EFFECT OF VARYING CONTROLLER PARAMETERS ON THE PERFORMANCE OF A FUZZY LOGIC CONTROL SYSTEM

Paul N. Ekemezie and Charles C. Osuagwu
Department of Electronic Engineering
University of Nigeria, Nsukka.

Abstract

This paper presents the results of computer simulation studies designed to isolate the effects of the major parameters of a fuzzy logic controller namely the range of the universe of discourse, the extent of overlap of the fuzzy sets, the rules in the rule base and the modes of the output fuzzy sets on the performance of a fuzzy logic control system. The controlled process was modeled by a nonlinear differential equation that was solved using the Runge-Kutta numerical method. The results show that varying the range of the universe of discourse of the inputs to the fuzzy controller affects both the transient response and the steady state error of the system, and that a desired system response could be achieved by adjusting the modes of the output fuzzy sets given a fairly good rule base. It has also been shown that the system response could be fine-tuned by varying the overlap of the input fuzzy sets.

1. Introduction

Conventional control system design requires the existence of a precise mathematical model of the process. For this reason, process dynamics are represented as integro-differential equations. The desired response characteristics of the control system are also expressed mathematically. Controller design then involves the use of all this mathematical information to deduce a mathematical expression that describes the action of the controller. Usually, this mathematical expression that describes the controller appears as a function relating plant states or outputs to the control action to be effected at the input to the plant [1].

A fuzzy controller approaches a control problem the way a human operator would. Instead of focusing on the mathematical modelling of the plant, as the fundamental starting point, the human operator's behaviour is given the primary attention. The adjustments of the control variable are handled by a fuzzy rule-based expert system, which is a logical model of the thinking processes a human operator might go through in the course of manipulating the system [2]. Fuzzy control system design does not depend on a mathematical model of the process. A fuzzy controller is based on an approximate representation of the control output and its behaviour as the plant states or outputs deviate from or correspond with desired or setpoint values [3]. This shift in focus from the process to a human operator changes the entire approach to automatic control problems.

E. H. Mamdani introduced the concept of fuzzy logic control (FLC) in 1974 [4]. FLC was strongly motivated by the theory of fuzzy sets developed by L. A. Zadeh [5]. Since then FLC based systems have proven to be superior in performance to conventional systems in a number of areas. Their superiority are particularly pronounced in those areas where mathematical modelling is rather difficult, that is, systems that are ill-defined.

In this paper we report the results of efforts we made to discover the influences of the various design parameters on the performance of fuzzy logic controllers. The work was done with the hope that this discovery will make it easier to design FLCs. We present the operating principles of a fuzzy logic controller in § 2. § 3

describes a series of computer simulation experiments that investigated the effects of variations of some parameters of a fuzzy reasoner, namely the range of the universe of discourse, the extent of overlap of the fuzzy sets, the rules in the rule base and the modes of the output fuzzy sets, on the behaviour of a fuzzy logic control system. § 4 presents the simulation results. § 5 discusses the results, and the paper ends with concluding remarks.

2. Principles of operation

A fuzzy controller is an approximate reasoner in which a control output is deduced by manipulating the current controller input with a collection of linguistic rules using some established inference procedures [6]. Figure 1 shows the block diagram of a control system that makes use of fuzzy logic control. Within one sampling period, the fuzzy controller must do the following: convert the measured error signals into the input signals required by the fuzzy subsystem, this function is performed by the Input Formation block; create a fuzzy set representation of the input signals, this function is performed by the Fuzztfication block; deduce the current controller fuzzy outputs by means of linguistic rules, the Inference Engine block performs this function by operating on the Rule- base; and finally obtain crisp controller outputs by operating on the inferred fuzzy outputs, this task is carried out by the Defuzzification block. The resulting control output is then sent to the process being controlled.

2.1. Input formation

In figure 1 the process states or outputs are compared with set point values, forming error signals. The function of the Input Formation block is to convert an m-dimensional vector that represents the error signals into a 2m-dimensional vector input to the fuzzy subsystem. Each error signal is extended into two elements: e and c, where e is the error signal itself and c is the derivative of error given by

$$c = \frac{de}{dt} \qquad (1)$$

In digital implementation of the controller, there is no need to implement the discrete equivalent of Eq. (1), instead the sample-to-sample change in error is used. If the sampling period is T, then at time kT the change in error is given by

$$c(KT) = e(KT) - e((k-1)T) \qquad (2)$$

2.2. Fuzzification

The function of the Fuzzification block is to normalize the inputs and convert to fuzzy representations. The control vector u must be transferred into a new vector U_0 by multiplication with suitable scaling factors g_i, and then quantizing the scaled values to the closest elements of the corresponding universes, i.e. u_0, = [U_{0l}, U_{02},…..u_{0n}] with U_{0i} = [g_i,U_i] and U_{0i} ε U_i; The settings of the gains g_i, is important in determining the response of the controller.

Each input to and output from the fuzzy subsystem must be decomposed into a set of fuzzy regions called the term set [7]. The fuzzy sets overlap, so that there are inputs which have non-zero grades of membership in more than one set. The lack of mutual exclusivity is important in determining the recommended control action [2].

2.3. Rulebase

The rulebase is the knowledge repository for the fuzzy expert system, where the linguistic rules are stored. Assume that there are p rules in the rule base. The jth rule has the form

$$IF u_1 is A_1^j AND\ u_2 is A_i^j \ldots\ldots AND\ u_m is A_n^j$$
$$THEN\ V_1 is\ B_1^j AND\ V_2\ is\ B_2^j \ldots. AND\ v_n is\ B_n^j \qquad (3)$$

Where u_i, and V_k are linguistic variables

corresponding to the input and output variables of the fuzzy reasoner; A_i and B_k, are fuzzy subsets representing some linguistic terms such as positive. big and negative medium, etc., and will be defined numerically in the respective universes of discourse by membership functions. Equation (3) shows that the premise of a rule refers to the inputs (which represents the condition of the controlled process), while the conclusion or action refers to the controller outputs.

2.4. Inference engine

The Inference block converts the input fuzzy sets into output fuzzy sets. The inference engine uses the fuzzified inputs to execute the rules in the rule base. Due to the fact that a particular input value may fall into more than one fuzzy set, more than one rule may contribute to a particular control action. All the rules are checked to see if they match the labels of the resulting input fuzzy sets.

For purposes of illustrating the inference procedure, consider the following rule.

[R_j] IF temperature error is *negative large* AND change in temperature error is zero THE flowrate' *is positive big*

Fuzzy controllers use standard fuzzy inferencing techniques called correlation minimum and the min/max inferencing method. However, some researchers (such as Kosko [8]) believe that the correlation product and the fuzzy additive inferencing method are superior, with some experiments indicating that these methods yield better results. Here is the procedure for the standard inferencing technique [7]:

(1) For all the predicate expressions connected by an AND take the minimum of their collective membership truths. This final truth is the truth of the rule premise.

$$P_{truth} = \min(E_1, E_2, E_3, \ldots . E) \quad (4)$$

In the previous rule [R_j] this means evaluating the fuzzy propositions (temperature error is positive large) and (change in temperature error is zero) to find their degrees of truth. The minimum of the two truth values forms the truth of the predicate.

(2) The fuzzy set on the right-hand side, i.e., the consequent or action (the "control" fuzzy set) is then reduced in height by this amount.

$$\mu_{control}(\times) = \min(\mu_{control}(\times), P_{truth}) \quad (5)$$

In [R_j] the control fuzzy set is *positive* big. The height of *positive big* is then reduced to the truth value of the predicate. This is the correlation minimum process.

(3) This newly modified fuzzy set (positive correlated to the truth value of the predicate) is then copied into the output variable's fuzzy set. If that region is *not* empty, then it is OR'd with the current contents by taking the maximum of the new fuzzy region and the currently existing fuzzy region at each point along the domain (the horizontal axis).

$$\mu_{solution}(\times) = \max(\mu_{solution}(\times), \mu_{control}(\times)) \quad (6)$$

2.5. Defuzzification

The outcome of the inference process so far is a fuzzy set, specifying a fuzzy distribution of control action. However, a single action only may be applied, so a single point need to selected from the set. This process of reducing a fuzzy set to a single point is known as defuzzification [9]. Most current FLCs employ the centroid or *centre of gravity* method of defuzzification. For the kth output fuzzy set, the general formula for centroid is [10]

$$v_k = \frac{\int B_K^*(v) v dv}{\int B_k^*(v) dv} \quad (7)$$

where v refers to a particular control output and ranges over the appropriate domain and $B_K(V)$ is the membership function of the corresponding fuzzy region. In digital implementations. discrete integration would be performed, i.e., for all $v \in B$ [10],

$$v = \frac{\sum \mu_B(v)v}{\sum \mu_{B(V)}} \quad (8)$$

3. Simulation studies

The objective of the simulation was to demonstrate the design process of a fuzzy logic controller and to observe the influence of its major components, particularly the fuzzy sets and the rulebase on the performance. The controlled variable of the plant is assumed to obey a nonlinear differential equation that is now used in the simulation.

The differential equation used in the simulation is:

$$\frac{d^2y(t)}{dt^2} = 1.6\frac{dy}{dt} - 0.1y(t)^2 - u(t) \quad (9)$$

where y(t) is the controlled variable and u(t) is the control signal. A computer program module was developed which solves this differential equation model of the process by means of the Runge-Kutta-Nystrorn numerical technique, with the assumption of zero initial conditions. The simulation program basically implements unit step response test on the control system.

As explained in section 2, the control error is extended into two signals, *error and* change-in- error, by the input formation block. These two signals serve as the inputs to the fuzzy controller. Figure 2 shows the fuzzy sets that were used to represent the two inputs, error and change-in-error, and one output, the control signal u(t). Each input or output variable was decomposed into seven fuzzy regions, large negative (LN) , medium negative (MN), small negative (SN), zero (ZR), small positive (SP), medium positive (MP) and large positive (LP).

Having specified the fuzzy sets, the next task in the design process is to elucidate the rulebase. Since there are only two inputs, each of which can fall into any of seven fuzzy regions, writing the rules simply involves deciding what the output fuzzy set should be for each possible input combination. From the interaction of the two inputs, a seven-by-seven matrix can be constructed showing the output for each input combination (figure 3). Such a matrix is called a fuzzy associative memory (F AM), as it associates the input of a fuzzy controller with the desired output action **[3].** The FAM is the rule base for the fuzzy controller.

The final step in the design process is the selection of a method of defuzzification. In the

$$(distance_i < \alpha) \rightarrow (\check{x} \epsilon A_i) \quad (11)$$

present case, the centroid method (Eq. 9), which is the most popular in FLC, is employed.

3.1. Programming considerations

Every block in figure 1 was implemented in software using the C++ programming language. In software, each of the universes of discourse represented by figure 2, that is, *error, change-in- error*, and *output* [u(t)], is decomposed into seven triangular fuzzy sets, each of which is identified by its modal value m_i. All the fuzzy sets in each universe of discourse have the same span a Figure 4 illustrates these ideas. Let the input and output variables be decomposed into fuzzy sets whose modal values and span are given as follows:

$$\text{error} = \{-\text{I}, -2/3, -1/3, 0, 1/3, 2/3, \text{I}\}$$

$$error\_span = 1/3$$

$$change - in - error = \{-1/2, -1/3, -1/6, 0, 1/6, 1/3, 1/2\}$$

$$error\_change\_span = 1/6$$

$$output = \{-2, -4/3, -2/3, 0, 2/3, 4/3, 2\}$$

$$output\_span = 2/3$$

Each universe of discourse is quantized into 101 values, and is associated with seven arrays, corresponding with the seven fuzzy regions. This can be stored in memory as a 7 x 101 2- dimensional array. For example, the quantized values for *error* would be -1, -.98, -.96,…, -.02, 0, .02, …,.98, 1. The elements of the array are the corresponding membership grades of the quantized values in each fuzzy set (each row of the array corresponds to a

fuzzy set, each column identifies with a quantized value). The elements of the array are computed and stored during the initialization stage of the computer program. The determination of the membership value of a quantized value *x* in fuzzy set i i = 1 2, ... , 7, begins with the computation of the distance of *x* from the modal values. m_ii=1,2,…,7, of the fuzzy sets in the corresponding universe of discourse. The distance is computed as follows:

$$distance_i = |m_i - x| \qquad (10)$$

where Ai is fuzzy set i in the universe of discourse. If the left hand side of (12) is true, meaning that *x* falls within fuzzy set i, the membership value is computed as follows:

$$\mu_{Ai}(x) = \frac{distance_i}{\alpha} \qquad (12)$$

If the left hand side of (12) is false

$$\mu_{Ai}(\times) = 0$$

Having stored the vocabulary fuzzy sets (or term set) in memory, let us show how fuzzification is then carried out. Consider that at a sampling instant *error* = 0.2345. By means of rounding, this value should be quantized to *error* = 0.24. The column position of 0.24 in the array is given by

$$1 + \frac{0.24 - (-1)}{0.02} = 63$$

Generally, the column position is given by

$$column\_position = 1 + \frac{q\_value - U_{min}}{quanta} \qquad (13)$$

Where q_ value is the quantized crisp value, U_{min} is the minimum value in the universe of discourse, and quanta is the value of the quantization step (0.02 in the present case).

Once the column position has been determined, the membership value of the quantized crisp value in each of the seven fuzzy sets can be read off the array by moving vertically downwards to the appropriate rows. This is fuzzification.

In the simulation software, the F AM of figure 3 is stored as a 7 x 7 matrix. The fuzzy sets NL, NM, etc., are represented by the numbers 1, 2, ... , 7, coresponding to the rows, columns and entries of the F AM. Essentially, the F AM of figure 3 is represented as follows:

{1, 1, 1, 2, 2, 3, 3},
{1, l, 2, 2, 3, 3, 4}
{l, 2, 3, 3, 4, 4, 5},
{2, 3, 4, 4, 5, 5, 6},
{2, 3, 4, 5, 5, 6, 7},
{3, 4, 5, 5, 6, 7, 7},
{4, 5, 5, 6, 7, 7, 7}

The inference engine should fire the rules in the rule base in accordance with the result of fuzzification. The rows of the fuzzification array that do not have all zeros correspond to the fuzzy sets within which the current input value falls-there cannot be more than two such rows in the present case, in line with figure 2. Fuzzification of *error* will yield one or two fuzzy sets and fuzzification of *change-in-error* will yield one or two fuzzy sets. Every fuzzy set for *error* must combine with every fuzzy set for *change-in-error* in executing the rule base (FAM). The fuzzy set(s) for *error* operate the rows of the FAM, while the fuzzy set(s) for *change-in-error* operate the columns. This way the implied output fuzzy sets can easily be read off the F AM. Apart from reading the implied output fuzzy set off the F AM, the inference engine also has to determine the predicate truth value according to (5). Then the maximum truth value for the implied output fuzzy set must be set to the predicate truth value, according to (6).

When the inference engine begins to scan the rule base, the maximum truth value for each output fuzzy set is set to zero; this is equivalent to saying that the solution fuzzy region is made empty. As each rule is fired, the maximum truth value of the implied fuzzy set is updated and the modified fuzzy set is copied into the solution fuzzy region. Equation (7) is employed in case this copying might interfere

with what has already been copied into the solution fuzzy region. In the end the result of the inference procedure is the fully constructed solution fuzzy region.

It now remains to defuzzify the solution fuzzy set, i.e., defuzzification. Since the solution fuzzy set exists as a single 1 x 101 array, it is very easy to implement Eq. (9) in software. Each element of the array, as already stated, is the membership value μB, and the associated domain value v is implied by the position of the element in the array [Eq. (14)].

The result of defuzzification serves as the input to the differential equation model of the process. The differentail equation is solved numerically using the Runge-Kutta method, to yield the output of the controlled process. The process output is compared with the setpoint (= 1) to form the error signal which is then sent to the Input Formation module. The Input Formation module computes the *error* and *change-in-error* signals that serve as inputs to the fuzzifier and the simulation cycle continues.

3.2. Effects of fuzzy sets

We studied the influences of some parameters of the input and output fuzzy sets on the performance of the controller. To this end, the fuzzy sets defined in section 3. I were regarded as standard. They were used as the term set in a simulation run. Refer to this experiment as experiment 1.

The range for fuzzy set *error* was then changed to [-1.5, + 1.5] and another simulation run was carried out as experiment 2. The range for fuzzy set *error* was further changed to [-0.66, +0.66] and once again another simulation run was carried out as experiment 3. At this point, the standard *error* fuzzy set was restored, and the span was reduced to 0.25. Another simulation run was carried out as experiment 4.

Now, turning attention to the fuzzy set for *change-in-error*, the standard term set was restored and the range for *change-in-error* was changed to [-1, +1] and then to [-0.3, +0.3], giving rise to experiments 5 and 6. Then the standard term set was restored and the span for *change-in-error* was reduced to 0.12 resulting in experiment 7.

The standard term set was restored, and the range for *output* fuzzy set was changed to [-2.7, +2.7] and then to [-1.5, + 1.5], giving rise to experiments 8 and 9. Then the standard term set was restored and the span for *output* was reduced to 1/3 giving rise to experiment 10. In all these experiments, the same rule base was used.

3.3. Influence of the rule base

Further studies were carried out to determine the effects of the rule base on the performance of the fuzzy controller. To this end, the FAM of figure 3a was used in conjunction with the standard term set to carry out a simulation run. Further, the F AM of figures 3 band 3c were also tried. These gave rise to experiments 11, 12, and 13.

3.4. Manipulations of the output fuzzy set

The *output* fuzzy set was manipulated severally, in attempts to improve the performance of the controller. In this report, we will show the results for the following three configurations:

$$output = \{-2, -4/3, -2/3, 0.05, 2/3, 4/3, 2\}$$
$$outputspan = 2/3$$
$$output = \{-2, -4/3, -0.2222, 0.45, 1.2, 1.7333, 2\}$$
$$outputspan = 2/3$$
$$output = \{-2, -4/3, -0.2222, 0.42, 1.2, 1.7333, 2\}$$
$$output - span = 2/3$$

These gave rise to experiments 14, 15, and 16.

4. Results and Discussion

In presenting the results and their discussion, it should be recalled that the simulation program

implements unit step response test. The criteria for comparing the performances of the controller then is how quickly the process output settles to unity.

Figure 5 shows the plots of the results of experiments 1 to 4. As stated in the previous section, experiment 1 is the standard, that is, no adjustment has been made to the parameters of the fuzzy controller. The result of experiment 2 indicates that expanding the range of the universe of discourse for *error* is equivalent to reducing the controller gain, so that the speed of the system (in reaching the setpoint value) is greatly reduced and there is highly increased steady state error. The result of experiment 3 shows that reducing the range of the universe of discourse for *error* makes the system to be faster and oscillatory, and there is some improvement in steady state error. This means that reducing the range of the universe of discourse increases the controller gain. The result of experiment 4 indicates that narrowing the domains of the *error* fuzzy sets slightly increases the speed of the system, there is no visible improvement in steady state error, and the system response curve is not smooth. Clearly, then, the overlap in the fuzzy sets creates smooth response curve.

Figure 6 shows the plots of the results of experiments 5, 6, and 7. The plot for experiment 5 indicates a system with faster transient response, which corresponds with the effect of reduction of the derivative term in a proportional + derivative (PD) controller. The plot for experiment 6 shows that reducing the range of *change-in-error* leads to a system that is slow and with large steady-state error, which corresponds with increasing the derivative term in a PD controller. The plot for experiment 7 shows that whereas narrowing the *change-in-error* fuzzy sets does not affect the system transient response, it could lead to oscillatory steady state response. This result, apparently, is a consequence of the fact that reducing the overlap of the fuzzy sets reduces the smoothness of the controller response.

Figure 7 shows the plots for experiments 8, 9, and 10. The plot for experiment 8 shows a very slight amplification of the plot for experiment 1 (the standard), but the smoothness of the system response is reduced. The plot for experiment 9 shows a slightly downward scaling of the plot for experiment 1. That is the effect of expanding or reducing the range of the universe of discourse for *output.* The plot for experiment 10 does not differ very much from the plot for experiment 1.

Figure 8 shows the plots for experiments 11, 12, and 13. The three plots indicate that the rule base has the greatest influence on the behaviour of the controller.

Figures 9 shows the plots for experiments 14, 15 and 16. The plots indicate that adjusting the modes of the *output* fuzzy sets have almost as much impact as adjusting the rule base. It means that when a fairly good rule base has been obtained, the desired system response can be achieved by adjusting the modes of the *output* fuzzy sets.

The simulation results confirm the generally held view that the greatest difficulty encountered in the design of a fuzzy logic controller and, in fact, in the design of any knowledge-based system, is the elucidation of the rules. This difficulty is commonly referred to as the Feigenbaum bottleneck, after Edward Fiegenbaum, who emphasized the fact that the real power of an expert system comes from the knowledge it possesses rather than the particular inference schemes and other formalisms it employs. As already stated, the results of experiments 11, 12, and 13 indicate that the rule-base has the greatest influence on the controller.

The simulation studies provided an explanation for a puzzle which came up during the early runs of the simulation program: why it is that most fuzzy logic controllers employ PD control structure for every kind of plant. Due to the fact that the process model used in the

simulation has no integrator, a significant amount of steady state error appeared. Such a problem would usually be taken care of by the inclusion of integral action in the controller. Initially, it was not very clear how this problem could be tackled in a control structure that is basically PD. However, further consideration of this problem cleared the difficulty: What integral action basically does is to maintain some control action even when the error is zero and non-changing. By adjusting the output fuzzy set that is asserted by the rule which fires when error is zero and non-changing in such a way that the centre of the domain of this fuzzy set is greater than zero something similar to integral action is achieved. This was how the steady state error (offset) was eventually nulled.

One problem immediately noticeable from the simulation studies is that. even though fuzzy controllers are known to be robust, if the parameters of the controlled process vary by wide margins. it will become necessary to retune the controller, for best performance. Because of the difficulties encountered in the tuning of a fuzzy logic controller. the current direction of research in FLC is towards self-tuning or self- adaptation.

5. Conclusion

This paper is a report of some simulation experiments carried out in order to isolate the effect of varying certain parameters of a FLC on the performance of the overall control system. The parameters investigated include the range of the universe of discourse the extent of overlap of the fuzzy sets, the rules in the rule base and the modes of the output fuzzy sets.

From the results of the simulation studies, it is very clear that there are many parameters a designer can vary in order to attain his design objectives. Varying the range of the universe of discourse for the inputs to the fuzzy controller affects the transient response of the system, and even the steady state error. Varying the width of the input fuzzy sets can be carried out to fine- tune the system. By far the greatest problem in the design process is the elicitation of the rules in the rule-base. It has been shown that, if it is possible to elucidate a fairly good rule base, the desired system response can be achieved by adjusting the modes of the output fuzzy sets.

References

1. E. H. Mamdani and B. S. Sembi, "Process control using fuzzy logic," in Fuzzy Sets: Theory and Applications to Policy Analysis and Information Systems, P. P. Wang and S. K. Chang, Eds. New York: Plenum Press, 1980.

2. D. G. schwartz and G. J. Klir, "Fuzzy logic flowers in Japan," IEEE Spectrum, July 1992, pp. 32-35.

3. E. Cox, "Adaptive fuzzy systems," IEEE Spectrum, Feb. 1993, pp. 27- 31.

4. E. H. Mamdani, "Application of fuzzy logic for control of simple dynamic plant," Proc. Inst. Elec. Eng., vol. 121, pp. 1585-1592, Dec. 1974.

5. L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, pp. 338-353, 1965.

6. J. Nie and D. A. Linkens, "Fast self-learning multi variable fuzzy controllers constructed from a modified CPN network," Int. J. Control. vol. 60, no. 3, pp 369-393, 1994.

7. E. Cox, The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using, and Maintaining Fuzzy systems. Boston, AP Professional, 1994.

8. B. Kosko, ″Addition as fuzzy mutual entropy," Information Sciences, vol. 73, pp. 273-284, Oct. 1993.
9. J. Efstathiou, Expert Systems in Process Control. Essex: Longman, 1989.

10. D. G. Schwartz et al., "Applications of fuzzy sets and approximate reasoning," Proc. IEEE, vol. 82, no. 4, pp. 482-498, April, 1994.

11.

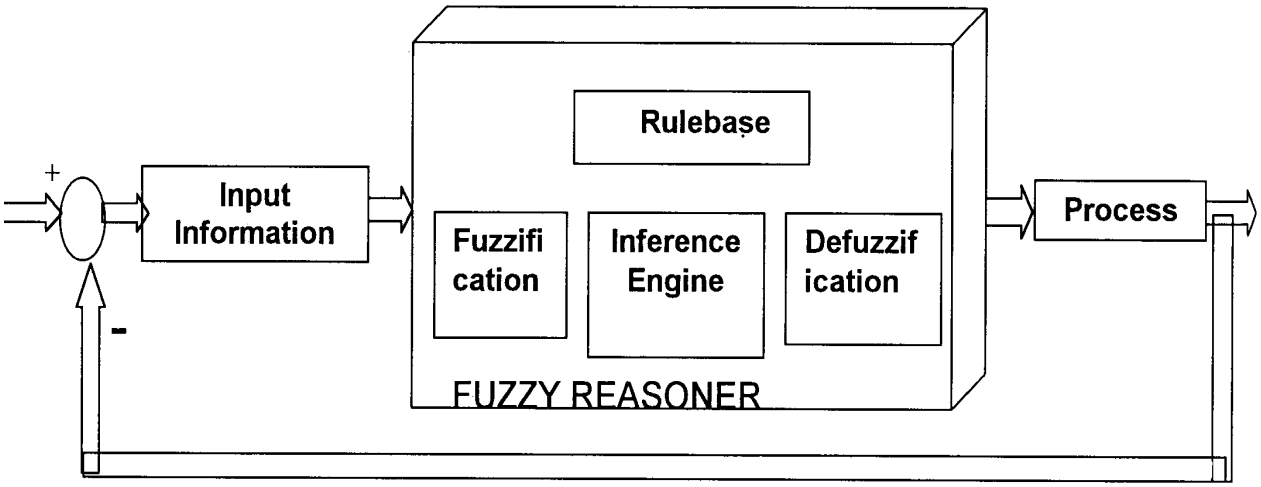

Figure 1. A fuzzy logic control system

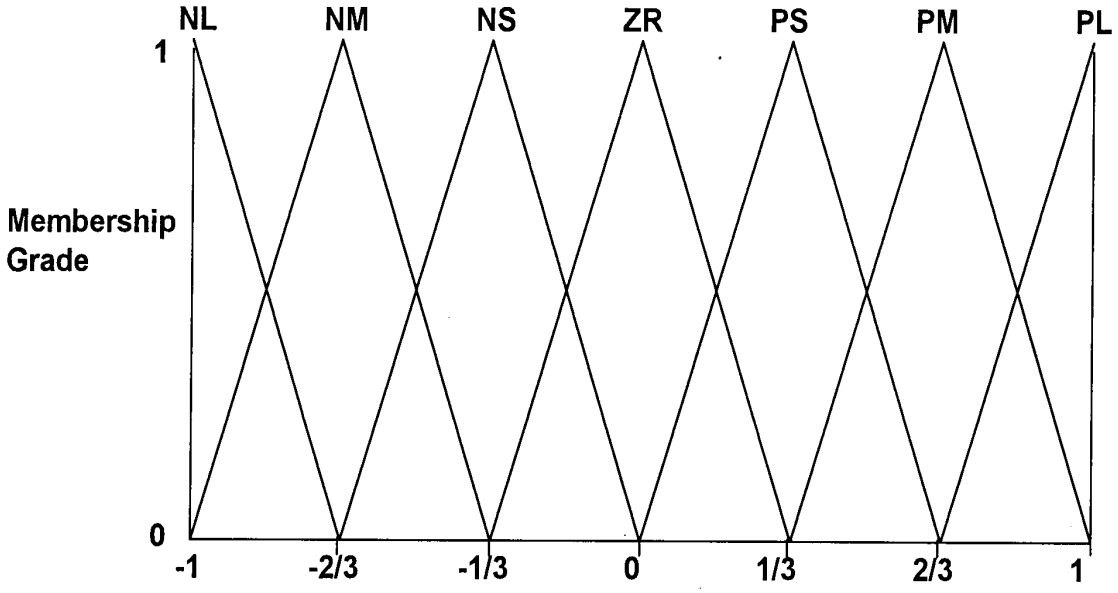

Figure 2: Fuzzy regions (term set) used for input and output variables

Error change / Error	LN	MN	SN	ZR	SP	MP	LP
LN	LN	LN	LN	MN	MN	SN	SN
MN	LN	LN	MN	MN	SN	SN	ZR
SN	LN	MN	SN	SN	ZR	ZR	SP
ZR	MN	SN	ZR	ZR	SP	SP	MP
SP	MN	SN	ZR	SP	SP	MP	LP
MP	SN	ZR	SP	SP	MP	LP	LP
LP	ZR	SP	SP	MP	LP	LP	LP

Figure 3. The rule-base.

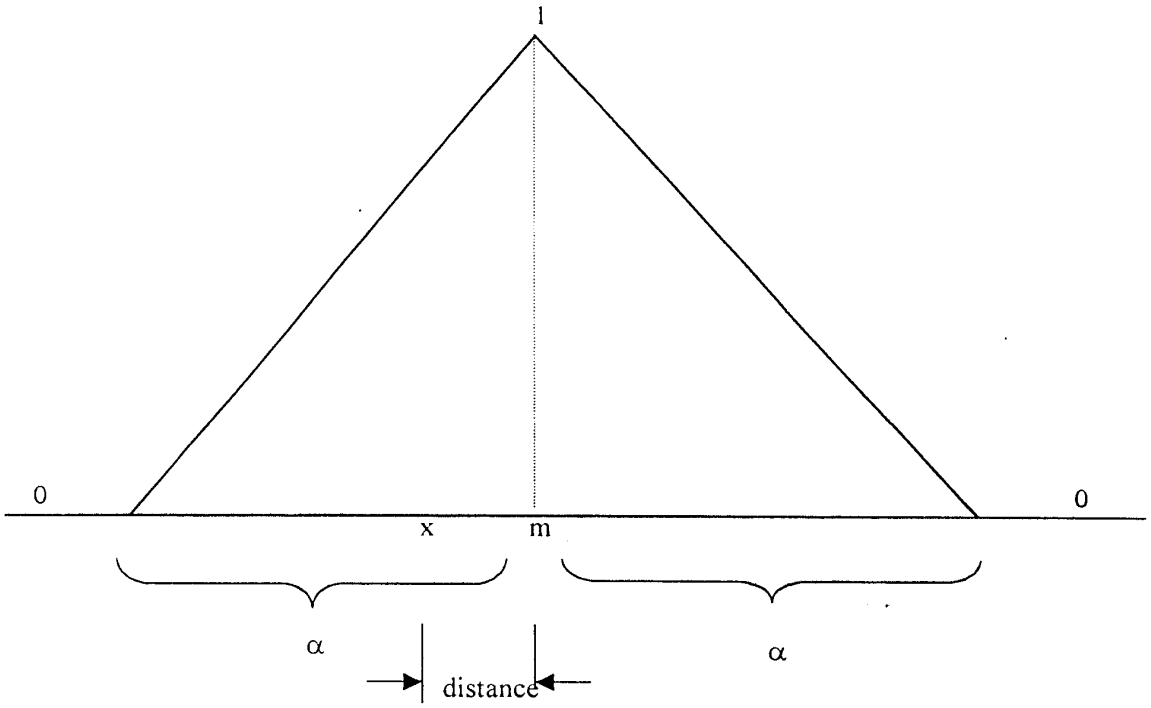

Figure 4. A triangular fuzzy set.

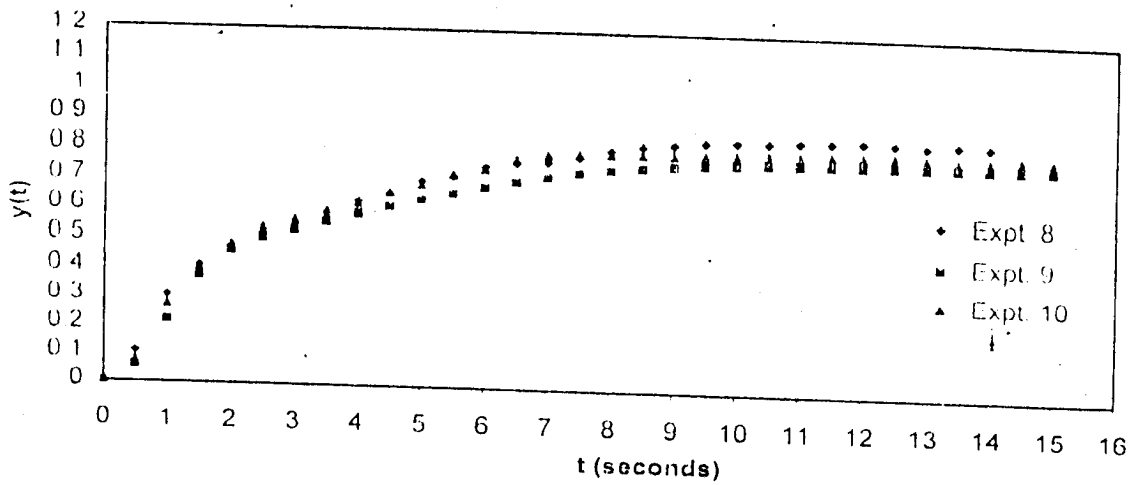

Figure 7. Results for experiments 8, 9, and 10

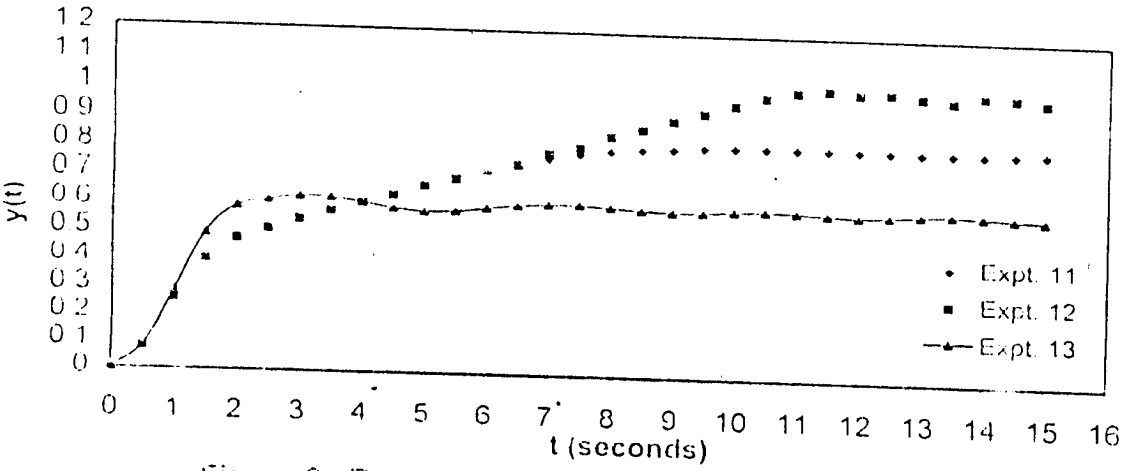

Figure 8. Results for experiments 11, 12, and 13

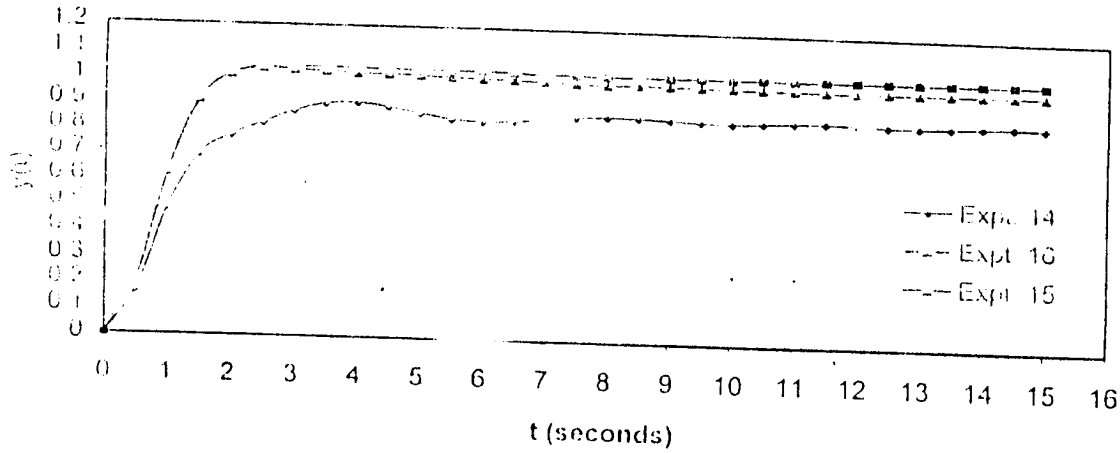

Figure 9. Results for experiments 14, 15, and 16

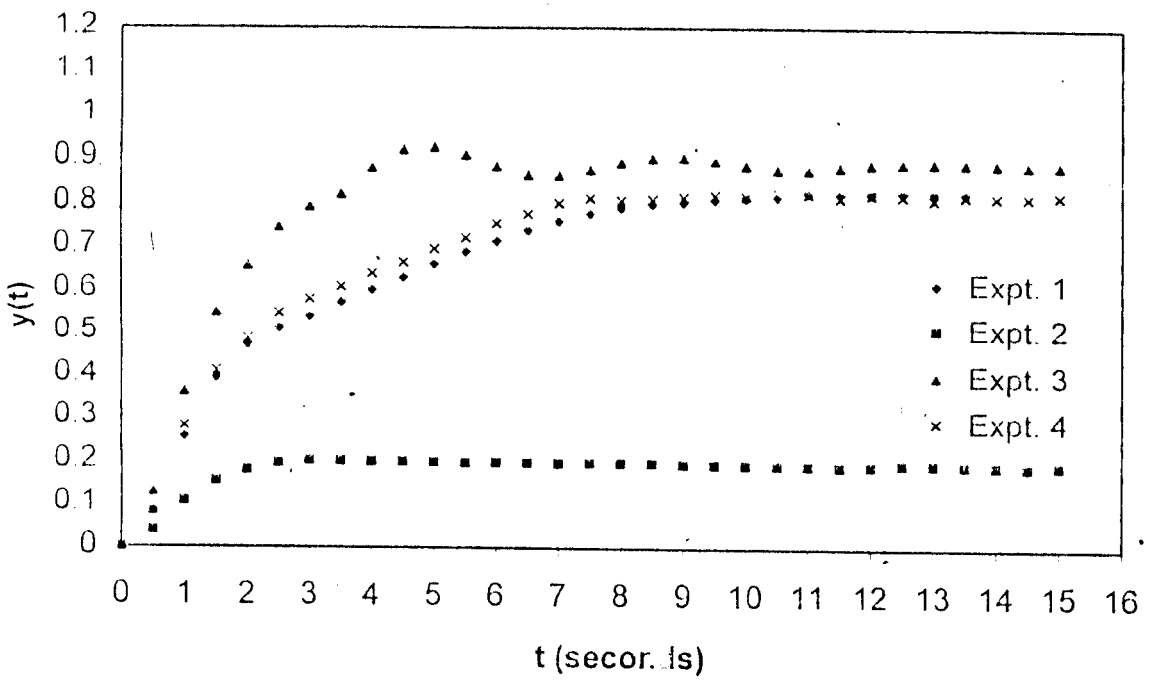

Figure 5. Results for experiments 1, 2, 3, and 4

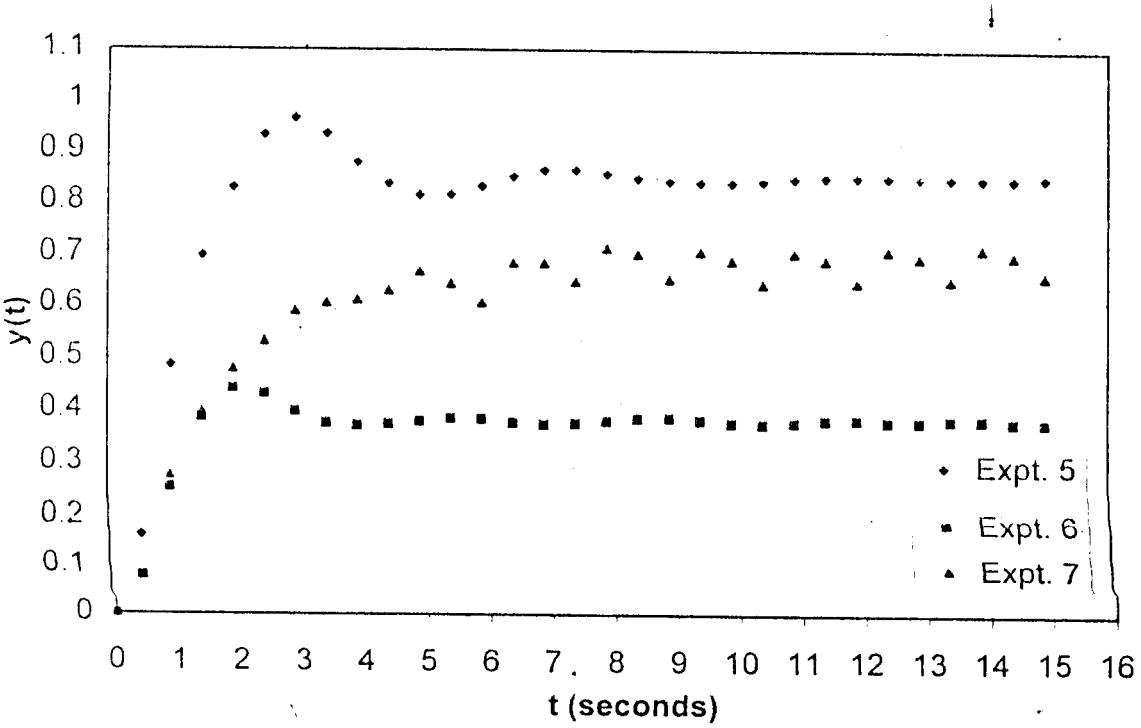

Figure 6. Results for experiments 5, 6, and 7