Engineering Journal of University of Qatar, Vol. 9, 1996, p.147–169

SELECTION OF OPTIMIZED FUZZY RULES FOR PROCESS CONTROL SYSTEMS: USING GENETIC ALGORITHMS

Seradj D. Katebi
Department of Computer Science & Engineering
School of Engineering, Shiraz University
Shiraz, Iran.

ABSTRACT

In this paper modeling, selection and optimization of fuzzy rules for multivariable feedback process control and decision making systems are developed and discussed. A general model for multivariable fuzzy rules is derived and sets of fuzzy parameters are assigned to the consequents of each rule. A genetic algorithm is used to optimize these parameters. A genetic algorithm search technique is also developed for the selection of the sub-set of appropriate rules from a larger theoretically possible set. The procedure is implemented on a multivariable dynamic model and results of extensive simulation studies are presented to demonstrate a satisfactory performance of the proposed approach.

INTRODUCTION

The difficulty in designing fuzzy decision making for multivariable feedback control systems lies in the existence of a large number of rules with complex structure (1). When the number of inputs to a multi-input multi-output system is increased, the number of rules required to cover all possible modes of system behavior will be increased accordingly. The complexity of the set of the required rules will also increase, because the antecedents of each rule require more connective operators. The increased number of outputs does not increase the number of rules explicitly, but to be able to control these outputs, it is required to introduce appropriate inputs with expanded antecedents. Therefore, one of the main tasks in designing any fuzzy decision making system is the generation of fuzzy rules and construction of the fuzzy rule base in an efficient and optimum manner (2). The rule base, essentially, describes the normal operation of the system from the view point of an expert operator. Several procedures have been suggested for generation of the fuzzy rules (3-6). Some of the more widely used methods are:

(a) The human operator's experience.
(b) Fuzzy modeling of the process, using fuzzy implications concerned with inputs, state variables and outputs.
(c) Self-organizing control, based on meta-rules from which the control rules can be created or changed.
(d) Modeling the operator's control action.
(e) Simulation of the mathematical model.
(f) The physical model.
(g) Neural networks.
(h) Learning from examples.

However, extraction of the appropriate rules describing the behavior of a system, from an experienced expert or any other methods is a difficult task and may not be practical or readily achievable in many cases.

Recently extensive research has focused on the field of soft computing and in particular many applications of Genetic Algorithms (GAs) in designing and optimizing fuzzy systems have been reported (7). Implementation of the genetic algorithms is conceptually straightforward and is basically an iterative search procedure inspired by the laws of natural selection and genetics. The object of the search by GA is to find an optimum fitness function. A "population" of possible values called individuals is considered over the given search space, and at each iteration a new population is generated. The new generation comprises individuals which are closer to the optimum value of the fitness function. The generation of these new populations is based on the evaluation of the fitness values and selection of the better fitted individuals. Using the genetic operators such as mutation, the selected individuals are combined to form a new generation, and as the algorithm successively iterates, the selected individuals normally tend towards the optimum fitness function. The GAs have been successfully applied in many diverse area, such as search, function optimization, scheduling, vision, control and machine learning (8-11). Efficient applications of GAs in conjunction with fuzzy logic controller design have been reported by several authors (10,14). In (10) a three-phase framework for learning dynamic control systems has been studied and a genetic algorithm is applied to drive control rules as decision tables. In the second phase the rules are automatically transformed into a comprehensive form and in the last stage the final rules are tuned via manipulation of the fuzzy relational matrix. Park and Kandel (14) show that the performance of fuzzy control system may be improved if the fuzzy reasoning model is supplemented by a genetic-based learning mechanism. They employed a GAs based procedure to optimize the set of parameters for the fuzzy reasoning model based either on their initial subjective selection or on a random selection. The main attention in this research was, therefore, paid to the selection of sets of optimized rules subject to satisfaction of specified performance criteria. GAs are applied in two distinct stages. In the first stage a number of effective and efficient rules is selected from a larger theoretically possible set and in the second stage the parameters of the selected rules are optimized by tuning the appropriate membership functions.

In this paper modeling, selection and optimization of fuzzy rules for multivariable feedback process control and decision making systems are developed and discussed. A general model for multivariable fuzzy rules is derived and sets of fuzzy parameters are assigned to the consequents of each rule. A genetic algorithm is used to optimize these parameters. A genetic algorithm search technique is also developed for the selection of the sub-set of the appropriate rules from a larger theoretically possible set. The procedure is implemented on a multivariable dynamic model and results of extensive simulation studies are presented to demonstrate a satisfactory performance of the proposed approach.

PROBLEM FORMULATION

A fuzzy system generally consists of three major components as shown in figure (1).

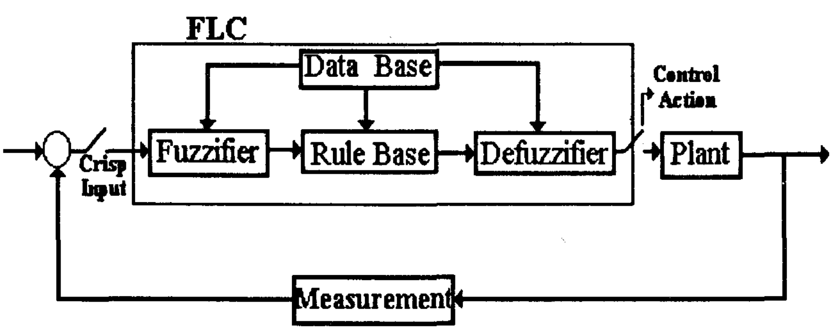


Fig. 1. Fuzzy logic control system configuration

The fuzzification and defuzzification procedure are fully described elsewhere (2) and for the purpose of brevity will not be discussed in this work. It is the inference engine which is the main concern of this work. The inference engine has an expert internal structure and is the knowledge base of the fuzzy logic controller. It consists of two distinct parts, the rule base and the database. One of the most important tasks in designing fuzzy logic control systems is to construct the knowledge base.

Consider a multivariable fuzzy decision making system with n inputs and m outputs. The processes of fuzzification assigns one or more fuzzy sets (represented by the appropriate membership functions) to each fuzzy input x_i, such that

$$x_i \subset X_i, \quad i = 1, 2, \ldots, n$$
$$x_i \subset \bigcup_{j=1}^{l_i} x_i^j \tag{1}$$

where x_i^j is a membership function and l_i is the number of assigned fuzzy sets to the fuzzy input x_i. Theoretically, the maximum number of applicable rules is given as.

$$r_a = \prod_{i=1}^{n} l_i \tag{2}$$

The formulation for the determination of the antecedents of each rule, represented by Eq.(2) is posed as a search problem in the space S_a, defined as

$$S_a = X_1 \times X_2 \times \ldots \times X_n \tag{3}$$

In a similar manner, each fuzzy output is represented by the union of fuzzy sets, assigned to the output variables, that is

$$y_i \subset Y_i, i = 1, 2, \ldots, m$$
$$y_i \subset \bigcup_{j=1}^{q_i} y_i^j \tag{4}$$

Hence the identification of the consequents of each rule may be viewed as a search problem in the space S_c, defined as;

$$S_c = Y_1 \times Y_2 \times \ldots \times Y_m \tag{5}$$

The maximum number of possible candidates for the consequent of each rule is given as follows;

$$r_c = \prod_{i=1}^{m} q_i \tag{6}$$

It can be deduced that $r_a * r_c$ different possibilities exist for formation of a rule. The construction of the fuzzy rule base may be posed as the problem of finding a sub-set of r optimum rules from the larger set of $r_a * r_c$ possible rules. The combinatory number of conditions is given by

$$t = r_a . r_c$$

$$v = \frac{t!}{(t-r)!} = \prod_{i=0}^{r-1}(t-i \tag{7}$$

where v is the number of IF/THEN conditions.

The problems formulated and posed above are solved in two steps. In the first step, each rule is modeled by a set of parameters representing the fuzzy association of inputs with outputs and GAs are used for optimization of these parameters. The second step is aimed at the selection of a set of rules based on the determination of the proper consequents related to the optimized antecedents obtained in first step. The GA is employed in the second step to select that set of fuzzy rules which satisfies the defined fitness functions related to the time domain performance of the system.

Although there are a variety of possible approaches to the development of control systems performance objectives, the first line of approach is generally to consider the time domain specifications. The reasons for this choice are many, but the most pervasive ones are that in practice the designer is mainly concerned with the transient response and its robustness, and the simplicity of automated calculation of the parameters in the context of numerical search techniques. Specifically, time domain based analysis assumes that the functional form of the required decision boundaries for the parameters can be selected a priori. This differs from the powerful graphical frequency domain performance which describes the desired design boundaries by the shapes of the frequency response displays.

In time domain a numerical synthesis procedure can be formulated as a set of inequalities:

$$F_i(R) < C_i \qquad i = 1, 2, \ldots m$$

where C_i is a set of real numbers, R denotes a vector $[r_1,...,r_n]$ and F_i are functions of R, m is the number of constrained specifications and n is the total number of rules. The component of R represent the fuzzy rules and the above inequalities represent time domain specifications. The dynamic behavior of the process can be specified by a set of time functions $f(t)$ for $0 < f(t) < \infty$, where $f(t)$ is chosen in this work to represent such figures of merit as system rise time, settling time, percentage overshoot and steady state error.

MODELING THE FUZZY RULES

Design and implementation of the rule base normally involves two basic problems. One is the generation of an appropriate and effective set of rules and the second is to attempt to keep the number of these rules within a reasonable and acceptable bound. The number of rules will increase considerably with the number of inputs as well as the number of the labels of the membership functions for each fuzzy variables. One the main objectives of any fuzzy expert system is to replace or assist the human expert operator. Due to the nonlinear and time varying behavior and other uncertainties in a practical control system, the control tasks may be so complex that either it can not be modeled by the conventional mathematical method or the derived model is so complicated that it can hardly be used. Therefore, when the controller model does not consist of crisply defined mathematical statements and relations, then traditional mathematical methods can not be applied directly and fuzzy algorithms that can deal with uncertain and incomplete information (fuzzy entities) may be employed. One of the widely used method of fuzzy knowledge representation is the IF/THEN rules.

Consider the following simple fuzzy rule:

$$\text{IF} \quad x \quad \text{THEN} \quad y \tag{8}$$

where x and y are both fuzzy sets defined on their respective universes of discourse. The above rule is the representation of a fuzzy relation between x and y defined on the product of x and y. Fuzzy relations in the same or different product spaces can be combined with each other by the operation "composition". Different compositions have been suggested which differ in their results and also with respect to their mathematical properties. One of the most widely used is the compositional rule of inference or the Max-Min (OR-AND operation) which has been suggested by Zadeh (12) and may conveniently be used for modeling the fuzzy rule as follows:

$$R = x \circ y \tag{9}$$

Where R is the fuzzy relational matrix and o denotes composition. The model identification of a rule can be performed by finding the corresponding relational matrix that relates every fuzzy output y to a fuzzy input x, using the above compositional rule of inference.

In general the relationship between two fuzzy variables may be expressed as:

$$y = x\Delta\Phi \tag{10}$$

where y the fuzzy relational matrix, Φ is a fuzzy parameter and Δ is a nonlinear fuzzy operator that relates fuzzy output y to fuzzy input x. Eq.(10) is the general form of Eq. (9), extended to complex and multivariable systems. Based on the choice of the membership functions, several different characteristics can be assumed for Φ and Δ. In fact, determination of Φ is viewed as an optimization process based on a set of performance indices. Fuzzy operator Δ is also determined according to the type and structure of Φ. Generally, three structures for Φ maybe defined (13) as follows.

a) Singleton

When is Φ defined as a fuzzy set with a membership function associated with only a single membership grade denoted by λ, then the fuzzy operator Δ will take the form of a simple minimum (MIN operator) such that;

$$y = x \wedge \lambda \tag{11}$$

$$\mu_Y(y) = \mathrm{MIN}(\mu_X(x)\ ,\lambda) \tag{12}$$

where μ_X and μ_y are the membership grades of x and y, respectively and $\wedge$ denotes the MIN operator.

b) Fuzzy Set

When Φ is defined as a fuzzy set, that is, a set of ordered pairs defined on some universe of discourse denoted by A and represented by a membership function or a set of linguistic variables (labels), then the fuzzy operator Δ may also be defined as a simple minimum operator such that:

$$y = x \wedge A \tag{13}$$

$$\mu_Y(y) = \mathrm{MIN}(\mu_X(x), \mu_A(A)) \tag{14}$$

where $\mu_A(A)$ is the membership grades of A.

c) Fuzzy Relation Matrix

When Φ is defined as a fuzzy relation matrix denoted by R, that is, Φ is itself a fuzzy relation , then Δ is defined as Max-Min operator, giving;

$$y = x \circ R \tag{15}$$

$$\mu_y = \mathrm{MAX}\{\mathrm{MIN}[\mu_X(x), \mu_{X\times Y}(R)]\} \tag{16}$$

DESIGN METHOD AND COMPUTATIONAL PROCEDURES

The techniques for the selection of the optimized fuzzy rules consist of the following algorithms and comprise the following steps;

Step 1: General form of the multivariable fuzzy rules with n inputs and m outputs is considered;

$$\mathrm{IF} \ \bigwedge_{i=1}^{n}(x_i) \ \mathrm{THEN} \ \bigwedge_{j=1}^{m}(y_j) \tag{17}$$

where X and Y are fuzzy vectors given as;

$$\begin{aligned} X &= [x_1, x_2, \ldots, x_n]' \\ Y &= [y_1, y_2, \ldots, y_m]' \end{aligned} \tag{18}$$

and $'$ denotes transposition. Each of the multivariable rules in Eq.(17) may be expressed as:

$$\text{IF} \bigwedge_{i=1}^{n} (x_i) \text{ THEN } Y = \Phi \Delta X \tag{19}$$

The model of the consequent part of the above rules is employed for the purpose of the optimum rule selection. The structure and dimension of Φ depends on the number of inputs and outputs of the controller and in general is expressed in matrix form as

$$\Phi = \begin{bmatrix} \phi_{11} & \cdots & \phi_{1n} \\ \vdots & \ddots & \vdots \\ \phi_{m1} & \cdots & \phi_{mn} \end{bmatrix} \tag{20}$$

Consequently;

$$\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} \phi_{11} & \cdots & \phi_{1n} \\ \vdots & \ddots & \vdots \\ \phi_{m1} & \cdots & \phi_{mn} \end{bmatrix} \Delta \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \tag{21}$$

At this stage, $\phi_{11},\ldots, \phi_{mn}$ are evaluated and the genetic algorithms are implemented for the parameter optimization. The following three characteristics are considered for the determination of the parameters of $\phi_{11},\ldots,\phi_{mn}$.

a) Singleton type: If $x_i \subset A$ and $y_j \subset B$, where A and B are the universes of discourse assigned to input x_i and output y_j respectively, then we have

$$y_j = \bigvee_{i=1}^{n} (\phi_{ij} \wedge x_i) \tag{22}$$

$$\mu_B(y_j) = \underset{i=1}{\overset{n}{\text{MAX}}}\{\underset{\text{for-all-cases}}{\text{MIN}}(\phi_{ij}, \mu_A(x_i))\} \tag{23}$$

where $\vee$ denote the MAX operation.

b) Fuzzy-set: In this case, ϕ_{ij} is represented by a membership function giving the compatibility degree of input x_i and output y_j. Fuzzy set ϕ_{ij} is defined on the related universe of discourse C. Therefore, the output y_j may be obtained by Eq.(22) and

$$\mu_B(y_j) = \underset{i=1}{\overset{n}{MAX}}\{\underset{for-all-cases}{MIN}(\mu_C(\phi_{ij}), \mu_A(x_i))\} \tag{24}$$

c) Fuzzy-relation: In this case, ϕ_{ij} is defined as a fuzzy relational matrix, defined on the Cartesian product of $A \times B$.

$$\phi_{ij} = \begin{bmatrix} r_{11} & \cdots & r_{1q} \\ \vdots & \ddots & \vdots \\ r_{q1} & \cdots & r_{qq} \end{bmatrix} \tag{25}$$

Where q is the number of discrete elements of the universe of discourse. Similarly the output y_j is obtained by the Eq. (22) and

$$\mu_B(y_j) = \underset{i=1}{\overset{n}{MAX}}\{\underset{for-all-cases}{MAX}[MIN(\mu_{A\times B}(\phi_{ij}), \mu_A(x_i))]\} \tag{26}$$

Therefore, in each of the above cases, different criteria will be optimized by GAs. In first case only the number of rules is optimized, since the membership function is a singleton. In case (b) the number of rules as well as the parameters of the membership function are optimized and so on. Table 1 shows the number of parameters to be optimized by GAs.

Table 1. Number of Parameters to be Optimized

TYPE	NUMBER OF PARAMETERS
SINGLETON	n* m
FUZZY-SET	n* m * q
FUZZY-RELATION	n* m* q * q

Step 2: The genetic algorithm is implemented for the purpose of determining the optimized values of the parameters ϕ_{11} to ϕ_{mn}. The length of each string depends on the choice of the characteristics ϕ_{11} to ϕ_{mn} and includes all parameters, in each case, as shown in Table 2 below.

Table 2. Structure of Each String

$\phi 11$	$\phi 12$	....	ϕmn

Step 3: In this step another GA is employed to determine the antecedents and select the number of optimum rules. In this case each string consists of a number of fuzzy rules with optimized parameters. The length of each string depends on the desired and heuristically determined number of fuzzy rules for the completion of the rule base. Table 3 shows the structure of each string when *s* optimum fuzzy rules are considered. Figure(2) shows the flow diagram of computational procedures.

Table 3. Structure of Each String of the Fuzzy Rules

INPUT 1	...	INPUT*n*	INPUT 1	...	INPUT*n*	...	INPUT 1	...	INPUT*n*
Antecedent 1			Antecedent 2				Antecedent *r*		

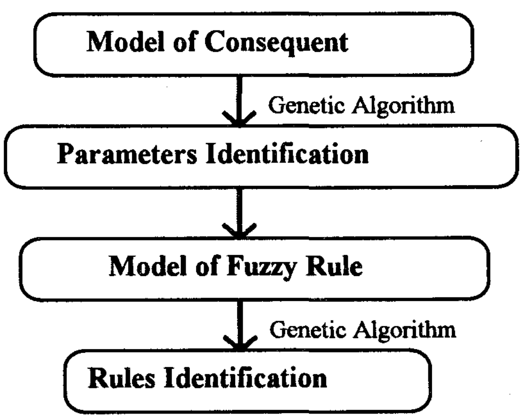


Fig. 2. Block diagram of the computational procedure

SIMULATION RESULTS

a) A Double Tank Reactor

In this example the multivariable system under study consists of two interconnected tank reactors as shown in figure (3). The inputs to the system are flow q_{i1} and q_{i2} and the output values are represented by q_{o1} and q_{o2}, respectively. The tanks are linked by the connecting pipe, the flow through which is dependent on the head difference and the pipe characteristics. A laminar flow is assumed and the other dimensions are identical. The nonlinear mathematical model describing the system is given by the following differential equations

$$A1\frac{dh_1}{dt} = q_{i1} + \frac{h_1 - h_2}{\Re} - a_1\sqrt{2gh_1} \quad (27)$$

$$A2\frac{dh_2}{dt} = q_{i2} - \frac{h_1 - h_2}{\Re} - a_2\sqrt{2gh_2} \quad (28)$$

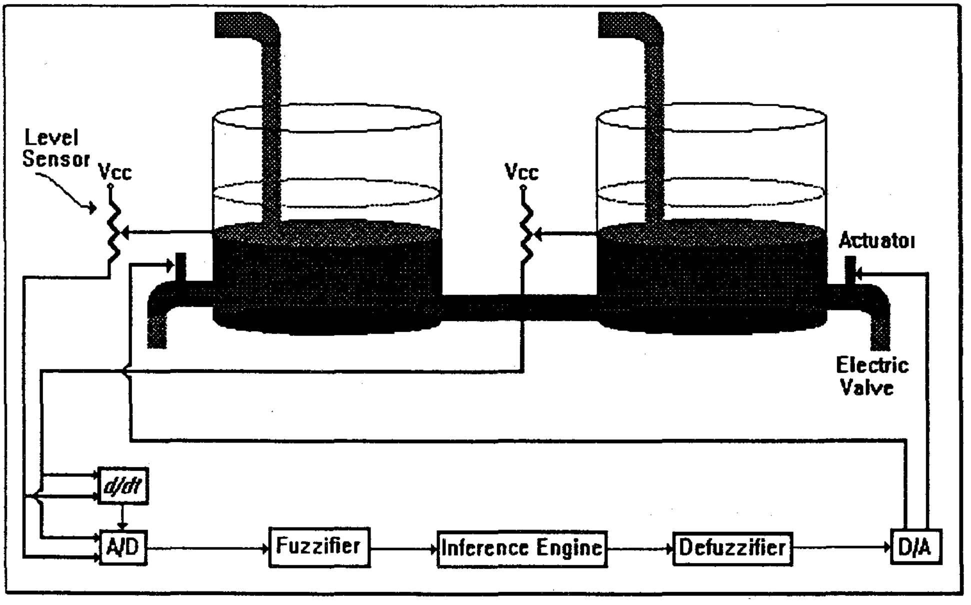


Fig. 3. The schematic diagram of the double tank reactor

where A_1, A_2 are the cross sectional area of the tanks, a_1, a_2 are the cross sectional area of the output valves, $\Re$ is the resistance of the connecting pipe, g is the acceleration due to gravity and h_1, h_2 are the liquid levels in the tanks. Control actions are applied to q_{o1} and q_{o2} which are output flows of the tanks. The control objective is to keep the liquid levels in the tanks at the desired set points. For the control system under study, there are four inputs to the Fuzzy Logic Controller, namely, h_1, h_2, h'_1, h'_2 and two outputs q_{o1}, q_{o2}. The block diagram for the control systems configuration is shown in figure (4). The Fuzzy Logic Controller (FLC) acts indirectly on the interactive components G_{12} and G_{21}.

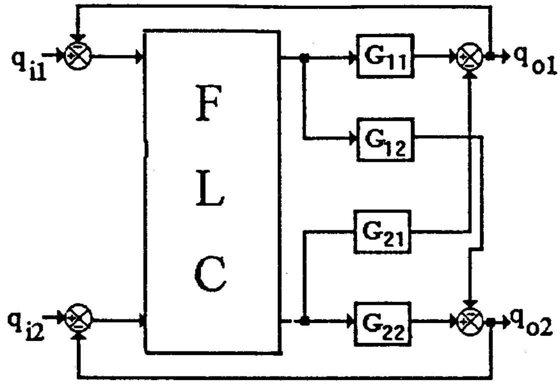


Fig. 4. Block diagram for the process control systems

The multivariable fuzzy controller is designed for the above dynamic system with the following specifications. Four fuzzy inputs, two fuzzy outputs and five labels assigned to each variable as shown in figure (5).

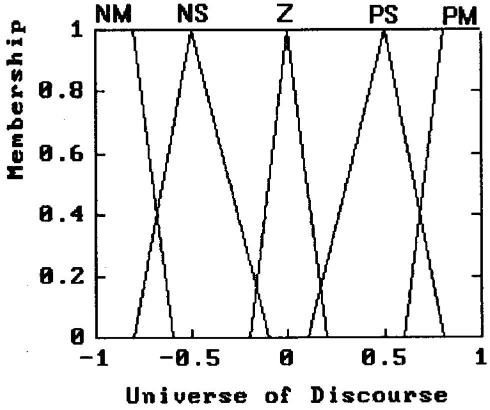


Fig. 5. The membership functions assigned to fuzzy variables

The maximum number of possible rules is obtained as:

$$r_a = 5^4 \ , \ r_c = 5^2$$

$$t = r_a . r_c = 15{,}625$$

If we select r number of rules from the total of t rules, then the number of possible cases is given by Eq. (7), for example for r=20 we have;

$$v = \frac{15{,}625!}{(15{,}625-20)!} = \prod_{i=0}^{19}(15{,}625-i) = 7.4 \times 10^{83}$$

In order to initialize the search, the parameters of GA such as population size, mutation rate, and crossover rate have to be defined. However, the theory behind parameter setting for a GA gives little guidance for their proper selection (15,16). The mutation and crossover rate are the source of exploration and the population size is the source of exploitation in the search procedures. Setting the mutation rate low allows the algorithm to exploit a particular hyperplanes. Setting the mutation rate high allows the algorithm to explore several different hyperplanes. Crossover is significantly effected by the exploration-exploitation tradeoff. Determination of the optimal population size is another problem. If the population size is too small then GA may have an improperly constrained search space. If the population size is too large, an inordinate amount of time will be needed to perform all the evaluations. Therefore, the difficulty is in seeking the balance between exploration and exploitation and more often setting of the parameters is heuristic and depends on the discretion of the designer for the problem at hand. In a recent study, Schaffer (16), through extensive research, found the best parameter settings for on-line performance to be:

- population size = 20-30
- crossover rate = 0.75-0.95
- mutation rate = 0.005-0.01

For our problem, which is not an on-line problem, the size of population need not be unduly small. Based on several experiments, the number of generations was set to 20 and parameters of the GA for the optimization procedure of the singletons in the equation (23) were chosen as follows:

- population size = 60
- crossover rate = 0.9
- mutation rate = 0.05

The optimized singleton values calculated by GA are

$$F = \begin{bmatrix} .800 & .533 & .400 & .200 \\ .0 & .333 & .267 & 1.000 \end{bmatrix}$$

Again the GA was run for 20 generations and the parameters of GAs for the determination of the antecedents, based on the choice of 15 rules are taken as:

- population size = 60
- crossover rate = 0.80
- mutation rate = 0.03

The design and construction of the fuzzy rule base can now be completed. The set of final rules obtained by the application of GA is given in figure (6). The following abbreviations have been used in the rule base of the controller.

LEVEL 1	= h_1 Height of liquid in tank 1
LEVEL 2	= h_2 Height of liquid in tank 2
dLEVEL 1	= h'_1 derivative of h_1
dLEVEL 2	= h'_2 derivative of h_2
VALVE 1	= qo1 output flow from tank 1
VALVE 2	= qo2 output flow from tank 2
NS	= Negative Small
NM	= Negative Medium
Z	= Zero
PS	= Positive Small
PM	= Positive Medium.

IF Level1 is NM AND dLevel1 is PM AND Level2 is NS AND dLevel2 is NS,
THEN valve1 is Z AND valve2 is Z ; ALSO
IF Level1 is NS AND dLevel1 is PM AND Level2 is PS AND dLevel2 is NM,
THEN valve1 is Z AND valve2 is NS; ALSO
IF Level1 is Z AND dLevel1 is NS AND Level2 is PS AND dLevel2 is PM,
THEN valve1 is Z AND valve2 is PS; ALSO
IF Level1 is Z AND dLevel1 is Z AND Level2 is PM AND dLevel2 is PM,
THEN valve1 is PS AND valve2 is PM; ALSO
IF Level1 is Z AND dLevel1 is PS AND Level2 is NS AND dLevel2 is NS,
THEN valve1 is Z AND valve2 is Z ; ALSO
IF Level1 is Z AND dLevel1 is PS AND Level2 is PM AND dLevel2 is NM,
THEN valve1 is PS AND valve2 is NS; ALSO
IF Level1 is Z AND dLevel1 is PM AND Level2 is PM AND dLevel2 is Z ,
THEN valve1 is PS AND valve2 is PS; ALSO
IF Level1 is PS AND dLevel1 is NM AND Level2 is PM AND dLevel2 is Z ,
THEN valve1 is Z AND valve2 is Z ; ALSO
IF Level1 is PS AND dLevel1 is Z AND Level2 is NS AND dLevel2 is PM,
THEN valve1 is Z AND valve2 is PS; ALSO
IF Level1 is PS AND dLevel1 is PS AND Level2 is PS AND dLevel2 is PS,
THEN valve1 is PS AND valve2 is PS; ALSO
IF Level1 is PS AND dLevel1 is PS AND Level2 is PM AND dLevel2 is PM,
THEN valve1 is PS AND valve2 is PM; ALSO
IF Level1 is PS AND dLevel1 is PM AND Level2 is PS AND dLevel2 is PM,
THEN valve1 is PS AND valve2 is PM; ALSO
IF Level1 is PM AND dLevel1 is NM AND Level2 is NM AND dLevel2 is PM,
THEN valve1 is Z AND valve2 is PS; ALSO
IF Level1 is PM AND dLevel1 is NS AND Level2 is PM AND dLevel2 is PS,
THEN valve1 is PS AND valve2 is PS; ALSO
IF Level1 is PM AND dLevel1 is Z AND Level2 is NM AND dLevel2 is NS,
THEN valve1 is Z AND valve2 is NS .

Fig. 6. Generated fuzzy rules

The time response of the control system based on the resulting fuzzy rules is shown in figure (7), for different set points of tank 1 and tank 2.

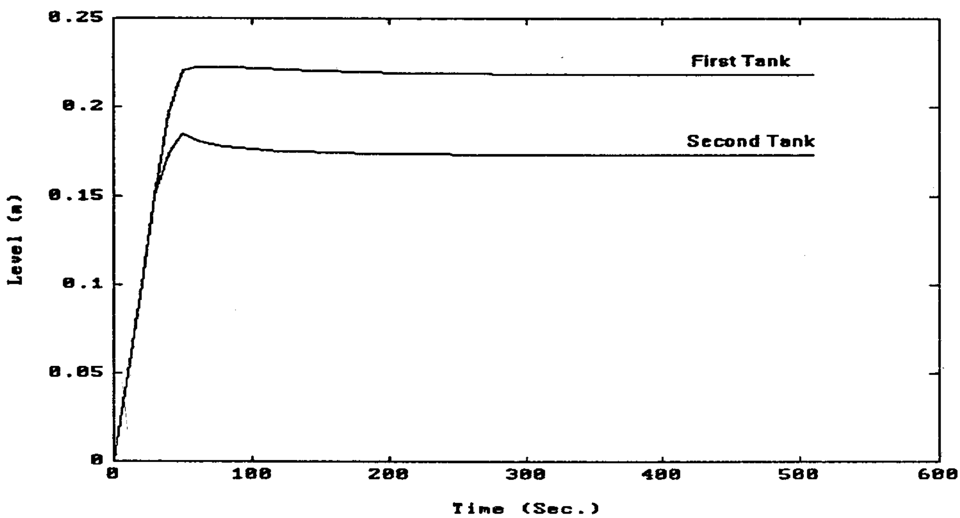


Fig. 7. System time response for the selected rules

b) The Cart-Pole System

In the second case study, the cart-pole control problem is considered. Besides being a standard benchmark problem for classical and alternative control approaches, it has much in common with a variety of tasks of greater practical importance, such as two legged walking and satellite attitude control (17). Figure (8) shows the schematic diagram of the system and the highly nonlinear mathematical model is given as:

$$\ddot{\theta} = \frac{g\sin\theta + \cos\theta[-f - m/\dot{\theta}^2\sin\theta + \mu_c \mathrm{sgn}(x)]/(m_c + m) - \mu_p\dot{\theta}/ml}{l[\frac{4}{3} - (m\cos^2\theta)/(m_c + m)]} \tag{29}$$

$$\ddot{x} = \frac{f + m/[\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta] - \mu_c\,\mathrm{sgn}(\dot{x})}{m_c} \tag{30}$$

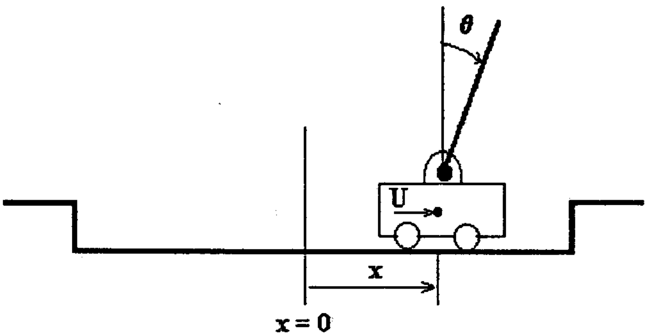
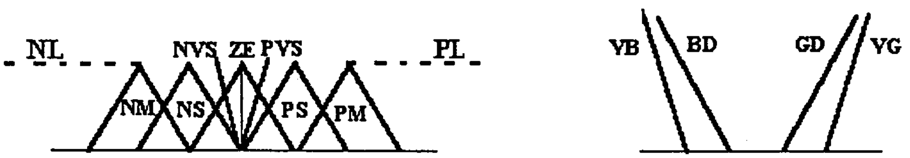


Fig. 8. Schematic diagram of the cart-pole systems

The state variables are chosen as x and $\dot{x}$, the position and the velocity of the cart and θ, $\dot{\theta}$ angular position and angular velocity of the pole with respect to vertical axis. The input to the system is the control force U(t). The objective is to keep the pole balanced while the cart is controlled to move in a prescribed range. The constants in equations (29) and (30) are: g the acceleration due to gravity, m_c and m are the masses of cart and pole respectively, l is the length of the pole, μ_c is the coefficient of friction between the cart and the track and μ_p is the coefficient of friction for the pole at the hinge. It is assumed that for $|x| > 2.4$ m and / or $|\theta| > 12°$, a failure has occurred. It is also assumed that the mathematical model of the system is not known to the control configuration, but a one dimensional vector representing the states of the system is available at the required instants. The fuzzy rules are derived based on the triangular membership functions for the state variables as well as the value of "goodness" of the control action. In order to obtain a satisfactory resolution and prevent possible oscillation, nine labels are considered for the state variables as shown in figure (9a).

The fuzzy labels for the state variable are; NL(Negative Large),NM(Negative Medium), NS(Negative Small), NVS(Negative Very Small), Zero(ZE), PSV(Positive Very Small), PS(Positive Small), PM(Positive Medium), LP(Positive Large). In the case of the "goodness" value, four labels VB.(Very Bad), BD(BaD), GD(GooD), and VG(Very Good), as shown in figure (9b) are used.



(a) State variables **(b) The "goodness value"**

Fig. 9. Membership functions

For this application, the rule base constructed in conjunction with GAs consists of eighteen rules, twelve of which are related to the control actions for balancing the pole. These are shown in figure (10) in the form of fuzzy associative memory. The other six are related to the positioning of the cart in the prescribed range and are shown in figure (11).

$\dot{\theta}$ \ θ	NL	NS	NVS	PVS	PS	PL
NM	VB	BD				GD
NS	BD				VG	GD
ZE						
PS	GD	VG				BD
PM	GD				BD	VB

Fig. 10. Fuzzy rules for controlling the pole

$C\dot{\theta}$ \ x	NL	NS
PS	GD	VG
PM	VG	

$C\dot{\theta}$ \ x	PS	PL
NM		VG
NS	VG	GD

Fig. 11. Fuzzy rules for controlling the cart

where $C\dot{\theta}$ denotes the change in $\dot{\theta}$ and is calculated as:

$$C\dot{\theta}_t = \dot{\theta}_t - \dot{\theta}_{t-1}$$

The system time response for the nominal values of the cart-pole physical parameters used (the length of the pole l=0.5 meter, weight of the pole m=0.1 kg, weight of the cart m_1=2.0 kg) are shown in figure (12). It is observed that the pole is balanced and the cart is controlled within the prescribed range.

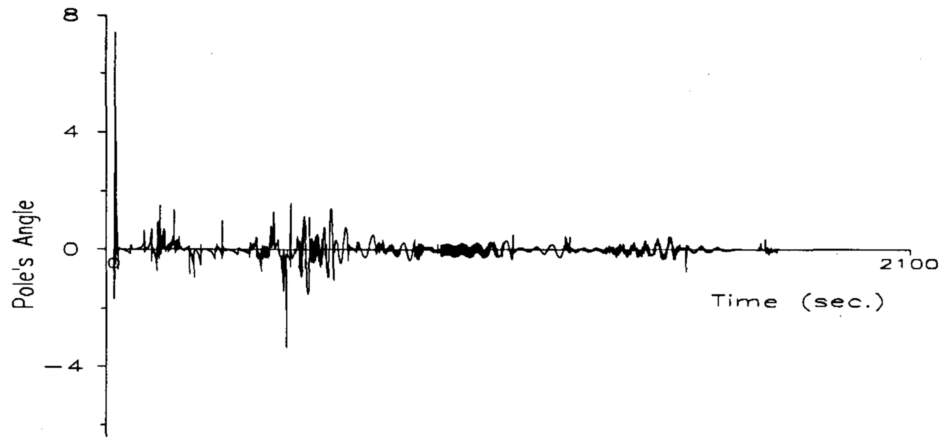


Fig. 12a. Time response of the cart-pole systems: The pole's angle

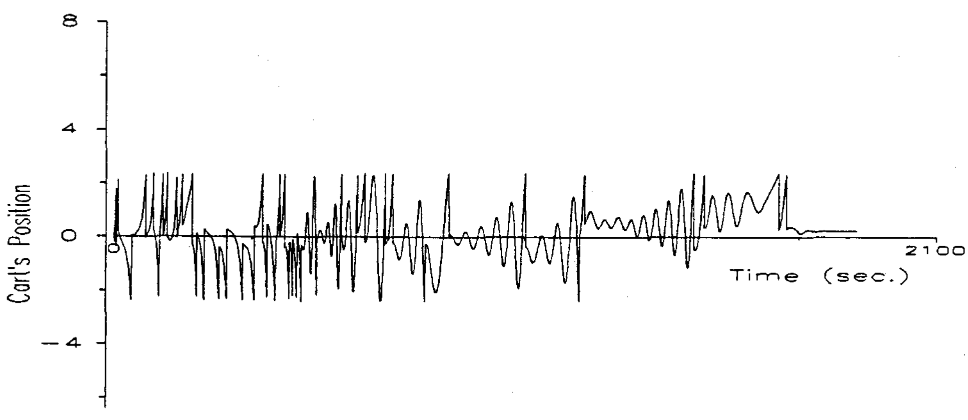


Fig. 12b. Time response of the cart-pole systems: The cart's position

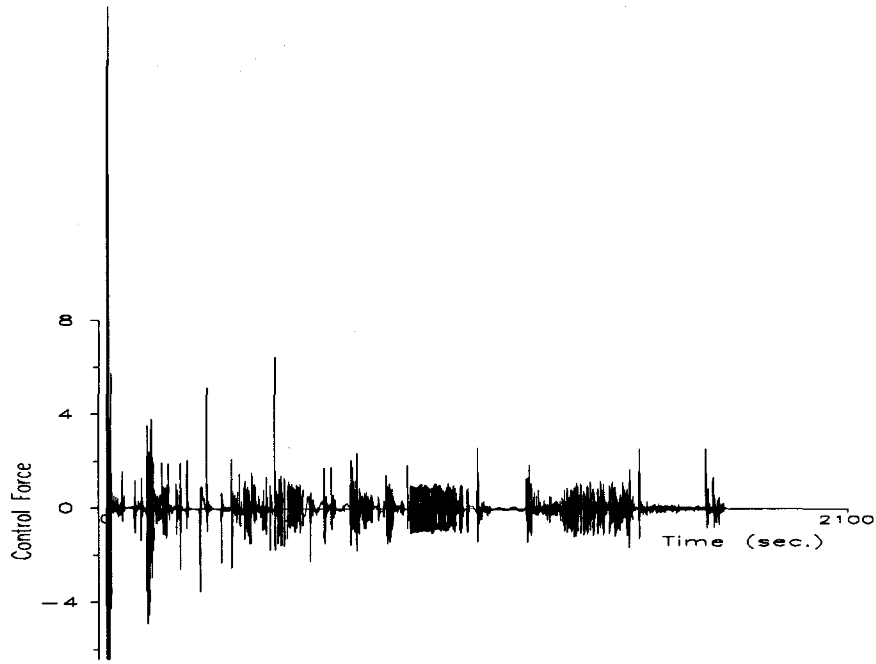


Fig. 12c. Time response of the cart-pole systems: Control force

In order to investigate the robustness of the designed fuzzy controller, extensive simulation studies were carried out for different values of the physical parameters of the cart-pole systems and in all cases acceptable results have been obtained.

CONCLUSIONS

A new method has been presented for the optimization and selection of the number of the fuzzy rules. A general model for the multi-dimensional fuzzy rules is derived and genetic algorithms are employed for the purpose of searching for optimum and appropriate rules. The salient features of the proposed approach are that no information about the behavior of the control system is required and the mathematical model of the system is not known to the controller. The procedure also provides the capability of finding certain desired number of rules in an optimized manner and is applicable to a wide range of systems and processes. In order to demonstrate the effectiveness of the proposed technique, it was implemented on a multivariable control systems in which the theoretically large number of possible rules is a disadvantage for design of the fuzzy controllers. Excellent results have been obtained in that only a set of 15 optimum rules out of a total of a very large number of possible rules, result in a good controller performance.

ACKNOWLEDGMENT

This research has been supported by the research consul of Shiraz University under the Grant Number 72-EN-805-448. The author would like to thank the anonymous reviewer for his very helpful and valuable suggestions for improving this paper.

REFERENCES

1. **Makrehchi, M., 1994.** Multivariable Fuzzy Logic Control Systems Design, M.Sc Thesis, Dept. of Computer Sci&Eng., Shiraz University, Shiraz, Iran.

2. **Lee, C. C., 1990.** Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I and II, IEEE Trans. SMC, Vol. 20, No. 2, pp. 404-432.

3. **Jamshidi, M., Vadiee, N. and Ross, T. J., 1993.** Fuzzy Logic and Control, Prentice Hall.

4. **Wang, L.X. and Mendel, J. M., 1992.** Generating Fuzzy rules by Learning from Examples, IEEE Trans. SMC, Vol. 22, No. 6, pp. 1414-1427.

5. **Takagi, T. and Sugeno, M., 1993.** Derivation of Fuzzy Control Rules from Human Operator's Control Action, in Proc. of IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis, Marseilles, France, pp. 55-67.

6. **Abdelnour, G. M., Chang, C. H., Huang, H. H. and Cheung, J. Y., 1991.** Design of a Fuzzy Controller Using Input and Output Mapping Factors, IEEE Trans. SMC, Vol. 21, No. 5, pp. 952-960.

7. **Kaar, C. and Gentry, E. J., 1993.** Fuzzy Control of pH Using Genetic Algorithms, IEEE Trans. Fuzzy Systems, Vol. 1, No. 1, pp. 46-53.

8. **Goldberg, D. E., 1994.** Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley.

9. **Davis, L., 1991.** Handbook of Genetic Algorithms, Van Nostrand Reinhold.

10. **Varsek, A., Urbanica, T. and Filipic, B., 1993.** Genetic Algorithms in Controller Design and Tuning, IEEE Trans. SMC, Vol. 23, No. 5, pp. 1330-1339.

11. **Kristinsson, K. and Dumont, G. A., 1992.** System Identification and Control Using Genetic Algorithms, IEEE Trans. SMC, Vol. 22, No. 5, pp. 1033-1046.

12. **Zadeh, L. A., 1973.** Outline of a New Approach to Analysis of Complex Systems and Processes, IEEE Trans. SMC, Vol. 3, No. 5, pp. 2-44.

13. **Pedrycz, W., 1992.** Association of Fuzzy Sets, IEEE Trans. SMC, Vol. 22, No. 6, pp. 143-14.

14. **Park, D., Kandel, A. and Langholz, G., 1994.** Genetic-based New Fuzzy Reasoning Models with Application to Fuzzy Control, IEEE Trans. SMC, Vol. 24, No. 1, PP. 30-47.

15. **Davis, L., 1989.** Adapting Operator Probabilities in Genetic Algorithms, in Proc. 3rd Int. Conf. Genetic Algorithms, pp. 61-69.

16. **Schaffer, J. D., Caruana, R. A., Eshelman, L. J. and Das, R., 1989.** A Study of Control Parameters Effecting On-Line Performance of Genetic Algorithm, in Proc. 3rd Int. Conf. Genetic Algorithms, pp. 51-60.

17. **Sammut, C. and Michie, D., 1991.** Controlling a `Black Box` Simulation of a Space Craft, AI Magazine, Vol.12, No. 1, pp. 56-63.