Intelligent Control of Dynamic Systems Using Type-2 Fuzzy Logic and Evolutionary Computing

Oscar Castillo, Gabriel Huesca, and Fevrier Valdez
Department of Computer Science
Tijuana Institute of Technology
Tijuana, Mexico

Abstract
We describe in this paper the use of evolutionary computing techniques for optimizing the design of intelligent controllers. Genetic algorithms can be used to optimize the topology of a fuzzy system for control. We are considering type-2 fuzzy logic for intelligent control and as a consequence the task of designing the fuzzy system is more difficult.

1. Introduction
We describe in this paper the application of a Hierarchical Genetic Algorithm (HGA) for fuzzy system optimization [1]. In particular, we consider the problem of finding the optimal set of rules and membership functions for a specific application [2]. The HGA is used to search for this optimal set of rules and membership functions, according to the data about the problem. We consider, as an illustration, the case of a fuzzy system for intelligent control.

Fuzzy systems are capable of handling complex, non-linear and sometimes mathematically intangible dynamic systems using simple solutions [3]. Very often, fuzzy systems may provide a better performance than conventional non-fuzzy approaches with less development cost [4]. However, to obtain an optimal set of fuzzy membership functions and rules is not an easy task. It requires time, experience and skills of the designer for the tedious fuzzy tuning exercise. In principle, there is no general rule or method for the fuzzy logic set-up, although a heuristic and iterative procedure for modifying the membership functions to improve performance has been proposed. Recently, many researchers have considered a number of intelligent schemes for the task of tuning the fuzzy system. The noticeable Neural Network (NN) approach [5] and the Genetic Algorithm (GA) approach [6] to optimize either the membership functions or rules, have become a trend for fuzzy logic system development.

The HGA approach differs from the other techniques in that it has the ability to reach an optimal set of membership functions and rules without a known fuzzy system topology [7]. During the optimization phase, the membership functions need not be fixed. Throughout the genetic operations [8], a reduced fuzzy system including the number of membership functions and fuzzy rules will be generated [9]. The HGA approach has a number of advantages:

1) An optimal and the least number of membership functions and rules are obtained
2) No pre-fixed fuzzy structure is necessary, and
3) Simpler implementing procedures and less cost are involved.

We consider in this paper the case of automatic anesthesia control in human patients for testing the optimized fuzzy controller. We did have, as a reference, the best fuzzy controller that was developed for the automatic anesthesia control [10, 11], and we consider the optimization of this controller using the HGA approach.

2. Evolution of Fuzzy Systems
To obtain an optimal set of fuzzy membership functions and rules is not an easy task. It requires time, experience, and skills of the operator for the tedious fuzzy tuning exercise. In principle, there is no general rule or method for the fuzzy logic set-up. Recently, many researchers have considered a number of intelligent techniques for the task of tuning the fuzzy set. Here, another innovative scheme is described [7]. This approach has the ability to reach an optimal set of membership functions and rules without a known overall fuzzy set topology. The conceptual idea of this approach is to have an automatic and intelligent scheme to tune the membership functions and rules, in which the conventional closed loop fuzzy control strategy remains unchanged, as indicated in Fig. 1.

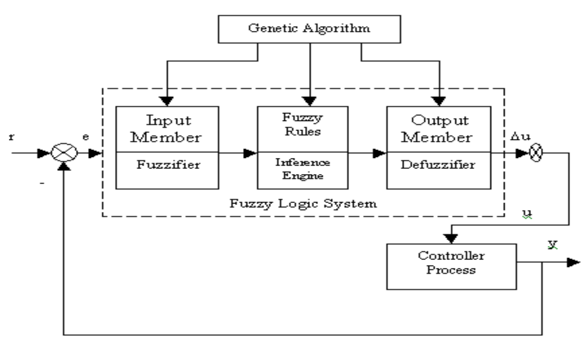

Fig. 1 Genetic algorithm for a fuzzy control system.

In this case, the chromosome of a particular system is shown in Fig. 2. The chromosome consists of two types of genes, the control genes and parameter genes. The control genes, in the form of bits, determine the membership function activation, whereas the parameter genes are in the form of real numbers to represent the membership functions.

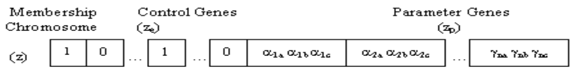

Fig. 2 Chromosome structure for the fuzzy system.

To obtain a complete design for the fuzzy control system, an appropriate set of fuzzy rules is required to ensure system performance. At this point it should be stressed that the introduction of the control genes is done to govern the number of fuzzy subsets in the system. Once the formulation of the chromosome has been set for the fuzzy membership functions and rules, the genetic operation cycle can be performed. This cycle of operation for the fuzzy control system optimization using a genetic algorithm is illustrated in Fig. 3. There are two population pools, one for storing the membership chromosomes and the other for storing the fuzzy rule chromosomes. We can see this in Figure 3 as the membership population and fuzzy rule population, respectively. Considering that there are various types of gene structure, a number of different genetic operations can be used. For the crossover operation, a one-point crossover is applied separately for both the control and parameter genes of the membership chromosomes within certain operation rates.

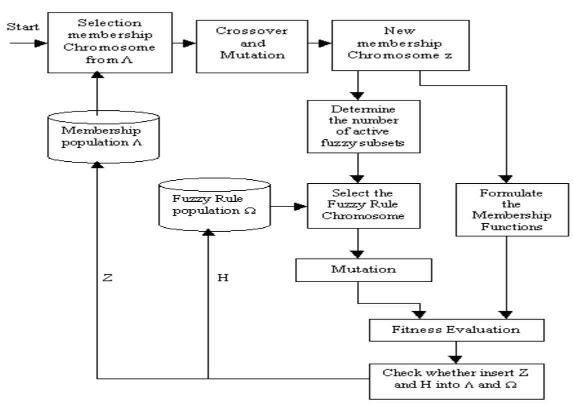

Fig. 3 Genetic cycle for fuzzy system optimization.

Bit mutation is applied for the control genes of the membership chromosome. Each bit of the control gene is flipped if a probability test is satisfied (a randomly generated number is smaller than a predefined rate). As for the parameter genes, which are real number represented, random mutation is applied. The fitness function can be defined in this case as:

$$f_i = \Sigma \mid y(k) - r(k) \; / \qquad (1)$$

where Σ indicates the sum for all the data points in the training set, and y(k) represents the real output of the fuzzy system and r(k) is the reference output. This fitness value measures how well the fuzzy system is approximating the real data of the problem.

3. Type-2 Fuzzy Logic

The concept of a type-2 fuzzy set, was introduced by Zadeh [14] as an extension of the concept of an ordinary fuzzy set (henceforth called a "type-1 fuzzy set"). A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership grade for each element of this set is a fuzzy set in [0,1], unlike a type-1 set [12, 14] where the membership grade is a crisp number in [0,1]. Such sets can be used in situations where there is uncertainty about the membership grades themselves, e.g., an uncertainty in the shape of the membership function or in some of its parameters. Consider the transition from ordinary sets to fuzzy sets [12]. When we cannot determine the membership of an element in a set as 0 or 1, we use fuzzy sets of type-1. Similarly, when the situation is so fuzzy that we have trouble determining the membership grade even as a crisp number in [0,1], we use fuzzy sets of type-2.

Example: Consider the case of a fuzzy set characterized by a Gaussian membership function with mean m and a standard deviation that can take values in $[\sigma_1,\sigma_2]$, i.e.,

$$\mu(x)=\exp\{-\tfrac{1}{2}[(x-m)/\sigma]^2\}; \quad \sigma \in [\sigma_1,\sigma_2] \qquad (2)$$

Corresponding to each value of σ, we will get a different membership curve (Figure 4). So, the membership grade of any particular x (except x=m) can take any of a number of possible values depending upon the value of σ, i.e., the membership grade is not a crisp number, it is a fuzzy set. Figure 4 shows the domain of the fuzzy set associated with x=0.7.

The basics of fuzzy logic do not change from type-1 to type-2 fuzzy sets, and in general, will not change for any type-n [13]. In Figure 5 we show the general structure of a type-2 fuzzy system. We assume that both antecedent and consequent sets are type-2; however, this need not necessarily be the case in practice.

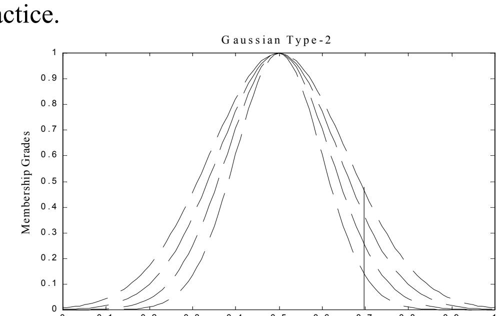

Fig. 4 A type-2 fuzzy set representing a type-1 set with uncertain deviation.

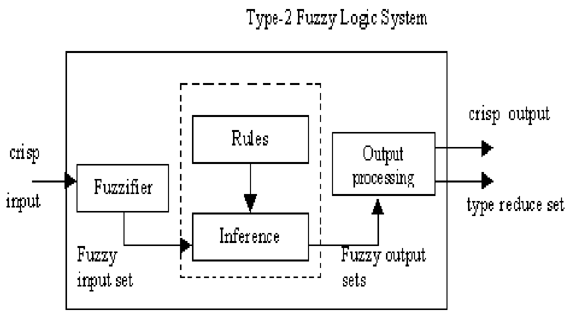

Fig. 5. Structure of a type-2 fuzzy system.

The structure of the type-2 fuzzy rules is the same as for the type-1 case because the distinction between type-2 and type-1 is associated with the nature of the membership functions. Hence, the only difference is that now some or all the sets involved in the rules are of type-2. In a type-1 fuzzy system, where the output sets are type-1 fuzzy sets, we perform defuzzification in order to get a number, which is in some sense a crisp (type-0) representative of the combined output sets. In the type-2 case, the output sets are type-2; so we have to use extended versions of type-1 defuzzification methods. Since type-1 defuzzification gives a crisp number at the output of the fuzzy system, the extended defuzzification operation in the type-2 case gives a type-1 fuzzy set at the output.

4. Application to Intelligent Control

We consider the case of controlling the anesthesia given to a patient as the problem for finding the optimal fuzzy system for control [11]. The complete implementation was done in the MATLAB programming language. The fuzzy systems were build automatically by using the Fuzzy Logic Toolbox, and genetic algorithm was coded directly in the MATLAB language. The fuzzy systems for control are the individuals used in the genetic algorithm, and these are evaluated by comparing them to the ideal control given by the experts. In other words, we compare the performance of the fuzzy systems that are generated by the genetic algorithm, against the ideal control system given by the experts in this application.

A. Anesthesia Control Using Fuzzy Logic

The main task of the anesthesist, during and operation, is to control anesthesia concentration. In any case, anesthesia concentration can't be measured directly. For this reason, the anesthesist uses indirect information, like the heartbeat, pressure, and motor activity. The anesthesia concentration is controlled using a medicine, which can be given by a shot or by a mix of gases. We consider here the use of isoflurance, which is usually given in a concentration of 0 to 2% with oxygen. In Figure 6 we show a block diagram of the controller.

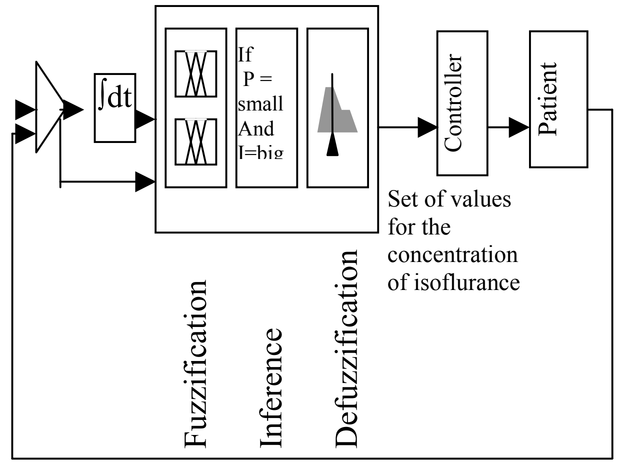

Fig. 6 Architecture of the fuzzy control system.

The air that is exhaled by the patient contains a specific concentration of isoflurance, and it is re-circulated to the patient. As consequence, we can measure isoflurance concentration on the inhaled and exhaled air by the patient, to estimate isoflurance concentration on the patient's blood. From the control engineering point of view, the task by the anesthesist is to maintain anesthesia concentration between the high level W (threshold to wake up) and the low level E (threshold to success). These levels are difficult to be determine in a changing environment and also are dependent on the patient's condition. For this reason, it is important to automate this anesthesia control, to perform this task more efficiently and accurately, and also to free the anesthesist from this time consuming job. The anesthesist can then concentrate in doing other task during operation of a patient.

B. Characteristics of the Fuzzy Controller

In this section we describe the main characteristics of the fuzzy controller for anesthesia control. We will define input and output variable of the fuzzy system. Also, the fuzzy rules of fuzzy controller previously designed will be described. The fuzzy sytem is defined as follows:

1) Input variables: Blood pressure and Error
2) Output variable: Isoflurance concentration
3) Nine fuzzy if-then rules of the optimized system, which is the base for comparison
4) 12 fuzzy if-then rules of an initial system to begin the optimization cycle of the genetic algorithm.

The linguistic values used in the rules are the following: PB = Positive Big, PS = Positive Small, ZERO = zero, NB =Negative Big, NS = Negative Small.

We show below a sample set of fuzzy rules that are used in the fuzzy inference system that is represented in the genetic algorithm for optimization.

if Blood pressure is NB and error is NB then conc_isoflurance is PS

if Blood pressures is PS then conc_isoflurance is NS
if Blood pressure is NB then conc_isoflurance is PB
if Blood pressure is PB then conc_isoflurance is NB
if Blood pressure is ZERO and error is ZERO then conc_isoflurance is ZERO
if Blood pressure is ZERO and error is PS then conc_isoflurance is NS
if Blood pressure is ZERO and error is NS then conc_isoflurance is PS
if error is NB then conc_isoflurance is PB
if error is PB then conc_isoflurance is NB
if error is PS then conc_isoflurance is NS
if Blood pressure is NS and error is ZERO then conc_isoflurance is NB
if Blood pressure is PS and error is ZERO then conc_isoflurance is PS.

5. Simulation Results

The genetic algorithm is able to evolve the topology of the fuzzy system for the particular application. We used 300 generations of 40 individuals each to achieve the minimum error. This is the case in which only nine fuzzy rules are needed for the fuzzy controller. The value of the minimum error achieved with this particular fuzzy logic controller was of 0.0064064, which is considered a small number in this application.

In Fig. 7 we show the simulation results of the fuzzy logic controller produced by the genetic algorithm after evolution. We used a sinusoidal input signal with unit amplitude and a frequency of 2 radians/second, with a transfer function of [1/(0.5s +1)]. In this figure we can appreciate the comparison of the outputs of both the ideal controller (1) and the fuzzy controller optimized by the genetic algorithm (2). From this figure it is clear that both controllers are very similar and as a consequence we can conclude that the genetic algorithm was able to optimize the performance of the fuzzy logic controller. We can also appreciate this fact more clearly in Fig. 8, where we have amplified the simulation results from Fig. 7 for a better view.

6. Conclusions

We consider in this paper the case of automatic anesthesia control in human patients for testing the optimized fuzzy controller. We did have, as a reference, the best fuzzy controller that was developed for the automatic anesthesia control [10, 11], and we consider the optimization of this controller using the HGA approach. After applying the genetic algorithm the number of fuzzy rules was reduced from 12 to 9 with a similar performance of the fuzzy controller. Of course, the parameters of the membership functions were also tuned by the genetic algorithm.

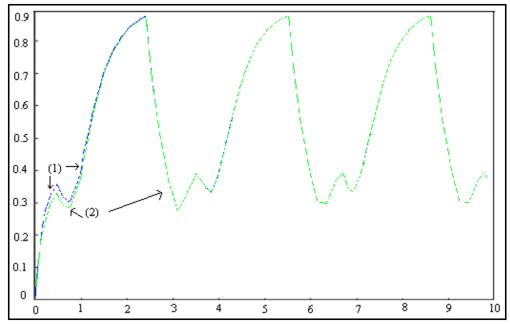

Fig. 7 Comparison between outputs of the ideal controller (1) and the fuzzy controller produced with the HGA (2).

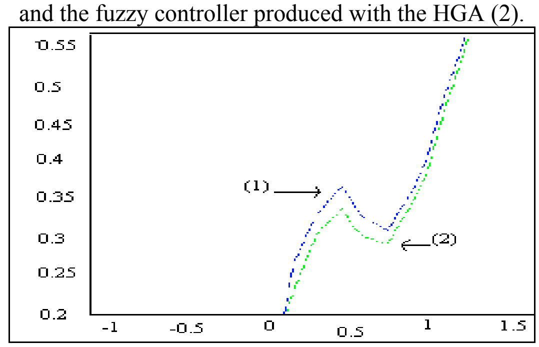

Fig. 8 Zoom in of figure 8 to view in more detail the difference between the controllers.

References

[1] K.F. Man, K.S. Tang, and S. Kwong (1999), "Genetic Algorithms: Concepts and Designs", Springer Verlag.
[2] J. Yen, and R. Langari (1999), "Fuzzy Logic: intelligence, control and information", Prentice Hall, Inc.
[3] J.-S. R. Jang, C.-T. Sun, and E. Mizutani (1997), "Neuro-fuzzy and Soft Computing, A computational approach to learning and machine intelligence", , Prentice Hall, Upper Saddle River, NJ.
[4] T.J. Procyk and E.M. Mamdani (1979), "A linguistic self-organizing process controller" *Automatica*, vol. 15, no. 1, pp 15-30.
[5] J.-S. R. Jang and C.-T. Sun (1995) "Neurofuzzy fuzzy modeling and control", *Proc. IEEE*, vol. 83, pp. 378-406.
[6] A. Homaifar and E. McCormick (1995), "Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms", *IEEE Trans. Fuzzy Systems*, vol. 3, pp. 129-139.
[7] K.-S. Tang, K.-F. Man, Z.-F. Liu and S.Kwong (1998), "Minimal fuzzy memberships and rules using hierarchical genetic algorithms", *IEEE Trans. on Industrial Electronics*, vol. 45, no. 1.
[8] J. Holland, (1975), "Adaptation in natural and artificial systems" (University of Michigan Press).
[9] T. Yoshikawa, T. Furuhashi and Y. Uchikawa (1996), "Emergence of effective fuzzy rules for controlling mobile robots using DNA coding method", *Proc. ICEC'96*, Japan, pp. 581-586.
[10] C.L. Karr and E.J. Gentry (1993), "Fuzzy control of pH using genetic algorithms", *IEEE Trans. Fuzzy Systems*, vol. 1, pp. 46-53.
[11] A. Lozano (2003), "Optimización de un Sistema de Control Difuso por medio de algoritmos genéticos jerarquicos", Thesis, Dept. of Computer Science, Tijuana Institute of Technology, Mexico.
[12] O. Castillo and P. Melin (2001), "Soft Computing for Control of Non-Linear Dynamical Systems", Springer-Verlag, Heidelberg, Germany.
[13] O. Castillo and P. Melin (2003), "Soft Computing and Fractal Theory for Intelligent Manufacturing", Springer-Verlag, Heidelberg, Germany.
[14] P. Melin and O. Castillo (2002), "Modelling, Simulation and Control of Non-Linear Dynamical Systems", Taylor and Francis, London, Great Britain.