Automatic Estimation of Parameters of Complex Fuzzy Control Systems

Yulia Ledeneva[1,2], René García Hernández[3] and Alexander Gelbukh[1]
[1]National Polytechnic Institute, Center for Computing Research
[2]Autonomous University of the State of Mexico
[3]Toluca Institute of Technology, Computer Science Department
Mexico

1. Introduction

Since the first fuzzy controller was presented by Mamdani in 1974, different studies devoted to the theory of fuzzy control have shown that the area of development of fuzzy control algorithms has been the most active area of research in the field of fuzzy logic in the last years. From 80´s, fuzzy logic has performed a vital function in the advance of practical and simple solutions for a great diversity of applications in engineering and science. Due to its great importance in navigation systems, flight control, satellite control, speed control of missiles and so on, the area of fuzzy logic has become an important integral part of industrial and manufacturing processes.

Some fuzzy control applications to industrial processes have produced results superior to its equivalent obtained by classical control systems. The domain of these applications has experienced serious limitations when expanding it to more complex systems, because a complete theory does not yet exist for determining the performance of the systems when there is a change in its parameters or variables.

When some of these applications are designed for more complex systems, the number of fuzzy rules controlling the process is exponentially increased with the number of variables related to the system. For example, if there are n variables and m possible linguistic labels for each variable, m^n fuzzy rules would be needed to construct a complete fuzzy controller. As the number of variables n increases, the rule base quickly overloads the memory of any computing device, causing difficulties in the implementation and application of the fuzzy controller.

Sensory fusion and hierarchical methods are studied in an attempt to reduce the size of the inference engine for large-scale systems. The combination of these methods reduces more considerably the number of rules than these methods separately. However, the adequate fusion-hierarchical parameters should be estimated. In traditional techniques much reliance has to be put on the experience of the system designer in order to find a good set of parameters (Jamshidi, 1997).

Genetic algorithms (GA) are an appropriate technique to find parameters in a large search space. They have shown efficient and reliable results in solving optimization problems. For these reasons, in this work we present a method that has proved to estimate parameters for the rule base reduction method using GAs.

The chapter is organized as follows. Section 2 summarizes the principles of rule base reduction methods. In Section 3, the sensory-fusion method, the hierarchical method and the combination of these methods are described. Section 4 proposes the GA which allows us to automatically find the parameters in order to improve the complex fuzzy control system performance. Inverted pendulum and beam-and-ball complex control systems are described and results are presented in Section 5. Finally, Section 6 concludes this chapter.

2. Complex Fuzzy Control Systems

A system may be called large-scale or complex, if its order is too high and its model is nonlinear, interconnected with uncertain information flow such that classical techniques of control theory cannot easily handle the system (Jamshidi, 1997). As the complexity of a system increases, it becomes more difficult and eventually impossible to make a precise statement about its behavior. Fuzzy logic is used in system control and analysis design, because it shortens the time for engineering development and sometimes, in the case of highly complex systems, is the only way to solve the problem.

Principle components of a fuzzy controller are: a process of coding numerical values to fuzzy linguistic labels (*fuzzification*), inference engine where the fuzzy rules (expert operator's experience) are implemented and decoding as the output fuzzy decision variables (*defuzzification*). Fuzzy control can be implemented by putting the above three stages on a computer device (chip, personal computer, etc.).

From a control theoretical point of view, fuzzy logic has been intermixed with all the important aspects of systems theory – modeling, identification, analysis, stability, synthesis, filtering, and estimation. One of the first complex system in which fuzzy control has been successfully applied is cement kilns, which began in Denmark. Today, most of the world's cement kilns are using a fuzzy expert system. However, the application of fuzzy control to large-scale complex systems is not, by no means, trouble-free. For such systems the number of the fuzzy IF-THEN rules as the number of sensory variables increases very quickly to an unmanageable level.

When a fuzzy controller is designed for a complex system, often several measurable output and actuating input variables are involved. In addition, each variable is represented by a finite number m of linguistic labels which would indicate that the total number of rules is equal to m^n, where n is the number of system variables. As an example, consider $n = 4$ and $m = 5$ than the total number of fuzzy rules will be $k = m^n = 5^4 = 625$. If there were five variables, then we would have $k = 3125$. From the above simple example, it is clear that the application of fuzzy control to any system of significant size would result in a dimensionality explosion.

3. Rule Base Reduction Methods

One of the most important applications of fuzzy set theory has been in the area of fuzzy rule based system. Rule base reduction is an important issue in fuzzy system design, especially for real time Fuzzy Logic Controller (FLC) design. Rule base size can be easily controlled in most fuzzy modeling and identification techniques.

The size of the rule base of complex fuzzy control systems grows exponentially with the number of input variables. Due to that fact, the reduction of the rule base is a very important issue for the design of this kind of controllers. Several rule base reduction methods have been developed to reduce the rule base size. For instance, fuzzy clustering is considered to be one of the important techniques for automatic generation of fuzzy rules from numerical examples. This algorithm maps data points into a given number of clusters (Klawonn, 2003). The number of cluster centers is the number of rules in the fuzzy system. The rule base size can be easily controlled through the control of the number of cluster centers. However, for control applications, often there is not enough data for a designer to extract a complete rule base for the controller. A designer has to build a generic rule base. A generic rule base includes all possible combinations of fuzzy input values. The size of the rule base grows exponentially as the number of controller input variables grows. As the complexity of a system increases, it becomes more difficult and eventually impossible to make a precise statement about its behavior.

A simple and probably most effective way to reduce the rule base size is to use Sliding Mode Control. The motivation of combining Sliding Mode Control and Fuzzy Logic Control is to reduce the chattering in Sliding Mode Control and enhance robustness in Fuzzy Logic Control. The combination also results in rule base size reduction. However, this approach has its disadvantages as the parameters for the switch function have to be selected by an expert or designed through classical control theory (Hung, 1993).

Anwer (Anwer, 2005) proposed a technique for generation and minimization of the number of such rules in case of limited data sets. Initial rules for each data pairs are generated and conflicting rules are merged on the basis of their degree of soundness. The minimization technique for membership functions differs from other techniques in the sense that two or more membership functions are not merged but replaced by a new membership function whose minimum and maximum ranges are the minimum value of the first and maximum of the last membership function and bisection point of the two or more will be the peak of the new membership function. This technique can be used as an alternative to develop a model when available data may not be sufficient to train the model.

A neuro-fuzzy system (Ajith, 2001; Kasabov, 1998; Juang, 1998; Jang, 1993; Halgamuge, 1994) is a fuzzy system that uses a learning algorithm derived from, or inspired by, neural network theory to determine its parameters (fuzzy sets and fuzzy rules) by processing data samples. Modern neuro-fuzzy systems are usually represented as special multilayer feedforward neural networks (for example, models like ANFIS (Jang, 1993), FuNe (Halgamuge, 1994), Fuzzy RuleNet (Tschichold-German, 1994), GARIC (Berenji, 1992), HyFis (Kim, 1999) or NEFCON (Nauck, 1994) and NEFCLASS (Nauck, 1995)). A disadvantage of these approaches is that the determination of the number of processing nodes, the number of layers, and the interconnections among these nodes and layers are still an art and lack systematic procedures.

Jamshidi (Jamshidi, 1997) proposed to use sensory fusion to reduce a rule base size. Sensor fusion combines several inputs into one single input. The rule base size is reduced since the number of inputs is reduced. Also, Jamshidi (Jamshidi, 1997) proposed to use the combination of hierarchical and sensory fusion methods. The disadvantage of the design of hierarchical and sensory fused fuzzy controllers is that much reliance has to be put on the experience of the system designer to establish the needed parameters. To solve this problem, we automatically estimate the parameters for the hierarchical method using GAs.

3.1 Sensory Fusion Method

This method consists in combining variables before providing them to input of the fuzzy controller (Ledeneva, 2006b). These variables are often fused linearly. For example, we want to fuse two input variables y_1 and y_2 (see Figure 1). The fused variable Y will be calculated as $Y = ay_1 + by_2$. Here, it is considered that the input variables of the fuzzy controller are represented by $m = 5$ linguistic labels. Therefore, in this case, the number of rules will be thus reduced from 25 to 5. As we can observe, more variables has the fuzzy controller, more reduction can be obtained (see Figure 4).

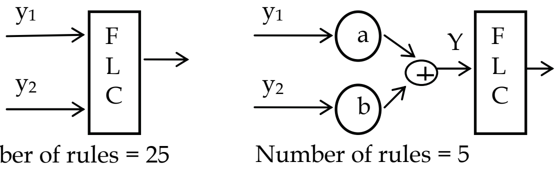


Fig. 1. Rule base reduction of sensory fusion fuzzy controller (when two variables are fused).

As another example, consider that a fuzzy controller has three inputs variables y_1 , y_2 and y_3. The total number of rules will be 125. In this case, we look into combining three variables in one of these four possible ways:

1. Variables y_1 and y_2 are fused in the new variables Y_1 and Y_2:

$$Y_1 = ay_1 + by_2$$
$$Y_2 = y_3$$

2. Variables y_1 and y_3 are fused in the new variables Y_1 and Y_2:

$$Y_1 = ay_1 + by_3$$
$$Y_2 = y_2$$

3. Variables y_2 and y_3 are fused in the new variables Y_1 and Y_2:

$$Y_1 = ay_2 + by_3$$
$$Y_2 = y_1$$

4. Variables y_1, y_2 and y_3 are fused in the new variable Y:

$$Y = ay_1 + by_2 +cy_3$$

The number of rules will be thus reduced by 125 to 25 if two variables are fused or from 125 to 5 if the three variables are combined.

The reduction of the number of rules is optimal if one can fuse all the input variables in only one variable associated. In this case, the number of rules is equal to the definite number of linguistic labels for this variable. But it is obvious that all these variables cannot be fused arbitrarily, any combination of variables has to be reasoned and explained. In practice only two variables are fused: generally the error and the change of error. The fusion can be done through the following rule

$$E = ae + b\Delta e \qquad (1)$$

where *e* and Δ*e* are error and its rate of change, *E* is the fused variable, and *a* and *b* found manually (Jamshidi, 1997).

We want to point out that the manually selection of the parameters *a* and *b* convert into fastidious routine. Below, we describe a new method (Ledeneva, 2006a), which permits to estimate these parameters automatically.

3.2 Hierarchical Method

In the hierarchical fuzzy control structure from (Ledeneva, 2007a), the first-level rules are those related to the most important variables and are gathered to form the first-level hierarchy. The second most important variables, along with the outputs of the first-level, are chosen as inputs to the second level hierarchy, and so on. Figure 2 shows this hierarchical rule structure.

IF y_1 is A_{1i} and … and y_1 is A_{1i} THEN u_1 is B_1
IF y_2 is A_{2i} and … and y_2 is A_{2i} THEN u_2 is B_2 (2)
…
IF y_{Ni+1} is A_{Ni1} and … and y_{Ni+nj} is A_{Ninj} THEN u_i is B_i

where *i*, *j* = 1, …, *n*; y_i are output variables of the system, u_i are control variables of the system, A_{ij} and B_i are linguistic labels; $N_i = \sum_{j=1}^{i-1} n_j \le n$ and n_j is the number of *j*-th level system variables used as inputs.

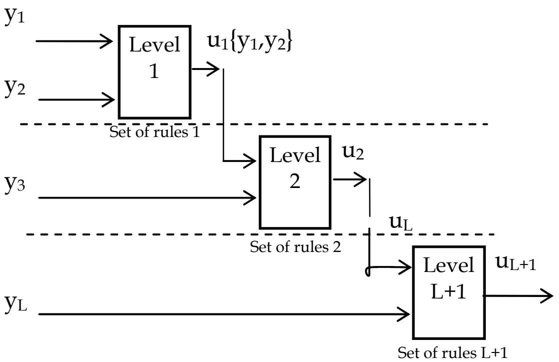


Fig. 2. Schematic representation of a hierarchical fuzzy controller.

The goal of this hierarchical structure is minimize the number of fuzzy rules from exponential to linear function. Such rule base reduction implies that each system variable provides one parameter to the hierarchical scheme. Currently, the selection of such parameters is manually done.

3.3 Combination of Methods

The more number of input variables of the fuzzy controller we have, the more it is interesting to combine the methods presented above with a goal to reduce more number of rules. We want to quote, as an example, the combination of the sensory fusion method (section 3.1) and the hierarchical method (section 3.2) for five variables as in Figure 3. Initially, the variables are fused linearly, as in Figure 1, and then are organized hierarchically according to a structure similar to that of Figure 2.

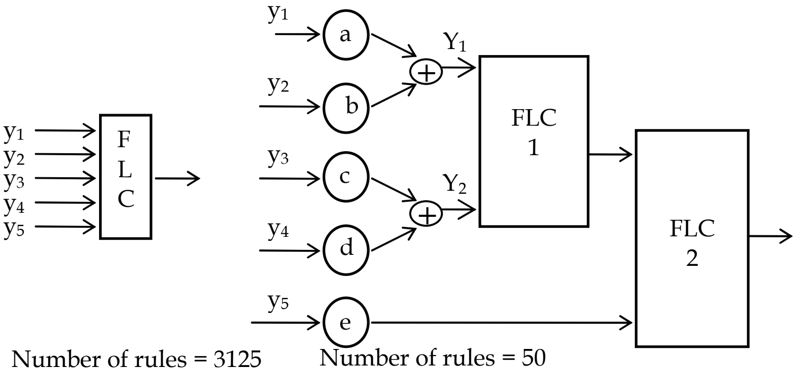


Fig. 3. Rule base reduction for the combination of sensory fusion and hierarchical methods (for n = 5 and m = 5).

The number of rules and the comparison of the sensory fusion method, the hierarchical method and the combination of these rule base reduction methods are presented in Table 1 and Figure 4 correspondingly. Take into account that the variables are fused here per pair and that on each level of the hierarchy one and only one variable is added. The most significant reduction can be obtained when the sensory fusion and hierarchical methods are combined (Ledeneva, 2007b).

Method used to reduce the number of rules	The number of variables $n > 1$	
	Even	Odd
Sensory Fusion	$m^{n/2}$	$m^{(n+1)/2}$
Hierarchical	$(n\text{-}1)\cdot m^2$	
Combination of methods	$((n/2)\text{-}1)\cdot m^2$	$((n+1)/2)\text{-}1$

Table 1. – The number of rules for the different reduction methods.

4. Genetic Optimization of the Parameters

Firstly, we give some basic definitions of GAs, than we present the proposed method to estimate the parameters of the sensory fusion method, the hierarchical method, and the combination of these rule base reduction methods.

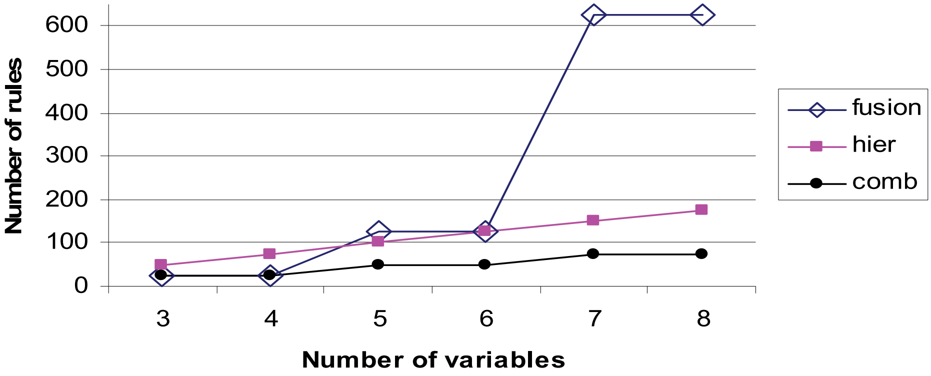


Fig. 4. Comparison of various rule base reduction methods with *m* = 5.

4.1 Step Response Characteristics

A fuzzy control system can be evaluated with the step response characteristics. We consider the following step response characteristics (see Figure 5):

Overshoot (%) is the amount by which the response signal can exceed the final value. This amount is specified as a percentage of the range of steps. The range of steps is the difference between the final value and initial values.

Undershoot (%) is the amount by which the response signal can undershoot the initial value. This amount is specified as a percentage of the range of steps. The range of steps is the difference between the final value and initial values.

Settling time is time taken until the response signal settles within a specified region around the final value. This settling region is defined as the step value plus or minus the specified percentage of the final value.

Settling (%) is the percentage used in the settling time.

Rising time is time taken for the response signal to reach a specified percentage of the range of steps. The range of steps is the difference between the final value and initial value.

Rise (%) is the percentage used in the rising time.

4.2 Genetic Algorithms

GA uses the principles of evolution, natural selection, and genetics from natural biological systems in a computer algorithm to simulate evolution (Goldberg, 1989). Essentially, the genetic algorithm is an optimization technique that performs a parallel, stochastic, but directed search to evolve the fittest population. GAs encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures so as to preserve critical information. GAs are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is quite broad. The more common applications of GAs are the solution of optimization problems, where efficient and reliable results have been shown. That is the reason why we will use these algorithms to find parameters for the rule base reduction methods.

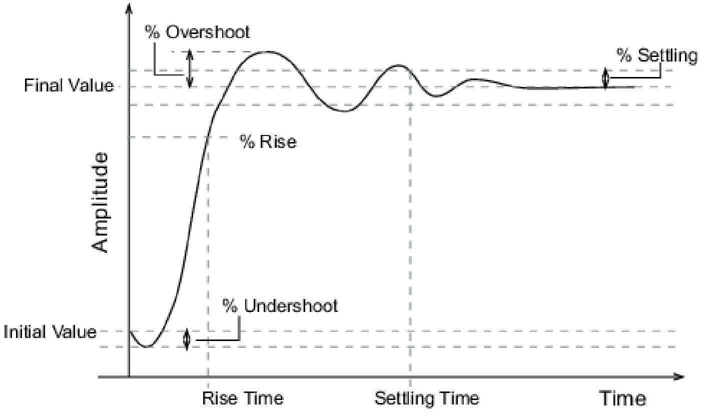


Fig. 5. Step response characteristics.

In the early 1970s, John Holland introduced the concept of genetic algorithms. His aim was to make computers do what nature does. Holland was concerned with algorithms that manipulate strings of binary digits. Each artificial "chromosome" consists of a number of "genes" and each gene is represented by 0 or 1:

0	0	0	0	1	1	1	1	1	1	1

Nature has an ability to adapt and learn without being told what to do. In other words, nature finds good chromosomes blindly. GAs do the same. Two mechanisms link a GA to the problem it is solving: encoding and evaluation. The GA uses a measure of fitness of individual chromosomes to carry out reproduction. As reproduction takes place, the crossover operator exchanges parts of two single chromosomes, and the mutation operator changes the gene value in some randomly chosen location of the chromosome.

4.2 Method for the Estimation of Parameters

The scheme of the proposed method is shown in Figure 5. We have three modules: System Module, Fuzzy Controller Module, and Genetic Algorithm Module. These three modules interconnect in two loops: an internal loop to control a system and an external loop to modify the fusion-hierarchical parameters. The internal loop comprises the fuzzy controller module and the system module. In other words, this loop represents a closed-loop control scheme. The external loop is composed of the genetic algorithm module, the fuzzy controller module, and the system module. The objective of the genetic algorithm module is to estimate the fusion-hierarchical parameters of the fuzzy controller through the minimization of the error between the design specifications and the output of the process.

Below we discuss each module of the proposed method.

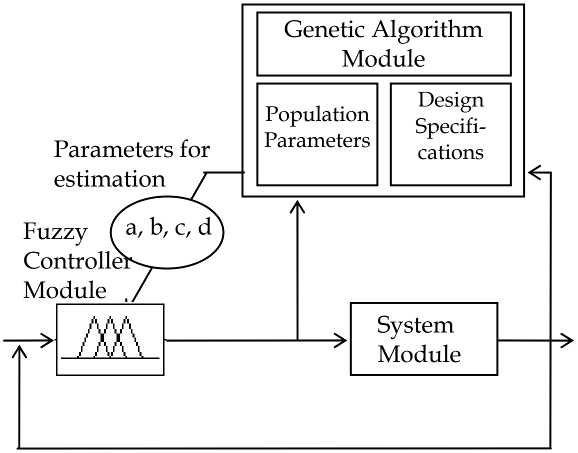


Fig. 5. Scheme of the proposed method.

4.2.1 Control System Module

The control system is defined as a complex system with p inputs and q outputs:

$$u = [u_1, \ldots, u_\mathrm{p}] \qquad (3)$$
$$y = [y_1, \ldots, y_\mathrm{q}]$$

4.2.2 Fuzzy Controller Module

The fuzzy controller module is represented by the fuzzy controller of reduced complexity which results after the application of the sensory fusion method, the hierarchical method, and the combination of these rule base reduction methods correspondingly such that it uses the combination of the fusion-hierarchical parameters.

Generally, the fuzzy controller is composed of one or several fuzzy controllers (depending on the number of variables). These controllers are of the Takagi-Sugeno type and each has a two inputs. The variation of these inputs results from the design of the sensory fusion method, the hierarchical method, and the combination of these methods; or the output variables of another fuzzy controller.

For example, let us describe general fuzzy controller with two input variables (see Figure 6) which are the vector of error $\varepsilon = y_d - y$ and variation of error $\Delta\varepsilon$, where y_d is the desirable system output. $K\varepsilon = [K\varepsilon, \ldots, \varepsilon_q]$ and $K\Delta\varepsilon = [K\Delta\varepsilon, \ldots, \Delta\varepsilon_q]$ are the gain input vectors. The output gain vector is noted as $Ku = [Ku_1, \ldots, Ku_q]$. The vector containing the resulting variables from the fusion module is noted as $X = [x_1, \ldots, x_q]$. So, for this example we have the output

$$X_i = K\varepsilon_i \cdot \varepsilon_i + K\Delta\varepsilon_i \cdot \Delta\varepsilon_I \qquad (4)$$

where $i = 1, \ldots, q$.

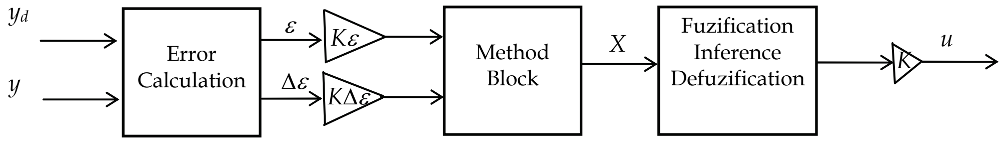


Fig. 6. General fuzzy controller structure.

4.2.3 Genetic Algorithm Module

Genetic Algorithm Module represents a genetic algorithm that maintains a population of chromosomes where each chromosome represents a combination of candidate parameters. This genetic algorithm uses data from the system to evaluate the fitness of each parameter in the population. The evaluation is done at each time step by simulating out with each combination of the parameters and forming a fitness function based on the design specifications which characterize the desired performance of the system. Using this fitness evaluation, the genetic algorithm propagates parameters into the next generation via the combination of the genetic operations proposed below. The combination of the parameters that is the fittest one in the population is used in the sensory fusion fuzzy controller.

This allows the proposed method to evolve automatically the combination of parameters from generation to generation (i.e., from one time step to the next, but of course multiple generations could occur between time steps), and hence to tune the combination of the parameters in response to changes in the system or due to user changes of the specifications in the fitness function of the GA.

The proposed procedure of estimating the combination of parameters by GA is summarized as follows:

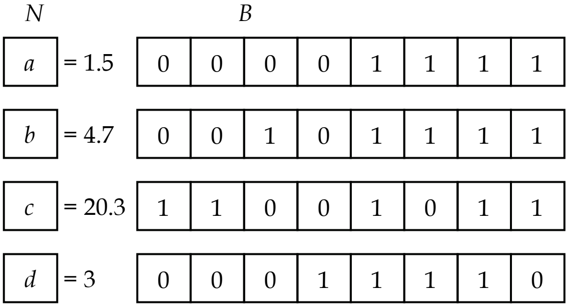
1. Determine the rule base reduction method and the number of parameters it is necessary to find.
2. Construct an initial population.
3. Encode each chromosome in the population.
4. Evaluate the fitness value for each chromosome.
5. Reproduce chromosomes according to the fitness value calculated in Step 4.
6. Create offspring and replace parent chromosomes by the offspring through crossover and mutation.
7. Go to 3 until the maximum number of iterations is reached.

4.2.3.1 Representation

To encode the combination of parameters, chromosomes of length $N \cdot B$ are used, where N is the number of parameters and B the number of bits which we use to encode the parameters. To decide how many bits to use for each parameter, we should consider the range of all possible values for each of them. For example, suppose that the parameters we want to obtain are positive with one decimal after the dot. To encode all possible values of each parameter we will use 8 bits. In Figure 7, there is one chromosome, representing the combination of parameters, which has $N = 4$ parameters with $B = 8$ bits each. So, the total range of the parameters will be in the interval [0, 256]. To obtain the required precision (one decimal after the dot), we multiply the output values of the parameters by 0.1. As a result, the searching parameters will be in the interval [0, 25.6].

4.2.3.2 Population

The initial population is randomly generated. Its size is fixed and equal to 50 individuals.



N		*B*							
a	= 1.5	0	0	0	0	1	1	1	1
b	= 4.7	0	0	1	0	1	1	1	1
c	= 20.3	1	1	0	0	1	0	1	1
d	= 3	0	0	0	1	1	1	1	0

Fig. 7. Example of representation of one chromosome (or one combination of parameters) which has $N = 4$ parameters with $B = 8$ bits each.

4.2.3.3 Fitness Function

The genetic algorithm maintains a population of chromosomes. Each chromosome represents a different combination of parameters. It also uses a fitness measure that characterizes the closed-loop specifications. Suppose, for instance, that the closed-loop specifications indicate that the user want, for a step input, a (stable) response with a rise-time of t^*_r, a percent overshoot of s^*_p, and a settling time of t^*_s. We propose the fitness function so that it measures how close each individual in the population at time t (i.e., each parameter candidate) is to meet these specifications. Suppose that t_r, s_p, and t_s denote the rise-time, the overshoot, and the settling time, respectively, for a given chromosome (we compute them for a chromosome in the population by performing a simulation of the closed-loop system with the candidate combination of the parameters and a model of the system). Given these values, we propose (for each chromosome and every time step)

$$J = w_1 (t_r - t^*_r)^2 + w_2 (s_p - s^*_p)^2 + w_3 (t_s - t^*_s)^2 \quad (4)$$

where $w_i > 0$, $i = 1, 2, 3$, are positive weighting factors. The function J characterizes how well the candidate combination of the parameters meets the closed-loop specifications; if $J = 0$ it meets the specifications perfectly. The weighting factors can be used to prioritize the importance of meeting the different specifications (e.g., a high value of w_2 relative to the other values indicates that the percent overshoot specification is more important to meet than the others).

Now, we would like to minimize J, but the genetic algorithm is a maximization routine. To minimize J with the genetic algorithm, we propose the fitness function

$$J_{res} = 1/J \quad (5)$$

Then, after knowing the design specifications of the system, and once we can obtain the step response characteristics for each chromosome in the population (rise-time, overshoot, and settling time), the fitness function is calculated in 2 steps:

1. We ask if the results coming from the GA is in the range of the design specifications of the system. If they are, we go to step 2. Else, the fitness value of this chromosome is set to 1000.
2. The fitness function is defined as described above (equations 4, 5).

4.2.3.4 Genetic Operators

In this section, we determine some genetic operators that we will use below (in Table 4).

Crossover: is a genetic operator that combines two chromosomes (parents) to produce one o two chromosomes (offspring). The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. First, the crossover operator randomly chooses a crossover point where two parent chromosomes "break", and then exchanges the chromosome parts after that point with a user-definable crossover probability. As a result, two new offspring are created (Melanie, 1999). The most common forms of crossover are one-point and two-point.

Mutation: represents a change in the gene. Its role is to provide and guarantee that the search algorithm is not trapped on a local optimum. The mutation operator uses a mutation probability denoted as p_m previously set by the user, which is quite small in nature, and it is kept low for GAs, typically in the range 0.001 and 0.01. According with this probability, the bit value is changed from 0 to 1 or vice versa (Melanie, 1999).

Elitism: copies the best individual (% of most fit individual) from the actual population to a new population and the rest of the new population is constructed according to the genetic algorithm.

Half Uniform Crossover (HUX): In this operator, bits are randomly and independently exchanged, but exactly half of the bits that differ between parents are swapped (see Figure 8). The HUX operator (Eshelman, 1991 ; Gwiazda, 2006) ensures that the offspring are equidistant between the two parents. This serves as a diversity preserving mechanism.

Truncation selection: implies that duplicate individuals are removed from population (Melanie, 1999).

In **roulette selection**: parents are selected according to their fitness. The better is the fitness, the bigger chance to be selected.

Parent A	1	1	0	1	1	0	1	1
Parent B	0	0	1	1	1	0	1	0
* Different Allels	*	*	*	1	1	0	1	*
x Allels to Interchange	x	*	x	1	1	0	1	*
Offspring A	0	1	1	1	1	0	1	1
Offspring B	1	0	0	1	1	0	1	0

Fig. 8. Example of Half Uniform Crossover.

5. Simulation Results

5.1 Inverted Pendulum System

The inverted pendulum control system (Messner, 1998; Aguilar, 2005; Aguilar, 2007) is used to test the proposed methods. The objective of this control system is, on one hand, to maintain the stem of the pendulum in high driving position, on the other hand, to bring the cart towards a given position x_0. The scheme in Figure 9 shows the main components of the system.

The basic variables are:

- the angular position of the stem θ;
- the angular velocity of the stem $\Delta\theta$;
- the horizontal position of cart x;
- the velocity of the cart Δx.

The design specifications of the inverted pendulum system are:

- the objective position of the cart is 30 cm;
- the overshoot of no more than 5 ;
- the settling time of no more than 5 sec.

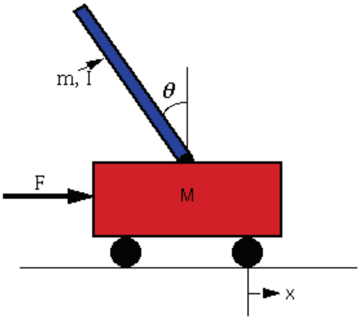


Fig. 9. Inverted pendulum, where $M = 1$ kg – mass of the cart, m =0.1 kg – mass of the pendulum, $l = 1$ kg – length to pendulum, F – force applied to the cart, x - cart position coordinate, θ - pendulum angle with vertical.

5.1.1 Design of the Sensory Fusion Method

The design of sensory fusion on a fuzzy controller is described in this section. First the sensory fusion of the input variables is done as follows:

$$\begin{cases} X_\theta = a\theta + b\Delta\theta \\ X_e = ce + d\Delta e \end{cases} \tag{6}$$

where a, b, c, and d are positive.

So, if X_e is null, that means that the cart reached its position of reference (e = 0 and Δe = 0), or that it moves towards this one (ce = -$d\Delta e$). Reasonably it is identical for X_θ. If X_θ is null, the angular position of the pendulum is stabilized to zero. Consequently, the stabilization of X_θ and X_e makes it possible to bring the pendulum towards a position of reference and ensure the maintenance of the stem of the pendulum in high driving position. The more absolute value of X_θ, more the horizontal position of the pendulum is critical. And the more absolute value of X_e, more the angular position of the pendulum is critical. The variables X_θ and X_e represent respectively the critical angular position and the critical horizontal position of the pendulum.

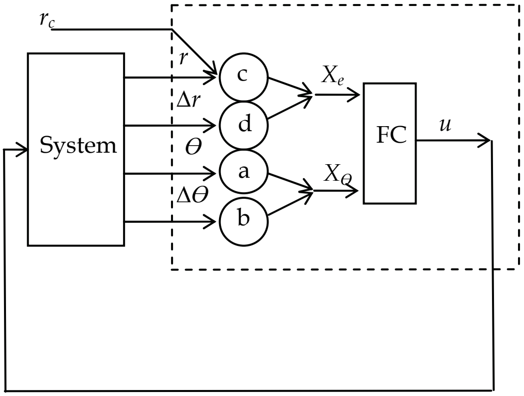


Fig. 10. Scheme of the sensory fusion fuzzy controller.

This control problem is far from being commonplace because two variables are to be controlled but only one has an action of control. The dynamics of θ being much faster than that of *r*, the adopted strategy is as follows: we initially seek to balance the pendulum (high driving position) then to gradually bring it towards its position of reference by unbalancing it on the "good side".

We can then write the five following rules in order to control the pendulum at the same time horizontally and vertically:

R1: IF X_θ is Negative THEN u is Negative,
R2: IF X_θ is Positive THEN u is Positive,
R3: IF X_θ is Zero y X_e is Negative THEN u is Negative, (7)
R4: IF X_θ is Zero y X_e is Zero THEN u is Zero,
R5: IF X_θ is Zero y X_e is Positive THEN u is Positive,

These five rules interpret well the priority objective which is the vertical stabilization of the pendulum: X_e is considered only when X_θ is null. Now let us examine the third rule (R3): "IF X_θ is Zero and X_e is Negative THEN u is Negative". This negative control involves X_θ positive. The second rule R2 is then activated: $u_s = ku$ becomes positive in order to balance the pendulum and the position of the cart increases as wished. The reasoning is similar for the fifth rule.

The five rules (7) can be written in a more compact way in the form of table (Table 2):

X_θ		N	Z	P
Xe	N	N	N	P
	Z		Z	
	P		P	

Table 2. Rule base of the sensory fusion fuzzy controller.

5.1.2 Design of the Hierarchical Method

The design of the hierarchical fuzzy controller is described in this section. The structure of the hierarchical fuzzy controller is represented in Figure 11.

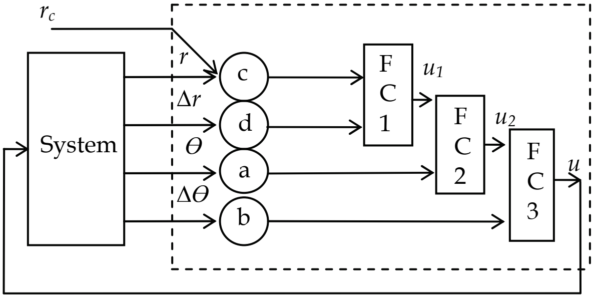


Fig. 11. - Hierarchical fuzzy controller.

The hierarchical fuzzy controller is composed of three fuzzy controller series connected. The corresponding rule bases are represented in Tables 3, 4 and 5. The total number of rules is 9 + 5 + 9 = 23 rules.

The objective of the first fuzzy controller (FC1) is to bring the cart towards its position of reference r_c. The first action u_1 consists in unbalancing the pendulum in the "good direction". This imbalance must have as a consequence the displacement of the cart in the desired direction.

$d\Delta e$		N	Z	P
ce	N	N	N	Z
	Z	N	Z	P
	P	Z	P	P

Table 3. Rule base of the FC1.

The objective of second fuzzy controller (FC2) is to balance the pendulum if this one is not so yet. The first decision of an action u_1 is preserved if the angular position of the pendulum is zero, but if the pendulum is not balanced, the new action u_2 is such as the pendulum converges towards a high driving position.

As for the third fuzzy controller (FC3) it aims to refine the preceding control by considering an additional variable $\Delta\theta$, the angular velocity of the pendulum.

If $\Delta\theta$ is zero, it does not have a reason there to modify the preceding control u_2. In the same way, if $\Delta\theta$ is negative (respectively positive) and u_2 is negative (respectively positive) then the preceding control does not have to be revised since it balances the pendulum. On the other hand, if the control u_2 is zero and the pendulum tends to be unbalanced then it is necessary to choose a control consequently.

5.1.3 Design of the Fusion-Hierarchical Method

The objective position where we must to bring a cart is x_o. The variables to fuse are θ and $\Delta\theta$, e and Δe, where e is the error in position given by $e = x - x_o$ and $\Delta e = \Delta x$. The sensory fusion of the error in position and its variation $X_e = ce + d\Delta e$ combined with the hierarchical method led to the fuzzy controller represented in Figure 12. The first fuzzy controller (FC1) calculates the first control action according to X_e and the angular position θ. In the second fuzzy controller (FC2), it refines the value of preceding control by considering an additional variable $\Delta\theta$. The fuzzy controller based on fusion-hierarchical combination is represented in the Figure 12. The rule bases of FC1 and FC2 are represented in Tables 6-7.

$a\theta$		N	Z	P
u_1	N	N	N	Z
	Z		Z	
	P		P	

Table 4. Rule base of the FC2.

$d\Delta\theta$		N	Z	P
u_2	N	N	N	Z
	Z	N	Z	P
	P	Z	P	P

Table 5. Rule base of the FC3.

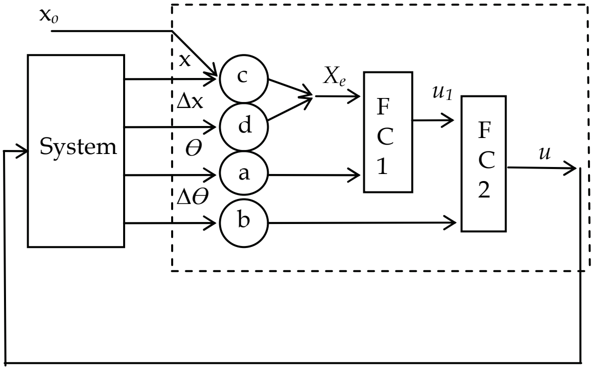


Fig. 12. Fuzzy controller based on the combination of the sensory fusion and hierarchical methods.

$a\theta$		N	Z	P
Xe	N	N	N	P
	Z		Z	
	P		P	

Table 6. Rule bases of the fuzzy controllers FC1.

$b\theta$		N	Z	P
u_1	N	N	N	Z
	Z	N	Z	P
	P	Z	P	P

Table 7. Rule bases of the fuzzy controllers FC2.

The simulation of the inverted pendulum is performed in *Simulink*, *Matlab* (Figure 13) starting from the nonlinear equations (Messner, 1998). The fuzzy controller is implemented in *Matlab´s FIS Editor*. The input fuzzy sets are represented by triangular functions (N, Z and P) regularly distributed on the universe of discourse [-1, 1]. The output fuzzy sets are singletons regularly distributed on [-1, 1].

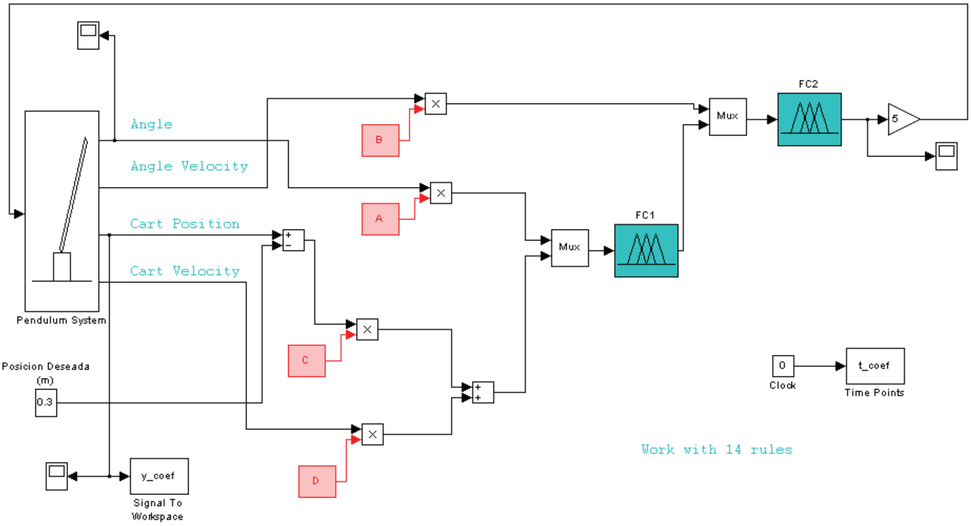


Fig. 13. Inverted pendulum control problem for the combination of methods implemented in Simulink.

5.1.4 Results

We apply the proposed method in order to find the parameters a, b, c, and d. The experiments were realized with the combination of some genetic operators in Table 8. The results of obtained parameters for each combination of genetic operators are presented in Tables 9-11. The best result is highlighted (Tables 9-11). The time response graphics are