

7916

N94-32422

**A FUZZY CLASSIFIER SYSTEM FOR PROCESS CONTROL**

**C. L. Karr**  
U. S. Bureau of Mines  
Tuscaloosa Research Center  
P.O. Box L, University of Alabama Campus  
Tuscaloosa, AL 35486-9777

52-63  
2485

**J. C. Phillips**  
U. S. Bureau of Mines  
Tuscaloosa Research Center  
P.O. Box L, University of Alabama Campus  
Tuscaloosa, AL 35486-9777

1-10

**ABSTRACT**

A fuzzy classifier system that discovers rules for controlling a mathematical model of a pH titration system has been developed by researchers at the U.S. Bureau of Mines (USBM). Fuzzy classifier systems successfully combine the strengths of learning classifier systems and fuzzy logic controllers. Learning classifier systems resemble familiar production rule-based systems, but they represent their IF-THEN rules by strings of characters rather than in the traditional linguistic terms. Fuzzy logic is a tool that allows for the incorporation of abstract concepts into rule based-systems, thereby allowing the rules to resemble the familiar "rules-of-thumb" commonly used by humans when solving difficult process control and reasoning problems. Like learning classifier systems, fuzzy classifier systems employ a genetic algorithm to explore and sample new rules for manipulating the problem environment. Like fuzzy logic controllers, fuzzy classifier systems encapsulate knowledge in the form of production rules. The results presented in this paper demonstrate the ability of fuzzy classifier systems to generate a fuzzy logic-based process control system.

**INTRODUCTION**

Researchers at the USBM have developed adaptive process control systems that utilize various tools and techniques from the field of artificial intelligence. The most important artificial intelligence tools used in these controllers have been expert systems, fuzzy logic controllers, and genetic algorithms. The quest for innovative, adaptive, and robust process control systems has progressed to the point that a new control system, called a fuzzy classifier system, has been developed that represents a synergism of several artificial intelligence techniques. The fuzzy classifier system was first proposed by Valenzuela-Rendón [1] to map functions of continuous variables. The current paper describes an extension to the work of Valenzuela-Rendón in that the fuzzy classifier system is used to solve a process control problem. The fuzzy classifier system described represents another step toward truly intelligent computer systems that are capable of manipulating complex problem environments because the fuzzy classifier system, in effect, "learns" to manipulate a pH titration problem environment without prior knowledge of an appropriate set of rules. It is worthwhile to note that this step is based on a large amount of prior USBM research.

Initially, USBM researchers developed expert systems for controlling mineral processing systems [2]. These expert systems utilized traditional production rules of the form **IF {condition} THEN {action}**, wherein the *conditions* and *actions* were described using conventional set theory. Next, fuzzy logic controllers were developed that replaced the conventional set theory employed by expert systems with fuzzy logic or approximate reasoning [3]. Fuzzy logic allows for the use of membership functions to describe or define abstract terms akin to those used in a human's "rule-of-thumb" approach to decision making [4], and the fuzzy logic controllers that resulted were more efficient than their expert system counterparts. Then, USBM researchers developed an innovative adaptive process control system in which genetic algorithms were used to tune the membership functions associated with fuzzy logic controllers [5-6]. Genetic algorithms are search algorithms based on the

mechanics of natural genetics, and they rapidly locate near-optimum solutions to difficult search problems [7]. The combination of fuzzy logic controllers with genetic algorithms marked a major step in achieving truly adaptive process control. Recently, fuzzy logic controllers have been combined with learning classifier systems to produce fuzzy classifier systems which require minimal information about the physical system being manipulated.

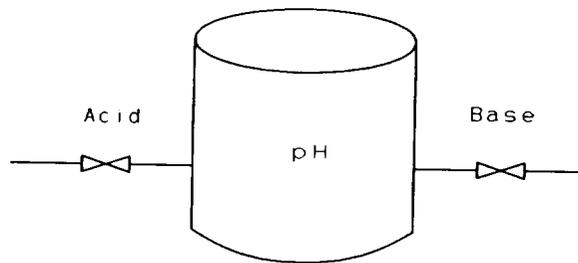
A fuzzy classifier system is a rule-based system that incorporates the "rule-of-thumb" approach used in human decision making with the rule discovery capabilities of learning classifier systems. These innovative systems generate both the rules and membership functions that constitute a fuzzy logic controller. Fuzzy classifier systems consist of three main components: (1) a rule and message system, (2) an apportionment of credit system, and (3) a genetic algorithm. The rule and message system is a mechanism by which the fuzzy classifier system interacts with the problem environment. The fuzzy classifier system receives information concerning the condition of the problem environment and takes an action on the environment based on its rule set. In the apportionment of credit system rules compete for the right to take their action on the problem environment, and are rewarded or punished in accordance with their performance. The genetic algorithm is used as a rule discovery system in which new rules are generated and inserted into the rule store of the fuzzy classifier system. The rewards and penalties accrued via the apportionment of credit system drive the genetic algorithm's search.

In this paper, the fuzzy classifier system developed at the USBM is applied to a specific problem environment: a pH titration system. The fuzzy classifier system developed for a computer simulation of the pH system is quite effective in achieving the process control objective. Its performance is compared to that of a fuzzy logic controller that has been shown to manipulate the pH environment in an effective manner. The performance of the fuzzy classifier system demonstrates the potential of this approach to adaptive process control. Its effectiveness in the highly nonlinear problem of pH control points to far-reaching implications for application in diverse industrial fields such as mineral processing, chemical engineering, and solid waste disposal.

The fuzzy classifier system that results from the synergism of artificial intelligence techniques is important to a number of industries for two reasons: (1) it is simple and (2) it is versatile. First, the use of fuzzy logic greatly simplifies the task of developing a rule-based controller, and rule-based controllers have been successfully implemented in numerous problem domains. Second, the approach to process control is versatile because it relies on genetic algorithms which have been used to efficiently solve a wide spectrum of search problems. Furthermore, the basic design of the control systems that result allow for the effective manipulation of complex problem environments despite the fact that the control systems have not been provided with rules for manipulating the problem environment. This trait is a virtual necessity in numerous industrial settings such as the minerals industry in which the mechanics of the processes in the plants are quite complicated and not understood well enough to write complete rule sets.

## THE PHYSICAL SYSTEM

A simple laboratory pH system is considered to present the details of a fuzzy classifier system. A schematic of this pH system is shown in Figure 1. The system consists of a tank initially containing a given volume of a solution having a known pH. There are two valved input streams into the beaker. The valves on these two *control input streams*, one a strong acid (0.1 M HCl) and one a strong base (0.1 M NaOH), can be adjusted to cause a change in the pH of the solution in the tank. The objective of the control problem is to neutralize the solution — drive the pH to 7 — in the shortest time possible by adjusting the valves on the control input streams. Additionally, the valves on the control input streams are to be fully closed after the solution is neutralized. 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), to restrict pressure transients in the associated pumping systems.



**Figure 1. Schematic of the pH titration system**

The development of a fuzzy classifier system requires a computer model of the problem environment, in this case the pH titration system. A model of the pH system is required due to the way in which a learning portion of the fuzzy classifier system operates. The fuzzy classifier system employs a genetic algorithm which evaluates a number of possible solutions to the problem of locating rules that are appropriate for completing the titration. Some of the possible rule sets the genetic algorithm investigates are totally unacceptable: they represent preposterous control strategies. Therefore, the potential solutions are investigated on a computer simulation of the pH system. Fortunately, the dynamics of the pH system are well understood, and can be modeled for a variety of reactions using conventional techniques [8]. In the pH system considered here, the development of a model of the physical system does not present an insurmountable obstacle. However, it should be realized that for many complex industrial systems, the development of an accurate computer model is an imposing task. In these cases some form of empirical model, such as a statistical or neural network model, offers a suitable alternative to a first principle model. It should also be noted that in situations such as the pH system in which a fundamental model can be produced, conventional control strategies can be quite effective. However, some systems are modeled using statistical approaches (even neural networks and fuzzy rule bases), and it is in these situations that the fuzzy classifier system has the greatest potential.

### LEARNING CLASSIFIER SYSTEMS

A classifier system is a genetics based machine learning system that learns rules, called classifiers, to govern its performance in a given environment e.g., the pH titration system. The systems are based on a model of a service economy in which money is exchanged for services, and include a mechanism for discovering new rules. Unlike in traditional expert systems, a rule's relative value is learned, not fixed by the programmer. In classifier systems the rules are forced to coexist in a service economy where a competition is held to decide which rule will be put into effect under a specified set of conditions in the environment. The competitive nature of the economy ensures good rules survive and bad rules die off, while the exploratory portion of the learning classifier system creates new rules. These systems operate incrementally, testing new rules while steadily improving performance in an environment.

In general, classifier systems are composed of three subsystems:

- 1) Rule and message system;
- 2) Credit assignment system;
- 3) Rule discovery system.

These three subsystems, when combined with a mathematical model of the environment, form a computationally complete system capable of learning effective rules for interacting in an environment and controlling processes. The mathematical model offers an arena in which the fuzzy classifier system can investigate new and improved control strategies. Furthermore, this model of the problem environment can be used to compute changes in the problem environment that can not be measured directly [6].

The rule and message system is similar to a production system. Production systems are schemes that use rules as their only means of operation. The rules are generally of the form:

**IF {condition} THEN {action}.**

When the *condition* exists in the environment, then the *action* is to be taken. In learning classifier systems the rule and message system involves the completion of some basic tasks. The learning classifier system evaluates the existing state of the environment; it determines what conditions exist. These conditions are compared to the current rule set to determine which rules are eligible to be put into effect. A competition is held among the eligible rules and a set of winning rules is selected. These rules take their associated action causing a change in the environment. It should be noted that the first competition is strictly a random decision since all rules initially have the same relative strength. However, as will be seen shortly, all future competitions are decided based upon the previous successes or failures of individual rules. At this point the rule and message system becomes temporarily inactive and the credit assignment system takes over. Before the credit assignment system is discussed, note that the rule and message system is designed to activate several classifiers in parallel which actually characterizes learning classifier systems as parallel production systems even though they are readily implemented on sequential machines.

The purpose of the credit assignment system [9] is to evaluate how useful each classifier has been in producing desirable effects on the environment (how useful it has been in solving the problem at hand). The method used most often is the bucket brigade algorithm of Holland [10]. In the bucket brigade algorithm every classifier is assigned a strength which is a value representing that classifier's usefulness in solving the problem. The classifiers that are eligible to take an action at a given time step bid a portion of their associated strength for the right to take their action. Once all the eligible rules have made their bids, winners are selected through probabilistic means based on the size of the bids; those who bid highest have the highest chance of being selected to take their action. If the environment is in a more desirable state than at the previous time step, the auction winners pay their bids to the classifiers that took action at the previous time step. Thus, the rules that caused desirable changes in the environment are rewarded with payoff. A mandatory payment is also received from every classifier at each time step. This payment, or tax, helps weed out poor rules because in the rule discovery system a classifier's existence depends on its associated strength. If a classifier fails to win the auction and cause a positive change on the environment, its chance of survival is reduced. The bucket brigade algorithm's redistribution of classifier strengths leads to bids that are representative of classifiers' potential for achieving the goal; effective rules bid proportionately higher.

The purpose of the rule discovery system is to generate new rules with potential for more efficiently reaching the goal assigned to the learning classifier system. The genetic algorithm is the technique generally used in rule discovery systems. The genetic algorithm requires the elements of the search space (in this case the rules) to be coded as finite length strings. Effective classifiers, those with high associated strengths, are selected for combination with other highly fit classifiers. Portions of the classifiers are selected at random and combined with portions of other classifiers through the standard genetic algorithm operators of reproduction, crossover, and mutation to produce new rules. The intent is to combine advantageous qualities from two separate classifiers to form two new, more highly fit classifiers.

The three subsystems discussed in this section form a system capable of learning and evaluating new rules for interacting in an environment. A more detailed description of learning classifier systems can be found in Goldberg [7].

## **FUZZY LOGIC CONTROLLERS**

The popularity of fuzzy logic controllers has increased dramatically in the last decade. This increase in popularity has also provided an increase in the number of approaches to implementing fuzzy logic in a process control system [11]. Perhaps the most efficient way to introduce fuzzy logic controllers is to provide

a step-by-step procedure that can be used. In this section, such a procedure is presented and applied to the pH titration system. The fuzzy logic controller that results is later used to evaluate the performance of a fuzzy classifier system.

The first step in developing a pH fuzzy logic controller is to decide on the *condition variables* (these variables appear on the left side of the fuzzy logic controller rules which are of the form: **IF {condition} THEN {action}**). Certainly 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 to a small fundamental set because the size of the rule set increases multiplicatively with the number of condition variables. After a period of experimentation (an inevitable requirement for the development of a quality fuzzy logic controller), two condition variables were selected: the current value of pH (pH) in the beaker and the current time rate of change of the pH in the tank ( $\Delta\text{pH}$ ).

The second step is to determine the specific actions that can be taken on the system, i.e., the *action variables* must be determined. In the pH system, the determination of the action variables is relatively straightforward. There are basically only two action variables that can be altered by the controller: the valve settings (and thus the flow rates) associated with the control input streams. Therefore, the two action variables are the flow rates for the strong acid ( $Q_{\text{ACID}}$ ) and the strong base ( $Q_{\text{BASE}}$ ), respectively, of the input streams. 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.

The third step is to choose linguistic terms that represent each of the condition and action variables. Eight terms were used to describe pH, four terms were used to describe  $\Delta\text{pH}$ , and four terms were used to describe both  $Q_{\text{ACID}}$  and  $Q_{\text{BASE}}$ . The specific linguistic terms used to describe the pertinent variables in the pH system follow:

pH	Very Acidic ( <b>VA</b> ), Acidic ( <b>A</b> ), Mildly Acidic ( <b>MA</b> ), Neutral Acidic ( <b>NA</b> ), Neutral Basic ( <b>NB</b> ), Mildly Basic ( <b>MB</b> ), Basic ( <b>B</b> ), and Very Basic ( <b>VB</b> );
$\Delta\text{pH}$	Negative Large ( <b>NL</b> ), Negative Small ( <b>NS</b> ), Positive Small ( <b>PS</b> ) and Positive Large ( <b>PL</b> );
$Q_{\text{ACID}}$	Zero ( <b>Z</b> ), Small ( <b>S</b> ), Medium ( <b>M</b> ), and Large ( <b>L</b> );
$Q_{\text{BASE}}$	Zero ( <b>Z</b> ), Small ( <b>S</b> ), Medium ( <b>M</b> ), and Large ( <b>L</b> ).

All of these linguistic terms are subjective, i.e., the terms can mean different things to different people, but the developers (the authors) of the pH fuzzy logic controller have some meaning they associate with each of the terms.

The fourth step is to provide the selected linguistic terms with some concrete, or crisp meaning. The linguistic terms are "defined" by membership functions. As with the requirement for selecting the necessary linguistic terms, there are no definite guidelines for constructing the membership functions; the terms are defined to represent the designers' general understanding of what the terms mean.

The fifth step in the design of a fuzzy logic controller is the development of a rule set. The rule set in a fuzzy logic controller must include a rule for every possible combination of the controlled variables as they are described by the chosen linguistic terms. Thus, the pH fuzzy logic controller, as described to this point, will contain  $8 * 4 = 32$  rules to describe all of the possible conditions that could exist in the pH system as described by the linguistic terms represented by the membership functions. For any combination of the condition variables, an appropriate choice of the action variables is prescribed. Due to the nature of the linguistic terms, most of the actions needed for the 32 possible condition combinations are readily apparent. For instance, when pH is **VA** and  $\Delta\text{pH}$  is **NS**, then  $Q_{\text{ACID}}$  should be **Z** and  $Q_{\text{BASE}}$  should be **L**. However, there are some conditions for which the appropriate action is not readily apparent. In these instances, some experimentation is often needed.

Now that both the condition and action variables have been chosen and described with linguistic terms, and a rule set has been written that prescribes an appropriate action for every possible set of conditions, it is left to determine a single crisp value of the acid and base valve settings. This may or may not be viewed as a step in the fuzzy logic controller development. Certainly, there are numerous approaches to the fuzzy computations performed by the fuzzy logic controller. The procedure for determining a single crisp value of the valve settings for the acid and base input streams is a concern because, unlike in traditional expert systems, more than one of the fuzzy logic controller's 32 rules can be applicable for a given state of the pH system. A common technique for accomplishing this task is the center of area method (sometimes called the centroid method). In the center of area method, the action prescribed by each rule plays a part in the final crisp value of the valve settings. The contribution of each rule to the final value of  $Q_{ACID}$  and  $Q_{BASE}$  is proportional to the minimum confidence (the minimum value of the membership function values on the left side of the rule) one has in that rule for the specific state of the physical system at the particular time. This is equivalent to taking a weighted average of the prescribed actions. Further explanation of the center of area method is provided in Sugeno [11].

One detail specific to the pH system should be considered here. 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. However, the membership functions used in the center of area method (shown in Figure 2) allow for values of  $Q_{ACID}$  and  $Q_{BASE}$  to range between 0.0 mL/s and 2.5 mL/s, irrespective of their current values. The constraint is imposed by computing the value of the flow rates using the center of area method. If this value exceeds the constrained flow rate (for a time period between steps of 1 s), the flow rate is changed by the maximum allowable value of 0.5 mL/s (for either increases or decreases in flow rate). With the determination of a strategy for resolving "conflicts" in the actions prescribed by the individual rules, the fuzzy logic controller is complete.

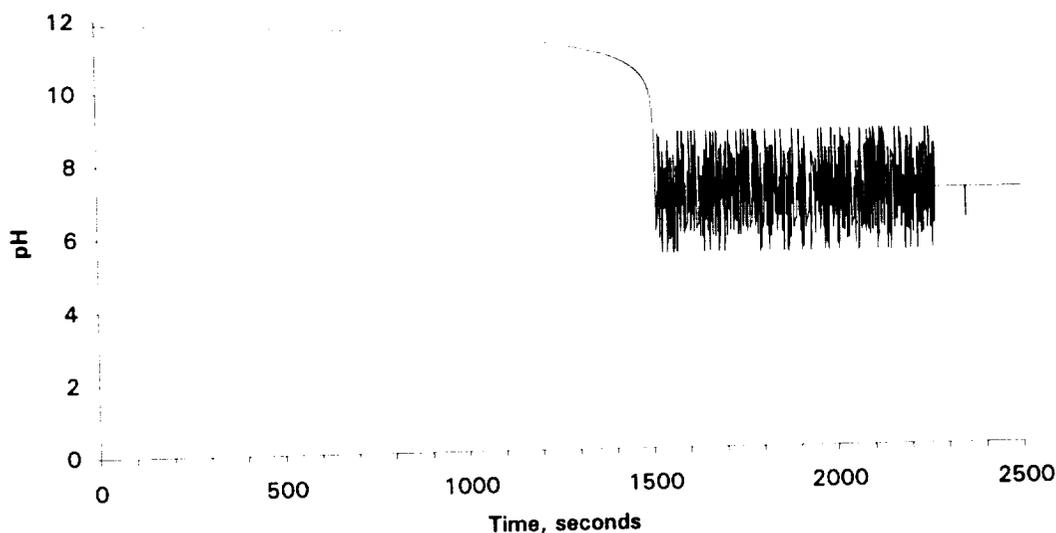


Figure 2. The fuzzy classifier system neutralizes a basic solution

### APPLICATION OF A FUZZY CLASSIFIER SYSTEM

The first step in producing a fuzzy classifier system for the pH titration system is to develop a means of representing linguistic rules and fuzzy membership functions as strings of characters. This step is required so that a genetic algorithm may be employed in the rule discovery system. An approach to mapping fuzzy membership functions has been well documented [5] and is used here to form a portion of the character strings. The remainder of the strings represent the rule set used in the controller. For this study a direct mapping was used to represent the rule sets. The condition portion of each classifier is a function of the current pH and the current

time rate of change of pH. The action portion of each classifier involves adjusting the net inflows ( $Q_{ACID}$  and  $Q_{BASE}$ ) at the following time step. The coding used for the rule set is summarized in Table 1. It is important to note that Table 1 includes only the coding used for the rules. The character strings used contained information concerning both the rules and the membership functions and were 69 bits long.

Table 1. - The coding used to represent the rule sets.

Condition				Action			
pH		$\Delta$ pH		$Q_{ACID}$		$Q_{BASE}$	
binary	set	binary	set	binary	set	binary	set
000	VA	00	NL	00	Z	00	Z
001	A	01	NS	01	S	01	S
010	MA	10	PS	10	M	10	M
011	NA	11	PL	11	L	11	L
100	NB						
101	MB						
110	B						
111	VB						

The basic binary coding in which each bit position is characterized by either a 1 or a 0 would be sufficient for the learning classifier system. However, to add flexibility and allow for more general rule representation, a third character is introduced, namely the # or "don't care" symbol. A bit position filled by a # is matched by either a 1 or a 0. An example rule follows:

condition → action  
00#00 → 0011.

In this rule, the value of pH at the current time is defined by the bits 00#. This condition is satisfied by either 000 or 001. Thus, this rule says that if the pH is currently very acidic or acidic with a negative large time rate of change of pH, then minimize the flow of acid into the tank and maximize the flow of base into the tank at the next time.

Now that a means for representing the rules and fuzzy membership functions has been established, the implementation of a fuzzy classifier system is discussed. A rule set of 250 rules was used to control a mathematical model of the pH titration system. These rules were used to manipulate the computer model of the pH titration system until 25,000 rules had been enacted. At the end of this period, 120 new rules were produced using a genetic algorithm. These 120 rules replaced 120 of the 200 poorest rules in the previous rule set. By crowding out the poor rules and relying on the genetic algorithm's survival-of-the-fittest approach, the rule sets achieved higher levels of performance. A learning cycle consisted of the 25,000 actions or one generation of new rules using the genetic algorithm. However, learning actually took place after each individual rule activation at which time the bucket brigade algorithm redistributed payment to the rules. The steps composing these learning-cycles are relatively straightforward and consist of the following:

- 1) Evaluate the condition or state of the physical system at the current time step as predicted by the model.
- 2) Compare the state of the physical system to the condition portion of each of the 250 classifiers to determine which rules are eligible to take their action.

- 3) Hold an auction between the eligible rules to determine which rules get to take their actions. The bid placed by a given rule is defined by

$$BID = S * C_{bid} \quad (1)$$

where BID is the rule's bid, S is the rule's strength, and C bid is a constant. In this work, C is set to a value of 0.1 which places an appropriate emphasis on the bid. Within the model of a service economy, strength becomes a measure of how effective a rule is at driving the system to its setpoint.

- 4) Evaluate the effectiveness of the action. If the action, which is a function of both the rules and the fuzzy membership functions, drives the system closer to its setpoint, then the rules that won the auction at the previous time step are rewarded with the current winners' bids and a tax paid by each classifier as described below. Otherwise no rule receives a reward and the tax is accumulated in a fund and carried over to the next time step.

- 5) Tax each of the rules in the current rule set. Every rule is taxed according to the following:

$$TAX = (C_{bidtax} - C_{lifetax} S) \quad (2)$$

where TAX is the tax paid by a classifier,  $C_{bidtax}$  is a constant (equal to  $5 \cdot 10^{-5}$  in this study if the rule's condition is met and thus it placed a bid; it is equal to 0.0 otherwise), and  $C_{lifetax}$  is a constant (equal to  $2 \cdot 10^{-5}$  in this study under every circumstance).

- 6) After 25,000 actions are taken, apply the genetic algorithm to generate a new rule set which includes 120 new rules. The genetic algorithm combines portions of the most effective rules as determined by their strength, which rises if a rule is effective and falls if a rule is non-effective due to the nature of the apportionment of credit algorithm.

Although the mechanics are relatively simple, the fuzzy classifier system is able to employ the exploratory power of the genetic algorithm and the structured credit assignment of the bucket brigade to learn new and improved rules for controlling pH titration system.

## RESULTS

Evaluating the effectiveness of a process control system is not a trivial endeavor; there are various criteria for efficient control that can be established, and there are numerous conditions over which the controller must be able to perform. In evaluating the performance of the fuzzy classifier system it is important to keep in mind the control objective which is to neutralize the solution as fast as possible while not violating the constraints placed on the flow rates. Additionally, it is important to realize that if the controller can accomplish the control goal from extreme portions of the control space (when the solution is initially extremely acidic or extremely basic), it should perform well in the more moderate portions of the control space. Thus, the performance of the fuzzy classifier system is compared to the performance of a fuzzy logic controller that was developed by the authors for the pH titration system.

Figure 2 summarizes the performance of the fuzzy classifier system. In the particular case depicted, the fuzzy classifier system was posed with the problem of neutralizing a basic solution. As can be seen in the figure, the fuzzy classifier system rapidly locates rules for driving the pH to values that range roughly between 5.2 and 8.5. Then, after a brief period of exploration, the controller is able to evolve into a form that forces the pH to the desired value of 7. During the period of exploration, the fuzzy classifier is considering rules selected by a genetic algorithm; some are good and some are bad. The result appears to be a random walk through the search space. However, the genetic algorithm is actually updating its selections based on the performance of the rules. Figure 3 shows the performance of a fuzzy logic controller that was designed by the authors to solve the pH problem. Note that this controller drives the pH to a value that is between 6.5 and 7.5. It is important to note that the

performance of the author-developed fuzzy logic controller can be improved by altering the membership functions [6], but this is a time-consuming task to accomplish without the assistance of some computational algorithm such as a genetic algorithm. Nonetheless, the two figures demonstrate the fact that the fuzzy classifier system contains a mechanism for improving its performance through the discovery aspects of a genetic algorithm.

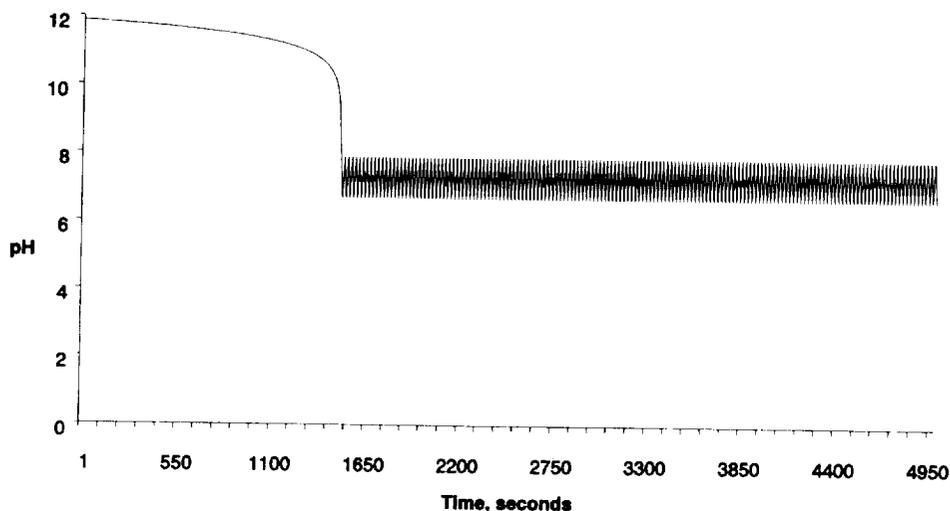


Figure 3. A fuzzy logic controller neutralizes a basic solution

Figures 2 and 3 demonstrate the effectiveness of the fuzzy classifier system in the pH problem environment. Although the results presented do not alleviate concerns as to the ability of the fuzzy classifier system to perform effectively in process control problems, they do point to the potential of these fuzzy rule discovery systems. These results demonstrate the ability of the system to discover both rules and fuzzy membership functions for effectively manipulating a highly nonlinear physical environment.

#### FUTURE WORK

The previous section provided results that indicate a fuzzy classifier system is capable of efficiently controlling a pH titration system. However, there are still a number of research issues left open. The following amplifications are currently under investigation:

- The pH titration system is being extended to include external perturbations such as additions of a buffering solution. This extension will make the pH titration system more like chemical systems currently used in industry, and will force the fuzzy classifier system to discover rules in real time more often than it has to with the current physical system.
- A general purpose credit assignment system is being developed. Currently, the credit assignment system is specialized for the pH titration system. For the fuzzy classifier system to be easily applied to alternative process control problems, a general purpose credit assignment algorithm must be developed.
- The fuzzy logic control aspect of the fuzzy classifier system is being improved. The center of area algorithm for selecting a single crisp action is but one of a number of potential solution algorithms that have been proven effective in fuzzy logic controllers. Current efforts are centered on including fuzzy singletons [11] into the fuzzy classifier system. Successful implementation of this algorithm should improve the effectiveness of the fuzzy actions prescribed by the rule sets.

- Alternative schemes for encoding the classifiers are being considered. The effectiveness of the genetic algorithm's search is highly dependant on the coding scheme used. Although the current coding seems to be effective, recent research efforts [5] indicate that there may be more efficient ways to encode the information concerning the rules and the fuzzy membership functions.
- A micro genetic algorithm is being incorporated into the fuzzy classifier system. The micro genetic algorithm is a small population genetic algorithm that has been reported to perform well across a spectrum of search problems, and in the problem of locating fuzzy membership functions in particular [6]. If this effort is successful, the fuzzy classifier system should be able to locate effective rules in less time than it currently takes.

## REFERENCES

1. Valenzuela-Rendón, M. (1991). "The Fuzzy Classifier System: A Classifier System for Continuously Varying Variables." In K. Belew and L. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, 346-353.
2. Davis, B. E., Jordan, C. E., and Stanley, D. A. (1990). "Expert Advisor for Phosphate Flotation." *Control '90 - Society for Mining, Metallurgy, and Exploration*, 77-85.
3. Karr, C. L., Freeman, L. M., & Meredith, D. L. (1990). "Genetic Algorithm Based Fuzzy Control of Spacecraft Autonomous Rendezvous." *Proceedings of Fifth Annual Conference on Artificial Intelligence for Space Applications*, NASA CP-3073, 43-51.
4. Zadeh, L. A. (1973). "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes." *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, 28-44.
5. Karr, C. L. (1991). "Genetic Algorithms for Fuzzy Controllers." *AI Expert*, 6(2), 26-33.
6. Karr, C. L., and Gentry, E. J. (1993). "Fuzzy Control of pH Using Genetic Algorithms." *IEEE Proceedings on Fuzzy Systems*, 1(1), 46-53.
7. Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.
8. Hand, C. W., and Blewitt, G. L. (1986). *Acid-Base Chemistry*. Macmillan Publishing Company, New York, NY.
9. Minsky, M. L. (1967). *Computation: Finite and Infinite Machines*, Prentice Hall, Englewood Cliffs.
10. Holland, J. H. (1962). "Outline for a Logical Theory of Adaptive Systems." *Journal of the Association of Computing Machinery*, 3, 297-314.
11. Sugeno, M. (Ed.). (1985). *Industrial Applications of Fuzzy Control*. Elsevier Science Publishers, Amsterdam.