Tuning of Fuzzy Logic Controller with Genetic Algorithm

Borut Zupančič, Marko Klopčič and Rihard Karba
University of Ljubljana, Faculty of Electrical Engineering, Tržaška 25, 1000 Ljubljana, Slovenia
Phone: +386 61 1768 417, Fax: +386 61 1264 631
E-mail: borut.zupancic@fe.uni-lj.si

Keywords: control system, fuzzy logic controller, optimization, genetic algorithm

Edited by: Anton P. Železnikar

Received: January 6, 1999 **Revised:** July 22, 1999 **Accepted:** September 13, 1999

The paper deals with the fuzzy logic controller (FLC) tuning by the aid of the optimization with genetic algorithm (GA). Because of the complexity of calculations the Sugeno 0th order FLC was used. The GA optimization tuned 25 consequent parameters while the membership functions remained fixed. Because of large number of parameters traditional optimization methods were not successful. Beside appropriately selected GA parameters the choice of appropriate reference signal of the control system was extremely important. Namely it must be selected so that the whole ranges of both FLC inputs are used. It is recommended to plot trajectory of FLC inputs to see which parts of the truth table is appropriately covered by the inputs and to find out which consequent parameters can not be optimized. Filtering of FLC characteristic is another useful method, which makes output characteristics smoother and so improves responses. The efficiency of the proposed approach were verified and validated on a hydraulic control system.

1 Introduction

The life cycle of a control system demands several optimizations in several steps. Probably the most demanding steps are those in the phases of process modelling and controller design. Sometimes these optimizations are very simple, based on experiences, tunning rules or simulation trials [3, 4]. Sometimes better result are obtained by conventional optimization techniques [1]. This approach is extremely important for control systems with lower number of loops and with so called parametric controllers, e.g. traditional PID controllers. Such optimization is extremely efficient when no more than app. 10 parameters must be tuned. However more advanced control algorithms contain usually much more parameters, which must be appropriately tuned. More complex control algorithms result in better efficiency, when systems are complex, nonlinear or time varying, multivariable, highly oscillating, with significant delays etc. Conventional optimization algorithms are not able to properly handle such problems, so there is a constant search for new and better methods. In this search researchers also started looking at how nature and people handle similar problems. Such thinking led to fuzzy logic and artificial neural nets as important elements for advanced control algorithms and genetic algorithms as new robust optimization techniques based on natural evolution.

One of the important advantages of Fuzzy Logic Controllers (FLC) in comparison with conventional linear controllers is that they provide an ability of non-linear control behaviour. However, the design of such a controller is not an easy task because there are many parameters, which are usually set with designer experiences or with simulation studies. Another approach is to use optimization. But FLC has several inputs (very usually two: for control error and its derivative) and each input has several membership functions. Beside there are many rules in the FLC knowledge base in which many so called consequent parameters appear.

As traditional optimization methods are too sensitive to the number of parameters, an advantage with optimization based on genetic algorithm (GA) was expected.

Modern tools, which were used for this study (MATLAB-SIMULINK, Fuzzy Logic Toolbox, Genetic Algorithm Toolbox [11]) give wide possibilities for efficient design and experimentation.

2 Description of genetic algorithm

GA [2], [6], [7] used for optimization has four standard operations:

- evaluation of individuals,
- reproduction,
- crossover and
- mutation

These operations are repeated until the terminating condition is met. In our case the optimization was stopped after a specified number of generations were evaluated. Table 1 shows the characteristic parameters of GA.

As a reproduction mechanism a method called deterministic roulette wheel was used [11]. Parameters were binary

number of generations	N_g
crossover probability	p_c
number of crossover points	N_c
mutation probability	p_m
number of individuals in gen.	N_p

Table 1: GA parameters

coded (with 12 bits). The most important parameter of the GA is fitness function, which is given by Eqs. (1), (2) and (3). Eq. (1) is well known criterion function often used in control systems design.

$$f = \int_0^{t_{max}} |e(t)|dt \tag{1}$$

$e(t)$ is the error between reference and controlled variable. Absolute error is used, as error is usually an oscillating signal. To get faster convergence, relative differences between fitness values for particular individuals are further increased by subtracting the minimal fitness in a generation k from the criterion defined with Eq. (1).

$$f_1(k) = f(k) - \min_{i=1\ldots N_p}(f(i)) + 1 \tag{2}$$

So the offset of the fitness function is removed. One is added to prevent the value zero of the fitness function $f_1(k)$. The differences between individuals are further increased by the transformation into final fitness function

$$f_2(k) = \left(\frac{f_1(k)}{\frac{1}{N_p}\sum_{i=1}^{N_p} f_1(i)}\right)^3 \tag{3}$$

Individuals with fitness above the average fitness get higher fitness value, while the ones below get lower fitness value. So with Eqs. (2) and (3) the relative differences between fitness values are greatly increased, enabling better individuals to have more offspring.

3 Description of the fuzzy logic controller

The controller used in experiments was Sugeno 0^{th} order type of FLC [14]. This type was chosen, because it is simpler in comparison with other types of FLCs also from the calculation complexity point of view. Namely each optimization performs many simulation runs and the CPU time needed for one simulation depends very much on the time needed for controller action evaluation. On the other hand its properties satisfied one of the important requirements - the ability to control non-linear process [14].

The FLC has usually two inputs x and y for control error and its derivative. Membership functions in our studies were equally spaced as it is shown in Figure 1. They were fixed during optimization.

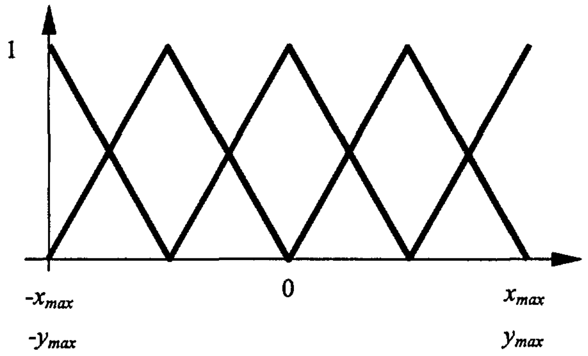

Figure 1: Membership functions of FLC input variables

Knowledge base of the FLC consisted of $P \times Q$ rules as is depicted in Figure 2.

1. IF $x = MF_{x1}$ AND $y = MF_{y1}$ THEN $u_{1,1} = C_{1,1}$
2. IF $x = MF_{x1}$ AND $y = MF_{y2}$ THEN $u_{1,2} = C_{1,2}$
3. IF $x = MF_{x1}$ AND $y = MF_{y3}$ THEN $u_{1,3} = C_{1,3}$

Q IF $x = MF_{x1}$ AND $y = MF_{yQ}$ THEN $u_{1,Q} = C_{1,Q}$
$Q+1$ IF $x = MF_{x2}$ AND $y = MF_{y1}$ THEN $u_{2,1} = C_{2,1}$

$P \times Q$ IF $x = MF_{xP}$ AND $y = MF_{yQ}$ THEN $u_{P,Q} = C_{P,Q}$

Figure 2: Knowledge base of FLC

x	the first input variable
y	the second input variable
MF_{xi}	i^{th} membership function of the first input variable
MF_{yi}	j^{th} membership function of the second input variable
u_{ij}	output of the $(i,j)^{th}$ rule
C_{ij}	consequent parameter
P	the number of membership functions of the first input variable
Q	the number of membership functions of the second input variable

The fuzzy logic operation and inference mechanism were realized as product [12].

$$u_{i,j} = C_{i,j}m_{xi}m_{yj} = C_{i,j}r_{i,j} \tag{4}$$

where

m_{xi}	membership grade of the first input
m_{yj}	membership grade of the second input
$r_{i,j}$	fulfilment of the $(i.j)^{th}$ rule, calculated as the product of membership grades

The controller output is calculated as the integral of the weighted average of consequent values

$$u(t) = \int_0^t \frac{\sum_{i=1}^{P}\sum_{j=1}^{Q} r_{i,j}C_{i,j}}{\sum_{i=1}^{P}\sum_{j=1}^{Q} r_{i,j}}dt = \int_0^t \frac{\sum_{i=1}^{P}\sum_{j=1}^{Q} u_{i,j}}{\sum_{i=1}^{P}\sum_{j=1}^{Q} r_{i,j}}dt \qquad (5)$$

With integration the fuzzy logic controller, which is actually a non-linear PD controller, was transformed into a PI one in order to eliminate steady state error.

4 Optimization

In our optimization study five equally spaced membership functions ($P = 5, Q = 5$) were used for each input variable (as shown in Figure 1). With optimization 25 consequent parameters $C_{i,j}$, each was coded with 12 bits, were determined. It is obvious that many troubles could be expected using conventional optimization techniques. The procedure is shown in Figure 3.

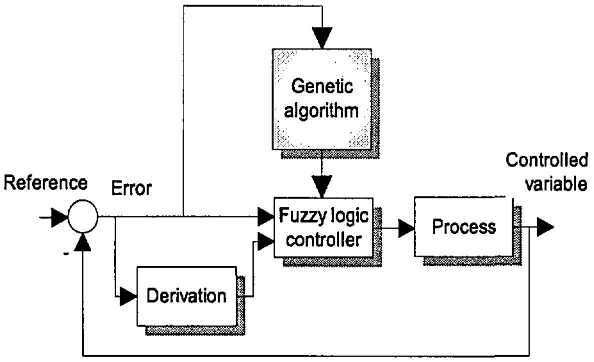

Figure 3: Optimization of fuzzy controller

FLC design is much more complex than linear controller design, because its output characteristic is non-linear. Output values for all possible input values have to be defined. The range of input values is divided into smaller intervals. The number of these intervals depends on the number of membership functions. Table 2 shows the truth table for control error e and its derivative de/dt. The fuzzyfied values are NB (negative big), NM (negative medium), ZE (zero), PM (positive medium) and PB (positive big).

Eq. 5 shows that control signal u is influenced only by those consequent parameters $C_{i,j}$ which have appropriate non zero membership grades $r_{i,j}$. In other words, only the rules with membership functions, which are defined on the domains of the current input variables are active. So it is obvious that the optimization of controller parameters $C_{i,j}$ is efficient only when the control error and its derivative cover the whole area defined by both variables during transient responses (simulation runs). Unfortunately as close loop system is optimized, it is not possible to select directly the appropriate signals $e(t)$ and $de(t)/dt$. Instead one has to select appropriately the reference signal. In Table 2 a typical trajectory of error and its derivative caused by a simple reference step change shows that with such input many consequent parameters can not be appropriately optimized. To overcome this problem, several types of reference signals were tested:

		e				
		NB_E	NM_E	ZE_E	PM_E	PB_E
de/dt	NB_{dE}	$C_{1,1}$	$C_{1,2}$	$C_{1,3}$	$C_{1,4}$	$C_{1,5}$
	NM_{dE}	$C_{2,1}$	$C_{2,2}$	$C_{2,3}$	$C_{2,4}$	$C_{2,5}$
	ZE_{dE}	$C_{3,1}$	$C_{3,2}$	$C_{3,3}$	$C_{3,4}$	$C_{3,5}$
	PM_{dE}	$C_{4,1}$	$C_{4,2}$	$C_{4,3}$	$C_{4,4}$	$C_{4,5}$
	PB_{dE}	$C_{5,1}$	$C_{5,2}$	$C_{5,3}$	$C_{5,4}$	$C_{5,5}$

Table 2: Trajectory of error and its derivative drawn on the FLC's truth table

- a signal consisting of sinusoidals with different amplitudes and frequencies,
- a signal consisting of steps with different amplitudes and delays,
- a square wave signal with increasing amplitude.

After substantial testing the square wave signal with increasing amplitude seemed to be the best solution. It is shown in Figure 4, together with control error for a typical example, while the appropriate trajectory is shown in Figure 5.

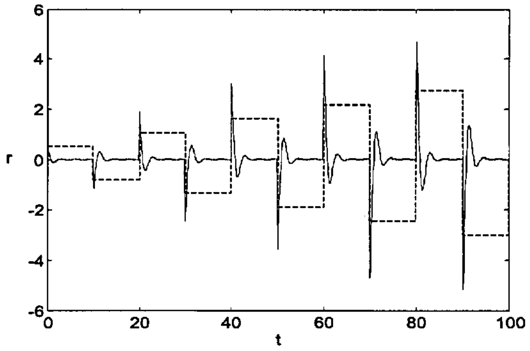

Figure 4: Reference signal (dashed line) and control error

As the whole phase area is fulfilled, it is possible to optimize all consequent parameters. However, in most cases GA finds solution close to the optimum, not the exact optimum. In our case this means, that some consequent parameters are slightly smaller than they should be, while the

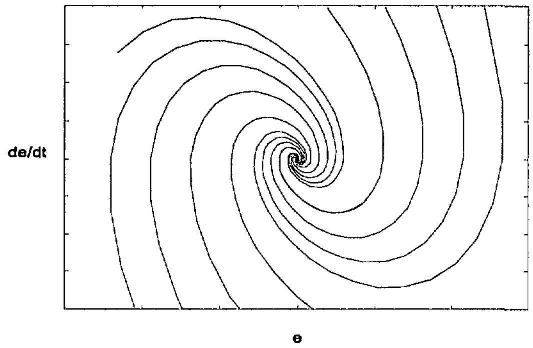

Figure 5: Trajectory in the error - derivative plane

others (perhaps the neighbours, see Table 2) are slightly larger. If the FLC output characteristics is considered as a non-linear function of two independent variables e and de/dt the surface is not very smooth as it has many local minima and maxima. Such surface can not assure the appropriate performance of the control system. As this inconvenience can be considered as a kind of noise introduced by stochastic features in GA, the idea to use a kind of filtering arose.

The idea of filtering is, to calculate the consequent parameter value by averaging in which the parameter itself and all parameter's neighbours are included (see Table 2)

$$C'_{i,j} = \frac{\frac{a}{4}(C_{i-1,j-1} + C_{i-1,j+1} + C_{i+1,j-1} + C_{i+1,j+1})}{a+b+c} + \frac{\frac{b}{4}(C_{i,j-1} + C_{i,j+1} + C_{i-1,j} + C_{i+1,j})}{a+b+c} + \frac{c \cdot C_{i,j}}{a+b+c} \tag{6}$$

where

$C'_{i,j}$ new (filtered) value of the consequent parameter

$C_{m,n}$ current values of consequent parameters ($m = i-1, i, i+1,\ n = j-1, j, j+1$)

a, b, c weights (parameters of the filter)

Using the filtering, the best results are obtained with optimization in several steps. After each step the filtering is used, what means that the new individuals are calculated from all individuals of the last generation of previous optimization step. Values of the filtering parameters a, b and c depend mostly on the type of process. There is no strict rule how to set them, but in our examples the starting values were set to 1 ($a = 1, b = 1 \mathrm{and} c = 1$). In some experiments b and c were intensified during optimization steps.

5 Experimental results: Optimization of the FLC controller of a hydraulic system

Our laboratory hydraulic set-up consists of three tanks and a main reservoir [5]. The transfer function which describes the relation between the input flow of liquid (incoming flow in the first tank) and the level in the third tank (controlled variable) is

$$G_P(s) = \frac{Y(s)}{U(s)} = \frac{1}{s^3 + 2s^2 + 3s + 1} \tag{7}$$

As mentioned FLC was Sugeno 0^{th} order with two inputs (error and its derivative), for each input five equally spaced membership functions were defined (Figure 1). Knowledge base was described with 25 rules (Figure2). The controller output was calculated with Eq.(5). Optimization with GA was used to calculate optimal values of 25 consequent parameters $C_{i,j}$, each was coded with 12 bits. GA selects the best controllers from the generation and performs other operations (crossover and mutation). Selection is made on the basis of fitness values that depend on the control error (see Eqs. (1), (2), (3)). The important parameters of GA are shown in Table 3.

number of generations	N_g	70
crossover probability	p_c	1
number of crossover points	N_c	3
mutation probability	p_m	0.01
number of individuals in gen.	N_p	30

Table 3: GA parameters

The overall scheme is shown in Figure 3.

The first generation of individuals was initialized with random numbers - no knowledge about the process was included. The optimal control system performance is shown in Figure 6.

The small oscillations are caused by rough FLC output characteristic which can be confirmed by Figure 7, where gray scale is used to denote the profile of the plane (darker means less).

After this study the filtering was introduced. Three optimization iterations were performed, each with 70 generations. After each optimization the filtering was used. After the first optimization and filtering the criterion function was 672, after the second 275 and after the third 262. As the change between the second and the third iteration was not significant, the iterative procedure was terminated. Other examples also confirm that three iterations are usually reasonable. Figures 8, 9 and 10 represent the optimal results (reference sp, controlled variable y and control signal u) after the first, second and the third iteration of optimization.

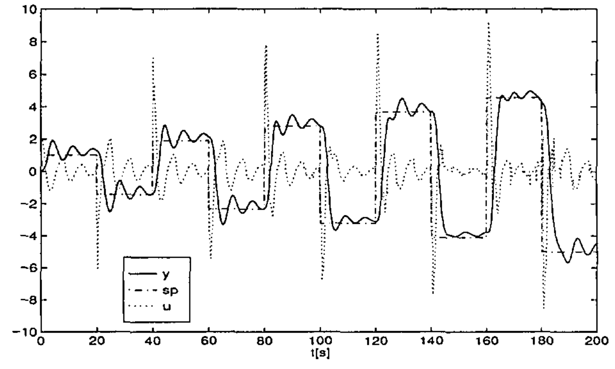

Figure 6: Optimal results (without filtering, y ... controlled variable, u ... control variable, sp ... reference)

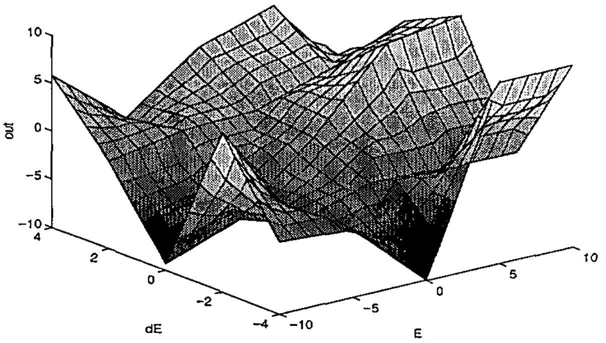

Figure 7: Output characteristic of the FLC optimized with GA (without filtering)

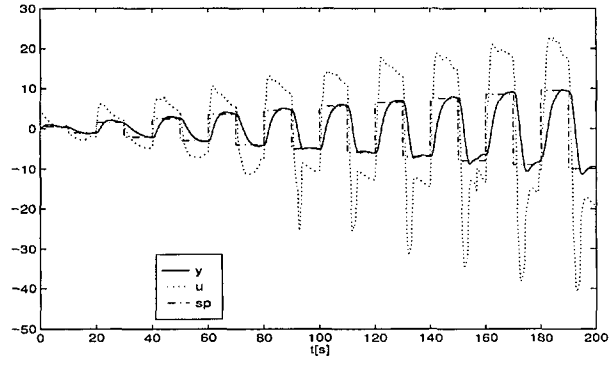

Figure 8: Results after the first iteration of optimization

It can be seen, that responses at rising and falling edges are not the same, because FLC characteristic is not symmetrical (linear). The response in Figure 10 is much better as it is fast and with a small overshoot. As the fitness function depends only on the control error, the values of controller signal u are very high in the points of reference change. If such values are unacceptable for actuator, the control variable u should be somehow included in the criterion function.

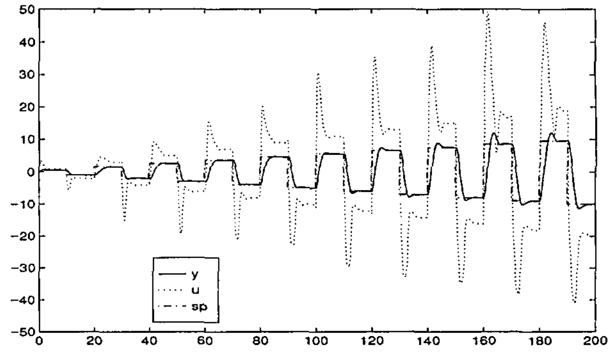

Figure 9: Results after the second iteration of optimization

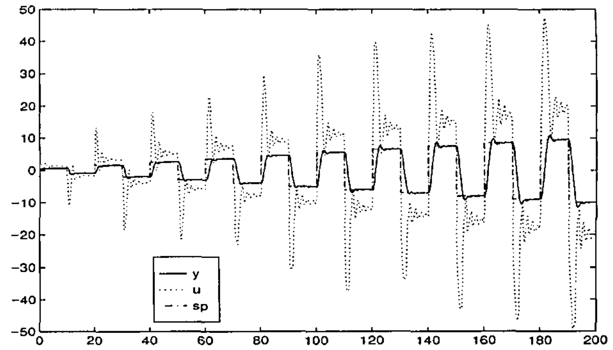

Figure 10: Results after the third iteration of optimization

Figure 11 depicts the output characteristic of the FLC. Using filtering it is much smoother.

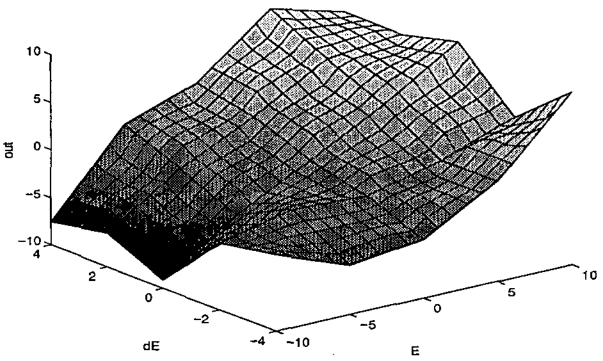

Figure 11: Output characteristic of the FLC optimized with GA (with filtering)

6 Conclusion

Genetic algorithms seem to be an efficient optimization approach in complex control systems with many tuning parameters. In fuzzy logic control systems there are many parameters, which can influence the behaviour: the number and the shape of membership functions, consequent parameters, etc. So conventional optimization techniques are usually not enough efficient or even unusable.

In our presented study the consequent parameters of FLC were optimized with GA. Experiences show that some of the parameters can have more or less random values after the optimization if some facts are not taken into account. To avoid such situation, appropriate type of reference signal should be used. It is also recommended to plot trajectory of FLC inputs (e.g. plane e, de/dt) to see which parts of the truth table is appropriately covered by the inputs and to find out which consequent parameters can not be optimized. Observation of FLC output characteristics is also useful because smooth shapes mean that the optimization produce at least near optimum values.

Filtering of FLC characteristic is another useful method, which makes output characteristics smoother and so improves responses. The procedure also decreases bad influence of parameters which are not satisfactory optimized, because their values get closer to the average of other optimized parameters.

However in the future more effort should be devoted to additional experiments with different reference or disturbance signals, which are more similar to shapes, which occur in reality. A lot of possibilities give also different types of FLCs as well as the study of the influence of different approaches in GA.

References

[1] Dixon L.C.W. (1972) *Nonlinear optimization.* The English universities press limited.

[2] Goldberg D. E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison Wesley Publishing Company.

[3] Åstrom. K. J. & Hägglund T. & Hang C. C. (1990), Automatic tuning and adaptation for PID controllers - a survey. *Automatica*, 20, p. 645-651.

[4] McMillan G. K. (1990) *Tuning and Control Loop Performance - A Practitioner's Guide (2nd ed.)*, Instrument Society of America, NC.

[5] Matko D. & Karba R. & Zupančič B. (1992) , *Simulation and Modelling of Continuous Systems: A Case Study Approach.* Prentice Hall Int., London.

[6] Beasley D. & Bull D. & Martin R. (1993) An Overview of Genetic Algorithms: Part 1, Fundamentals. *University Computing*, 15, 2, p. 58-69.

[7] Beasley D. & Bull D. & Martin R. (1993) An Overview of Genetic Algorithms: Part 2, Research Topics. *University Computing*, 15, 4, p. 170-181.

[8] Punch W. F. & Goodman E. D. & Min Pei & Lai Chia-Shun & Hovland P. & Enbody R. (1993) , Further research on Feature Selection and Classification Using Genetic Algorithms. *ICGA93*, Champaign III, p. 557-564 .

[9] Beasley D. & Bull D. & Martin R. (1993) A Sequential Niche Technique for Multimodal Function Optimization, Technical Report No. 93001. *University of Bristol*, Bristol, UK.

[10] Beasley D. & Bull D. & Martin R. (1993) , Reducing Epistasis in Combinatorial Problems by Expansive Coding. *Proc.* 5^{th} *International Conference on Genetic Algorithms*, Ed. S. Forrest, Morgan Kaufman, p. 400-407

[11] Prong van Hoogeveen J. W. D. (1995) *Optimization using Genetic Algorithms.* Ph.D. thesis, Technical university Delft. 3-14.

[12] Jager R. (1995) *Fuzzy Logic in Control.* Ph.D. thesis, Technical university Delft.

[13] Zhao J. &, Gorez R. & Wertz V. (1996) Genetic algorithms for the elimination of redundancy and/or rule contribution assessment in fuzzy models *Mathematics and Computers in Simulation*, 41, p. 139-148.

[14] Babuška and Verbruggen H. B. (1996) An Overview of Fuzzy Modelling for Control *Control Eng.Practice*, 4, 11, p. 1593-1606.