Automatic Determination of Parameters for Rule Base Reduction of Complex Fuzzy Control Systems

Yulia Ledeneva,[1] Alexander Gelbukh,[1] Carlos A. Reyes García[2] and René A. García Hernández [3]

[1] Instituto Politécnico Nacional, Centro de Investigación en Computación, México
yledeneva@yahoo.com, gelbukh@gelbukh.com

[2] Instituto Nacional de Astrofísica, Óptica y Electrónica, México
kargaxxi@inaoep.mx

[3] Instituto Tecnológico de Toluca, México
reneamulfo@hotmail.com

Abstract. Fuzzy control methods follow the human way of making control decision. They are based on a usually large number of simple rules describing the reaction of the system under control to each combination of its control variables. Applied systems based on fuzzy control are of great importance in navigation of space vehicles, flight control, missile speed control, industry and manufacture, etc. In many cases, these are complex systems having many variables to control. For such systems, the rule base explodes exponentially in the number of variables. There are methods that considerably reduce the number of rules; however, the performance of such reduced system depends on the choice of some parameters, which have been so far determined only manually based on the experience and knowledge of a skilled system designer. We propose a method that uses a genetic algorithm to automatically determine these parameters for the combination of sensory fusion and hierarchical rule base reduction methods. The implementation process and simulation experiments are presented.

1 Introduction

Since the decade of the 80's, fuzzy logic has been the main source of practical and simple solutions for a great diversity of applications in engineering and science. Fuzzy control algorithms have been the most active area of research in fuzzy logic in the recent years. They are crucial in navigation systems, flight control, satellite control, speed control of missiles, as well as in industrial and manufacturing processes. Some fuzzy control applications to industrial processes produce results superior to those obtained by classical control methods. Moreover, these methods have serious limitations in their expanding to more complex systems, because

© S. Torres, I. López, H. Calvo. (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 27, 2007, pp. 37-48

Received 03/03/07
Accepted 08/04/07
Final version 22/04/07

currently there is no complete theory of how to predict the behavior of the system after a change in its parameters or variables.

Unlike the classical methods, fuzzy control methodology allows for simple and intuitive expansion to highly complex systems. However, for complex systems the number of fuzzy rules is increased exponentially in the number of variables describing the system: for n variables taking m possible linguistic labels each, m^n rules is needed to construct a complete fuzzy controller. As n grows, the rule base quickly becomes unmanageable. Various methods have been suggested to reduce the number of rules used by the controller.

A combination of sensory fusion and hierarchical methods is very effective in reducing the rule base. However, the popularity of these methods has been limited due to necessity for manually choosing quite a number of parameters: as many as there are variables. So far it has required tedious work and great experience of the system designer in order to find a good—even if not optimal—set of parameters [1].

In this paper, we propose a completely automatic method for choosing the very optimal parameters. Specifically, we use a genetic algorithm (GA) to choose their optimal combination. The paper is organized as follows. Section 2 introduces complex fuzzy control systems. In Section 3, the principles of rule base reduction methods are described. Section 4 proposes the GA-based algorithm that allows for automatically finding the optimal combination of parameters. Experimental results are presented in Section 5 and conclusions in Section 6.

2 Complex Fuzzy Control Systems

A system is complex if its order (the number of control variables) is too high and its model is nonlinear, interconnected with uncertain information flow, so that classical techniques of control theory cannot easily deal with such a system [1]. 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 simplifies engineering development; sometimes, in the case of highly complex systems, it is the only way to solve the problem.

The main components of a fuzzy controller are: a process of coding numerical values into fuzzy linguistic labels, an inference engine where the fuzzy rules are implemented, and decoding of the output fuzzy decision variables. Fuzzy control can be implemented by putting the above three stages on a chip or a personal computer.

Dealing with a complex fuzzy system remains a big challenge for any control paradigm to manage the number of the fuzzy IF-THEN rules. When a fuzzy controller is designed for a complex system, often several output and input variables are involved. In addition, each variable is represented by a finite number m of linguistic labels which 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 $k = m^n = 5^4 = 625$. For five variables, $k = 3125$. 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 Methods for Reduction of the Rule Base

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 [2]. The rule base size can be 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 rule base for the controller.

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 [3].

Anwer [4] proposed a technique for generation and minimization of the number of such rules in case of limited data sets availability. Initial rules for each data pairs are generated and conflicting rules are merged on the basis of their degree of soundness. 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 [5–9] 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 [8], FuNe [9], Fuzzy RuleNet [10], GARIC [11], HyFis [12] or NEFCON [13] and NEFCLASS [14]). 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 [1] proposed to use sensory fusion to reduce a rule base size. Sensory fusion combines several inputs into one single input. The rule base size is reduced since the number of inputs is reduced. Also, Jamshidi [1] 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. 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. So in this case, the number of rules will be thus reduced from 25 to 5. More variables has the fuzzy controller, more reduction can be obtained (see Figure 4).

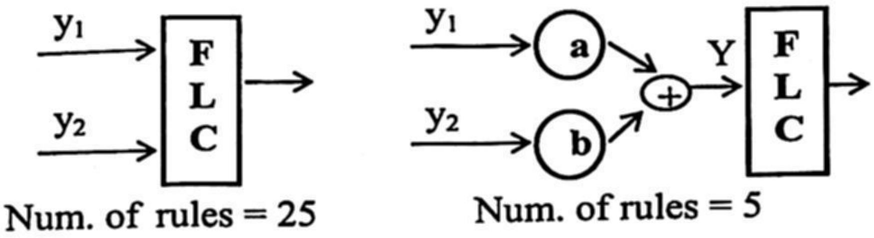

Fig. 1. Fuzzy logic controller's rule base reduction when two variables are fused.

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

$$E = ae + b\Delta e \tag{1}$$

where e and Δe are error and its rate of change, E is the fused variable, and a and b found manually [1].

We want to point out that the manually selection of the parameters a and b convert into fastidious and time-consuming routine. And the described method which permits to reduce significant the number of rules can't be used easily.

3.2 Hierarchical Method

In the hierarchical fuzzy control structure from [1], 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_n is A_{ni} THEN u_1 is B_1 **(2)**
IF y_{Ni+1} is A_{Ni1} and … and y_{Ni+nj} is A_{Ninj} THEN u_i is B_i,

where $i,j = 1, \ldots, n$; y_i are the control system's output variables, u_i are the system's control variables, 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.

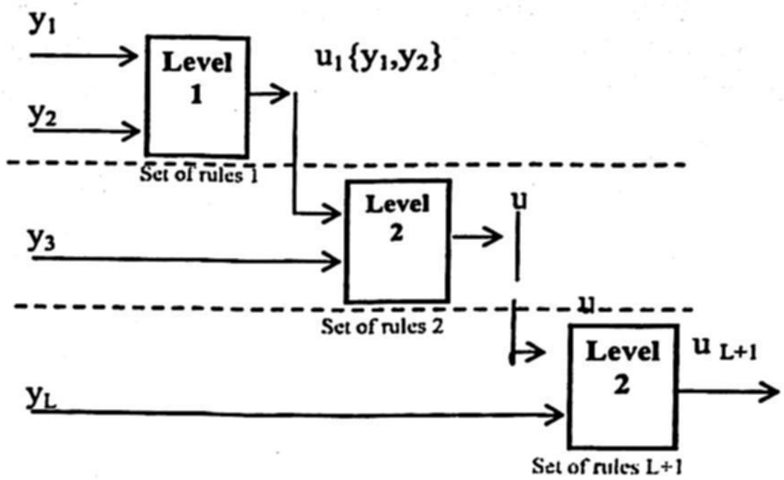

Fig. 2 Schematic representation of a hierarchical fuzzy controller.

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 done manually, which is a tedious and time-consuming.

3.3 Combination of the 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 the rule base. We want to quote, as an example, the combination of the sensory fusion method and the hierarchical method. The sensory fusion method (section 3.1) combined to the hierarchical method (section 3.2) led to an approach illustrated 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.

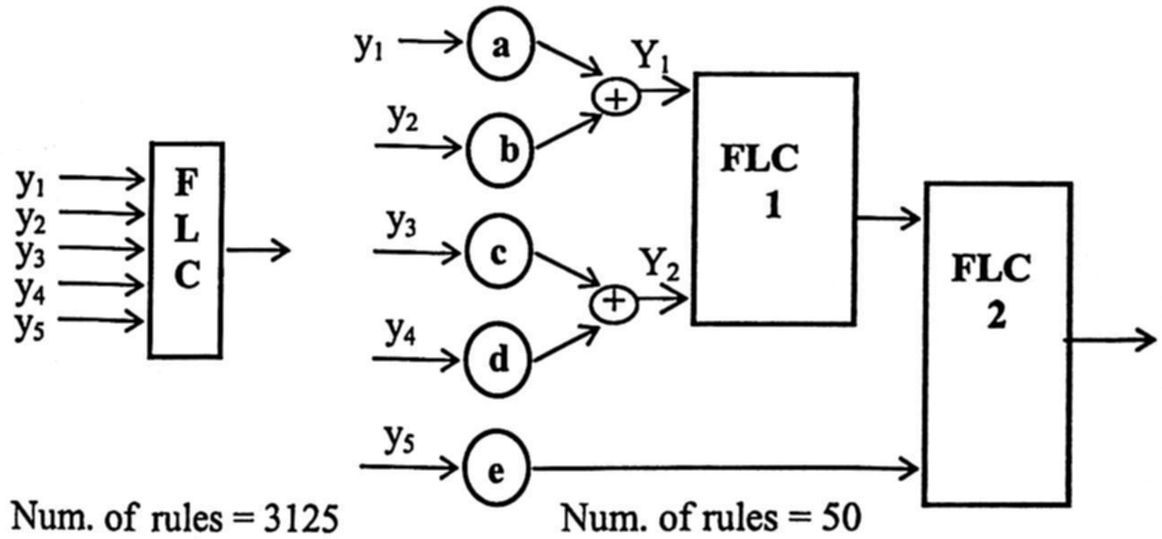

Fig. 3 Combination of the methods for $n=5$ and $m=5$.

The comparison of the sensory fusion method, the hierarchical method and the combination of these rule base reduction methods is presented in Figure 4. 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.

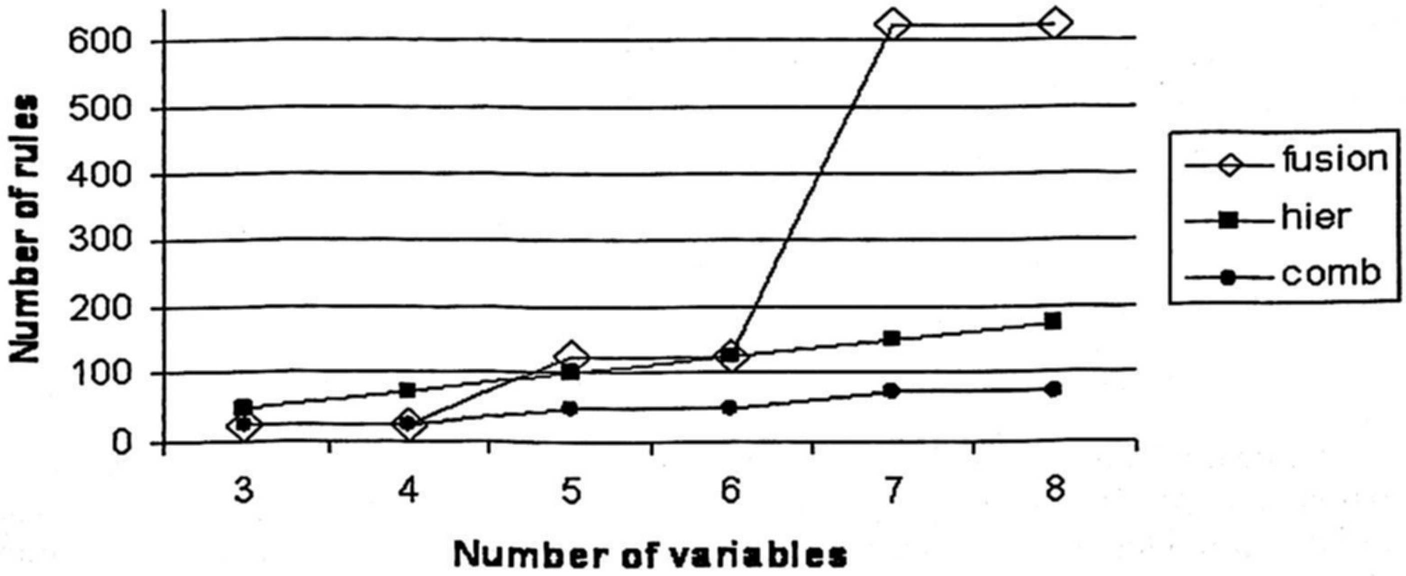

Fig. 4 Comparison of various rule base reduction methods with $m=5$.

4 Genetic Optimization

In this section, we present the proposed method to estimate the parameters of the combination of the sensory fusion and hierarchical rule base reduction methods. 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.

4.1 Control System Module

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

$$u = [u_1, \ldots, u_p];\ y = [y_1, \ldots, y_q]; \quad (3)$$

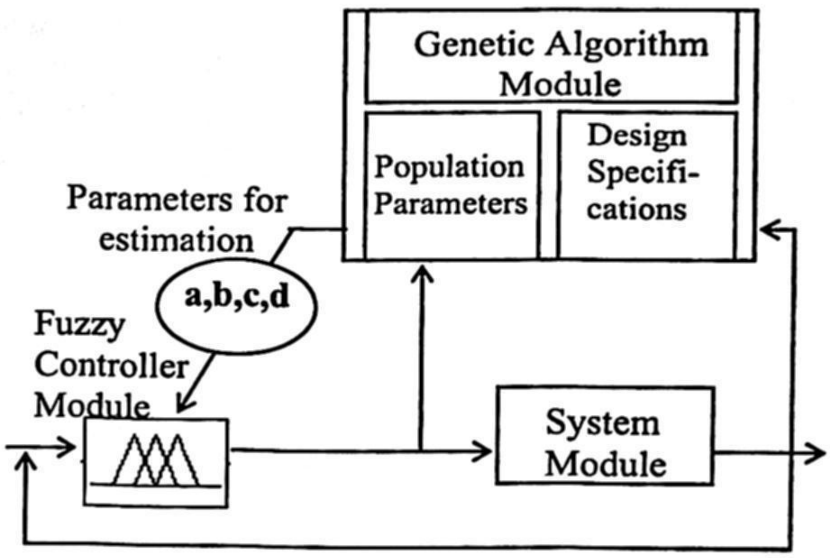

Fig. 5. Scheme of the proposed method.

4.2 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 and hierarchical rule base reduction methods (described above) such that it uses the combination of the fusion-hierarchical parameters.

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 maximum of two inputs. The variation of these inputs results from the design of the sensory fusion and hierarchical methods or the output variables of another fuzzy controller.

4.3 Genetic Algorithm Module

The Genetic Algorithm module represents a genetic algorithm that maintains a population of chromosomes where each of which represents a combination of candidate parameters. This genetic algorithm uses data from the system to evaluate the fitness of each parameter in the population. It does this evaluation 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.

We use a standard GA algorithm, which can be summarized as follows:

1. Determine the rule base reduction method and the number of parameters 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 the parent chromosomes by the offspring through crossover and mutation.
7. Go to 3 until the maximum number of iterations is met.

4.3.1 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 6, 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].

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

4.3.3 Fitness Function

The genetic algorithm maintains a population of chromosomes, each of which 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)

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

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

$$J_{res} = 1/J. \tag{5}$$

We know the design specifications of the system and we can obtain the step response characteristics for each chromosome in the population (rise-time, overshoot, and settling time). If the results given by the GA are in the range of the design specifications of the system, then the fitness function is defined by (4), (5); otherwise, it is set to 1000.

5 Simulation Results

The proposed method was tested in the inverted pendulum control system [15]. 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_o. The scheme in Figure 7 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.

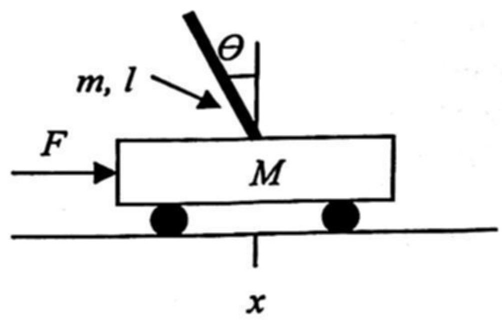

Fig. 7. Inverted pendulum, where $M = 1$ kg is the mass of the cart, $m = 0.1$ kg is the mass of the pendulum, $l = 1$ m is the length to pendulum, F is the force applied to the cart, x is the cart position coordinate, θ is the pendulum angle with vertical.

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.

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 a hierarchical method led to the fuzzy controller represented in Figure 8. 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 8. The rule bases of FC1 and FC2 are represented in Table 1.

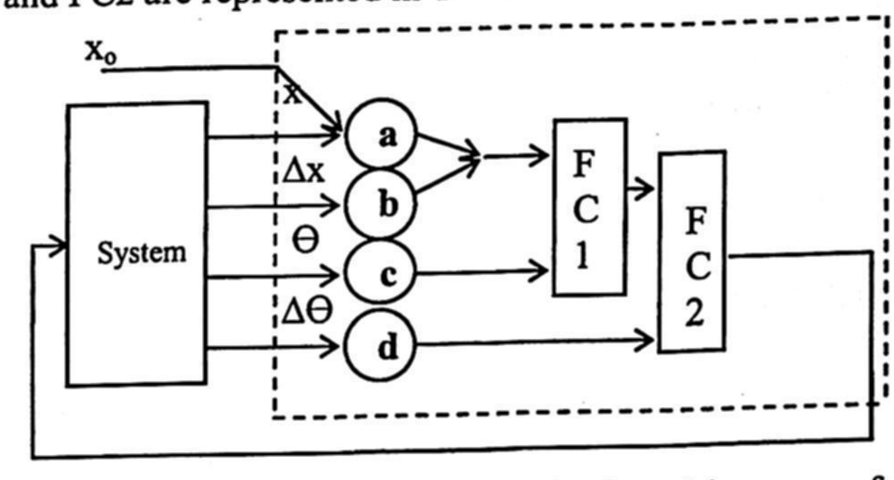

Fig. 8 Fuzzy controller based on the combination of the sensory fusion and hierarchical methods.

The simulation of the inverted pendulum is performed in *Simulink, Matlab* starting from the nonlinear equations [15]. The fuzzy controller is implemented in *Matlab 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].

Table 1. Rule bases of the fuzzy controllers FC1 (left) and FC2 (right).

X_e \ $a\theta$	N	Z	P
N		N	
Z	N	Z	P
P		P	

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

For the reduction with the combination of methods we obtained the following parameters: $a = 25.3$, $b = 10.1$, $c = 3.4$ and $d = 5.5$. With these parameters the horizontal position of the cart is stabilized in 5 seconds with overshoot equal to 0 (see Figure 9), and the behavior of the angle position of the stem of pendulum is shown in the Figure 10.

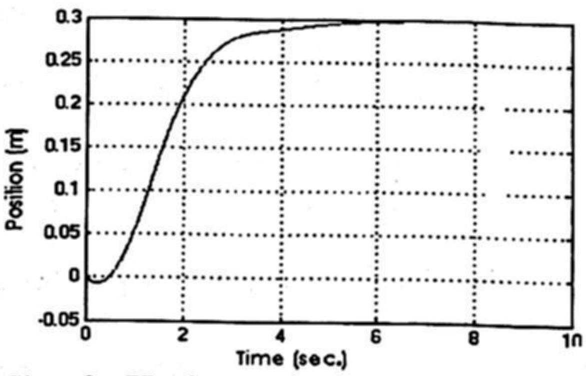

Fig. 9. Horizontal position of the cart.

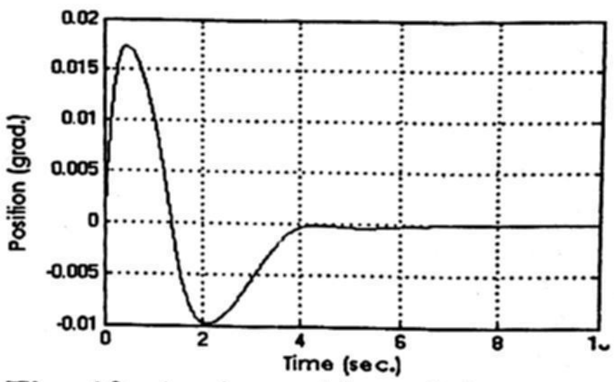

Fig. 10. Angle position of the stem of pendulum.

6 Conclusions

The combination of sensory fusion and hierarchical methods makes it possible to reduce the dimensionality of the control problem more significantly than any of the two methods separately. So far the parameters for such combination have been determined only manually, which requires months of tedious work of highly skilled expert. We suggested a method for automatically finding the optimal combination of such parameters. Our main contribution is the function to be optimized; then we use a genetic algorithm for optimization of this fitness function. We tested the proposed algorithm on a simulation of the inverted pendulum control problem and showed that the fusion-hierarchical parameters for the design specifications of this problem were adequately found.

Due to the fact that the fitness function is based on the design specification of the system, we can apply it to any combination of fusion-hierarchical variables. Another very important advantage is that when the user changes the design specifications, we can obtain the necessary fusion-hierarchical parameters very quickly by using the proposed GA. GA helps not only to automatically estimate the fusion-hierarchical parameters, but also to improve the results obtained by the combination of methods.

References

1. M. Jamshidi. Fuzzy Control Systems. *Springer-Verlag*, chapter of Soft Computing, pp. 42–56, 1997.
2. F. Klawonn, F. Höppner. What is Fuzzy About Fuzzy Clustering? -- Understanding and Improving the Concept of the Fuzzifier. In: M.R. Berthold, at al (Eds.): Advances in Intelligent Data Analysis. *Springer Berlin*, pp. 254-264, 2003.
3. Jonh Y. Hung, et. al. Variable Structure Control: A Survey. *IEEE Trans.on Industrial Electronics*, vol. 40, no.1, pp.2-21, 1993.
4. Zaheeruddin, Anwer M.J. A Simple Technique for Generation and Minimization of Fuzzy Rules. *IEEE International Conference on Fuzzy Systems*, CD-ROM Memories, Nevada, May 2005.
5. Abraham Ajith. Neuro Fyzzy Systems: State-of-the-art. Modeling Techniques. *Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence, Lecture Notes in Computer Science.* Springer-Verlag Germany, Jose Mira and Alberto Prieto (Eds.), Spain, vol. 2084, pp. 269-276, 2001.
6. N. Kasabov, R. Kozma, and W. Duch. Rule Extraction from Linguistic Rule Networks and from Fuzzy Neural Networks: Propositional versus Fuzzy Rules. *Proceedings of the Conference on Neural Networks and Their Applications NEURAP'98*, Marseilles, France, March, 1998, pp. 403-406, 1998.
7. Chia-Feng Juang, Chin-Teng Lin. An On-Line Self-Constructing Neural Fuzzy Inference Network and Its Applications. *IEEE Transaction on Fuzzy Systems*, vol. 6, No.1, pp. 12-32, February 1998.
8. Jyh-Shing Roger Jang, ANFIS: Adaptive-Network-Based Fuzzy Inference Systems, *IEEE Trans. System Man & Cybernetics*, vol. 23, pp. 665-685, 1993.
9. S. K. Halgamuge and M. Glesner. Neural networks in designing fuzzy systems for real world applications. *Fuzzy Sets and Systems*, vol. 65, pp. 1-12, 1994.
10. N. Tschichold-German. RuleNet - A New Knowledge--Based Artificial Neural Network Model with Application Examples in Robotics. PhD thesis, ETH Zerich, 1996.
11. H. R. Berenji and P. Khedkar. Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Trans. Neural Networks*, vol. 3, pp. 724-740, 1992.
12. Kim J., Kasabov N. Hy FIS: adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. Neural Networks, 12, pp. 1301-1319, 1999.
13. D. Nauck, R. Kruse. NEFCON-I: An X-Window Based Simulator for Neural Fuzzy Controllers.*IEEE-ICNN*, WCCI'94 in Orlando, 1994.
14. D. Nauck, R. Kruse. NEFCLASS - A Neuro-Fuzzy Approach for the Classification of Data. *Symposium on Applied Computing*, SAC'95 in Nashville, 1995.
15. Messner William C. and Tilbury Dawn M. Control Tutorials for *Matlab* and *Simulink*: A Web-Based Approach. Addison-Wesley, 1998.