in the mineral processing industry, wherein the beaker of the pH system may represent a holding tank in which a

Figure 9



Comparison of controllers when buffer is added. A, The random agent adds acid, base, and buffer; B, the adaptive GA-FLC drives the system pH to the setpoint of seven, and maintains the setpoint better than the nonadaptive FLC.

mineral is being separated. If the mineral of interest is changed (if two different processes occur in the same holding tank because of streamlining of plant operations), the pH of the system may need altering for efficient separation. As in the above examples, the adaptive pH GA-FLC must alter its membership functions in response to an "environmental" change. Realize that declaring a new setpoint is actually changing the objective of the FLC. Changing the objective of the controller often requires a modification of the FLC rule set. However, the technique of using a GA to alter a set of membership functions is powerful enough to allow the FLC to maintain a suitable level of control over the pH system by altering only the meaning of the fuzzy linguistic variables despite the demanding environment in which it must operate. Figure 10 compares the performance of an adaptive GA-FLC with a nonadaptive FLC. As in the previous example, the adaptive pH GA-FLC outperforms the nonadaptive FLC.

Finally, consider a very disruptive change to the pH system, a case where the concentration of the acid and base that the FLC is using to manipulate the pH system is altered. This is perhaps the most severe change in process dynamics that could be implemented. The response of the system is now completely different: additions of acid or base induce changes in the pH of the system that are far different from the changes in pH that the very same additions of acid or base induced before their concentrations were changed. Figure 11 compares the performance of the adaptive pH GA-FLC with the performance of a nonadaptive FLC. The adaptive GA-FLC is able to maintain a high degree of control over the pH system despite the dramatic changes in the environment.

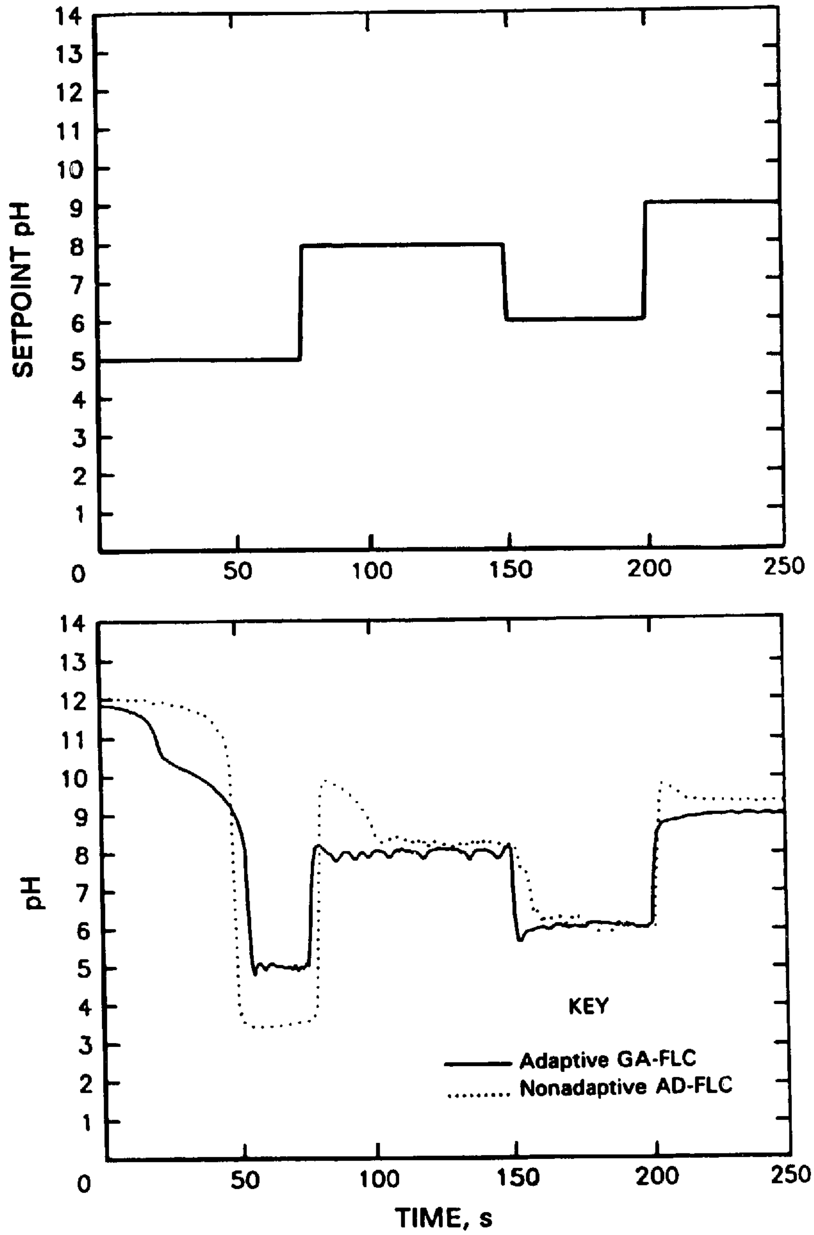
SUMMARY AND CONCLUSIONS

Scientists at the USBM have developed an AI-based strategy for adaptive process control. This strategy uses GA's to fashion three components necessary for a robust, comprehensive adaptive process control system: (1) a control element to manipulate the problem environment, (2) an analysis element to recognize changes in the problem environment, and (3) a learning element to adjust to changes in the problem environment. In this report, the strategy has been applied to the development of an adaptive controller for a laboratory pH system in which the process dynamics change in several different ways. Initially, the overall makeup of an adaptive control system was described. Next, the pH problem environment was introduced. Finally, the basic structure of each of the three individual components was developed, and results were provided demonstrating the merit of using GA's to compose the three components.

The results presented in this report demonstrate much of the power of adaptive control systems based on GA's and FLC's. These adaptive control systems are able to recognize when the physical system has changed, to quantify the changes in the physical system, and to maintain a high degree of control over the physical system despite drastic changes in the system characteristics. Based on the results presented, it is concluded that adaptive GA-FLC's allow industrial pH systems to be controlled via on-line changes to the membership functions used in the rule base associated with the control system.

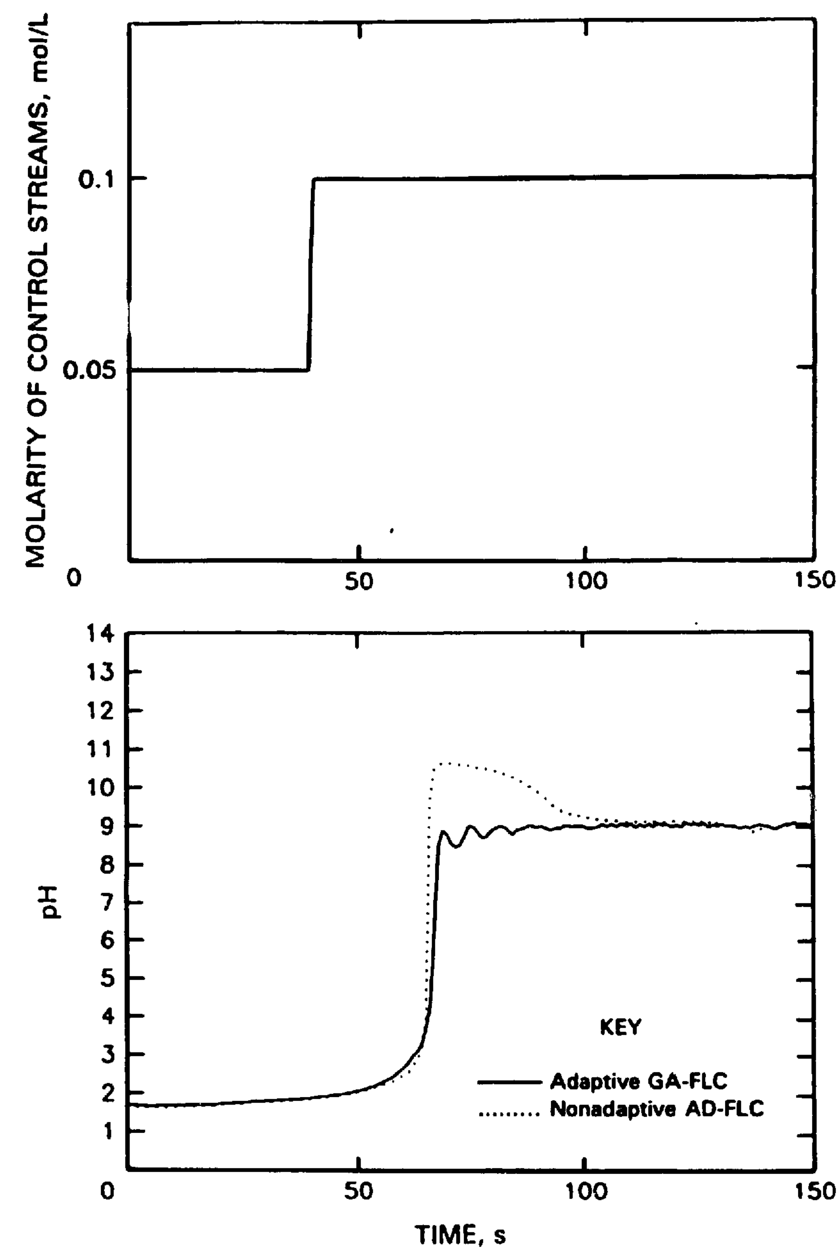
Adaptive control systems are becoming vital to the efficient operation of today's industrial plants because of the rapidly changing process dynamics brought about by increased competition and changing economic factors. If the efficiency of such control systems is going to increase, researchers must focus on the synergism of techniques from various fields of study. In this light, the field of AI contains a vast number of untapped resources. Specifically, GA's and FLC's demonstrate characteristics that allow for the production of control systems that mimic the approach adopted by humans to the task of process control. And, in the final analysis, humans actually perform the task of adaptive control quite well, as does the adaptive control system presented in this report.

Figure 10



Comparison of controllers when setpoint is changed. A, Changes in the system setpoint can create problems for a nonadaptive FLC; B, the adaptive GA-FLC adjusts to changes in the system setpoint more efficiently than a nonadaptive FLC.

Figure 11



Comparison of controllers when acid and base concentration is altered. A, Changes in the concentrations of the control input streams dramatically alter the system dynamics; B, the adaptive GA-FLC reaches the system setpoint faster than a nonadaptive FLC.

REFERENCES

1. Kelly, E. G., and D. J. Spottiswood. Introduction to Mineral Processing. Wiley, 1982, 491 pp.
2. Fogler, H. S. Elements of Chemical Reaction Engineering. Prentice-Hall, 1986, 769 pp.
3. Gottinger, W. W. Economic Models and Applications of Solid Waste Management. Gordon and Breach Sci. Publ., 1991, 119 pp.
4. Coughanowr, D. R., and L. B. Koppel. Process Systems Analysis and Control. McGraw-Hill, 1965, 491 pp.
5. Wang, T. S. Adaptive Multivariable PID Control of a Distillation Column with Unknown Dead Times. Ph.D. Thesis, Univ. AL, Tuscaloosa, AL, 1992, 591 pp.
6. Astrom, K. J., U. Borisson, L. Ljung, and B. Wittenmark. Theory and Applications of Self-Tuning Regulators. Automatica, v. 13, 1977, p. 457.
7. Clarke, D. W., and P. J. Gawthrop. Implementation and Application of Microprocessor-Based Self-Tuners. Automatica, v. 17, 1981, pp. 79-85.
8. Waterman, D. A. A Guide to Expert Systems. Addison-Wesley, 1970, 398 pp.
9. Zadeh, L. A. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. IEEE Trans. on Systems, Man, and Cybernetics, v. SMC-3, No. 1, 1973, pp. 28-44.
10. Evans, G. W., W. Karwowski, and M. R. Wilhelm (eds.). Applications of Fuzzy Set Methodologies in Industrial Engineering. Elsevier, 1989, 335 pp.
11. Turksen, I. B. (ed.). Proceedings of NAFIPS'90 Quarter Century of Fuzzyness. N. Am. Fuzzy Inf. Soc., 1990, 408 pp.
12. Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989, 412 pp.
13. Belew, R. K., and L. B. Booker (eds.). Proceedings of the Fourth International Conference on Genetic Algorithms. Kaufmann Publ., 1991, 576 pp.
14. Krauss, T. Self-Tuning Control: An Expert System Approach. Adv. in Instrum., v. 39, 1984, pp. 695-703.
15. Krauss, T., and T. J. Myron. Self-Tuning PID Controller Uses Pattern Recognition Approach. Control Eng., June 1984, pp. 451-467.
16. Odetayo, M. O., and D. R. McGregor. Genetic Algorithm for Inducing Control Rules for a Dynamic System. Paper in Proceedings of the Third International Conference on Genetic Algorithms, ed. by D. Schaffer (Proc. Conf. George Mason Univ., Fairfax, VA, June 4-7, 1989). Kaufmann Publ., 1989, pp. 177-182.
17. Valenzuela-Rendon, M. The Fuzzy Classifier System: A Classifier System for Continuously Varying Variables. Paper in Proceedings of the Fourth International Conference on Genetic Algorithms, ed. by R. K. Belew and L. B. Booker (Proc. Conf. San Diego State Univ., San Diego, CA, July 13-16, 1991). Kaufmann Publ., 1991, pp. 346-353.
18. Procyk, T. J., and E. H. Mamdani. A Linguistic Self-Organizing Process Controller. Automatica, v. 15, 1978, pp. 15-30.
19. Galluzzo, M., V. Cappellani, and U. Garofalo. Fuzzy Control of pH Using NAL. Int. J. of Approximate Reasoning, v. 5, No. 6, 1991, pp. 505-519.
20. Karr, C. L. Genetic Algorithms for Fuzzy Logic Controllers. AI Expert, v. 6, No. 2, 1991, pp. 26-33.
21. McAvoy, T. J., E. Hsu, and S. Lowenthal. Dynamics of pH in Controlled Stirred Tank Reactor. Ind. Eng. Chem. Process Design Dev., v. 11, No. 1, 1972, pp. 68-70.
22. Larkin, L. I. A Fuzzy Logic Controller for Aircraft Flight Control. Ch. in Industrial Applications of Fuzzy Control, ed. by M. Sugeno. North-Holland, 1985, pp. 87-104.
23. Hand, C. W., and G. L. Blewitt. Acid-Base Chemistry. Macmillan Publ. Co., 1986, 275 pp.
24. Karr, C. L., and E. J. Gentry. Real-Time pH Control Using Fuzzy Logic and Genetic Algorithms. Soc. Min. Eng. AIME preprint 92-49, 1992, 6 pp.
25. Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. Numerical Recipes in C. Cambridge Univ. Press, 1988, 735 pp.
26. Karr, C. L., D. A. Stanley, and B. J. Scheiner. A Genetic Algorithm Applied to Least Squares Curve Fitting. USBM RI 9339, 1991, 15 pp.
27. Kargupta, H., and R. E. Smith. System Identification With Evolving Polynomial Networks. Paper in Proceedings of the Fourth International Conference on Genetic Algorithms, ed. by R. K. Belew and L. B. Booker (Proc. Conf. San Diego State Univ., San Diego, CA, July 13-16, 1991). Kaufmann Publ., 1991, pp. 370-376.

INT.BU.OF MINES,PGH.,PA 30132

☆ USGPO 609-012/21.024