AUTOMATIC GENERATION OF FUZZY MODELS BY GENETIC ALGORITHMS

Elaine Yassue Nagai¹ Lúcia Valéria Ramos de Arruda²

^{1,2}Centro Federal de Educação Tecnológica do Paraná Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial Av. Sete de Setembro, 3165, Curitiba, Paraná, 80250-901, BRAZIL eynagai@cpgei.cefetpr.br, arruda@cpgei.cefetpr.br

Abstract: Genetic Algorithms (GA's) and others evolutionary optimization methods to design fuzzy models from data have received much attention in recent literature. This paper presents a methodology to simultaneous design of membership functions and rules sets for fuzzy logic models. The validity of the proposed identification method has been demonstrated by simulation example of inverted pendulum. *Copyright* © 2002 IFAC

Keywords: Fuzzy systems, fuzzy modeling, genetic algorithms, system identification, intelligent control.

1. INTRODUCTION

The design of mathematical models to complex realworld systems is essential in many fields of science and engineering. A common approach is the physical modeling, where everything is considered known a priori. While complete analytical knowledge is rare in complex technical environments, process measurements provide a powerful source of information about their dynamic behavior (Hollatz, 1997). Also, in many situations, a lot of information is available in qualitative form. These aspects lead to a situation of uncertain and inaccuracy where the physical modeling is not suitable.

The fuzzy logic provides an effective way to capture the approximate, inexact nature of real world (Lee, 1990). Therefore, instead of to design a model by means mathematical modeling, the fuzzy modeling defines a model directly from expert knowledge. However, there are situation where these expert knowledge is unavailable. In these cases, it is necessary to perform the fuzzy model identification automatically by process behavior observation.

In this paper, it is presented a fuzzy model identification methodology based on GA's, when no a priori knowledge about the system is available.

2. PROBLEM DESCRIPTION

This methodology considers, for multi-input singleoutput (MISO) systems, a linguistic fuzzy model consisting in a collection of m fuzzy rules in the form:

R_i : If x_1 is A_{i1} and ... and x_n is A_{in} Then y is B_i (1)

Where x_j , $(1 \le j \le n)$, are input variables and *n* is the number of input variables, *y* is the output variable, and A_{ij} , B $(1 \le i \le m)$, where *m* is the number of rules in the rule base, are the linguistic values of the input and output variables, respectively, in the i-th fuzzy rule.

The input and output variables take their values in a normalized universes of discourse [0,1]. The meaning (semantics) of linguistic values is characterized by membership functions $\mu_{Arj}(x_j)$ and $\mu_{Br}(y)$ defined on the universe of discourse [0,1]. In this paper, it is considered triangular membership functions. A computationally efficient way to characterize this type of membership function is the parametric representation achieved by means of a two-tuple (a_{rj}, b_{rj}) for input variables, and (a_r, b_r) for the output variable, where the first parameter is the center of the triangle base and the second one is the length of the triangle base, as is showed in figure 1.



Fig. 1- Triangular membership functions

The identification procedure is concerned with:

- The fuzzy model structure identification, which allows to obtain the rule base, and,
- The fuzzy model parameters estimation, which allows obtaining the database.

Considering the linguistic rule given by equation (1), the Zadeh's Compositional of Inference and the COA (center of area) defuzzification method (Lee, 1990) are used in the identified fuzzy model.

To apply this identification strategy, it is considered an input-output data set E_p . This data set is composed of *p* numerical input-output data tuples $e_l \in E_p$, called examples, each example having the form (for a system with *n* inputs and a output):

$$\mathbf{e}_{l} = (\mathbf{e}\mathbf{x}_{1}^{l}, ..., \mathbf{e}\mathbf{x}_{n}^{l}, \mathbf{e}\mathbf{y}^{l}), \ l = 1, ..., p$$
 (2)

In a conventional linguistic fuzzy model, the set of linguistic values taken by input and output variables is defined in advance. Furthermore, the meaning (semantics) of each linguistic value A_{ij} is determined by the membership function $\mu_{Aij}(.)$ and a same linguistic value may appear in a number of fuzzy rules. However, in every fuzzy rule in which this linguistic value appears it has the same semantics, i.e., the same membership function. This type of linguistic fuzzy model is called a descriptive linguistic fuzzy model. This paper does not take restrictions with regard to location and format of membership function. Thereby, it is considered a linguistic unconstrained free semantics fuzzy model (Cordón and Herrera, 1997).

3. STRUTUCTURE OF THE GENETIC FUZZY IDENTIFICATION

Two parts compose the genetic fuzzy identification method proposed herein:

- A module to fuzzy rules construction based on a real coded AG, and a covering method for the input-output example set. This part results in an initial fuzzy model structure and parameters;
- A module to fuzzy rule base reduction and simplification, where is verified the completeness of the fuzzy system.

3.1 Module to Genetic Construction of Fuzzy Rules

An important requirement to identify a fuzzy rule set R describing the structure of a linguistic fuzzy model is the completeness property. This property ensures that for any input in the input domain (universe of discourse), this system must be able to infer a suitable output, in other words, the system will cover all possible states in the input and output domains to fuzzy systems (Lee, 1990).

To ensure the completeness property of system it is necessary to define the cover value of a example $e_l \in E_p$ and the compatibility degree between a rule and an example, according to the following: Let the nonempty union of the membership functions $\mu_{Aij}(x_j)$ and $\mu_{Bi}(y)$, then:

$$\mu_{Ai}(ex^{l}) = \min(\mu_{Ai1}(ex^{l}_{1}), ..., \mu_{Ain}(ex^{l}_{n}))$$
(3)

$$R_i(e_l) = \min(\mu_{A_i}(ex^l), \mu_{B_i}(ey^l))$$
(4)

Where: $R_i(e_l)$ is the compatibility degree between the rule R_i and the example e_l ; l=1,...,p; $\mu_{Aij}(.)$ is the membership function to the *j*-th antecedent in *i*-th rule; $\mu_{Bi}(.)$ is the membership function of the *i*-th rule consequent.

Given a fuzzy rules set R, the covering value to an example e_l is defined as:

$$CV_R(e_l) = \bigcup_{i=1}^m R_i(e_l),$$
(5)

And it is required that:

$$CV_{R}(e_{I}) \geq \tau$$
, (6)

Where: *m* is the size of the rule base; τ is the minimum cover value; $CV_R(e_l)$ is the cover value of the example $e_l \in E_p$ in relation to the rule set *R*.

The fuzzy rules set must satisfy both of the conditions presented above, i.e., it has to ensure the completeness property and to have an adequate covering value.

The genetic construction of the fuzzy rules consists of a building method and a covering method, both working on a given set of examples. The building method is realized by means a GA encoding of a single fuzzy rule in each chromosome. The GA finds the best fuzzy rule in every run over the set of examples according the GA fitness function. The covering method is realized as an iterative process. It allows the construction of a fuzzy rules set such that they cover the set of examples. In each iteration, the construction method chooses the best chromosome (fuzzy rule), considers the relative covering value this fuzzy rule with respect to the example set, and removes all examples with a covering value greater than τ (equation (6)).

To verify the rule quality, it is used a variant of the well-known confidence factor (*CF*) measure. Let $A \rightarrow B$ denotes a rule, where *A* is the rule antecedent (a conjunction of conditions) and *B* is the rule consequent. The *CF* measure is simply $|A^{A}B|/|A|$, where |x| denotes the cardinality of set *x*. In other words, *CF* is the ratio of the number of examples that both satisfy the conditions in the rule antecedent and consequent over the number of examples satisfying the conditions in the rule antecedent (Quinlan, 1987).

A variant of this measure used in this paper is defined as:

$$\left(\left| A \wedge B \right| - 1/2 \right) / \left| A \right| \ge \delta \tag{7}$$

Where the operator \wedge is defined as a t-norm *min* with $\delta = 0.4$. The subtraction of $\frac{1}{2}$ factor from the numerator is to favor the discovery of more general rules, by avoiding the overfitting of the rules to data. (Arruda et al, 1999).

Generation of initial population. As mentioned before, a fuzzy rule will be a chromosome vector coded as a vector of real numbers. In the chromosome pool, each chromosome C_p , r=1,...,Q, Q= population size, represents a fuzzy rule as:

If
$$x_1$$
 is A_{r1} and ... and x_n is A_{rn} Then y is B_r (8)

Where the real vectors $(\mathbf{a}_{rj} \ \mathbf{b}_{rj})$, $(\mathbf{a}_r \ \mathbf{b}_r)$, shown in figure 1, are the parameter vectors of the membership functions $\mu_{Arj}(x_j)$ and $\mu_{Br}(y)$, respectively; C_r codifies these vectors as:

$$(\mathbf{a}_{r1} \ \mathbf{b}_{r1} \dots \mathbf{a}_{rj} \ \mathbf{b}_{rj} \dots \mathbf{a}_{rn} \ \mathbf{b}_{rn} \ \mathbf{a}_{r} \ \mathbf{b}_{r},$$
(9)

The initial chromosome pool is created partially, considering the example set E_p as follows:

- *t* chromosomes are created in this phase, where $t = \min(|E_p|, (Q/2)), Q =$ size of the population;
- It is randomly selected *t* examples *e_l* ∈ *E_p*, composing the set *E_t* and for each example of *E_t* is determined a chromosome belonging to the initial chromosome pool;
- Consider the example $e_l \in E_i$, and its component $ex_j^i \in [0,1]$, $\Delta ex_j^i = \min((ex_j^i 0), (1 ex_j^i))$. Let $\gamma(ex_j^i)$ be a random value in the range $[0, \Delta ex_j^i]$. The membership function will be: $(ex_j^i, 2\gamma(ex_j^i)) = (a_j^i, b_j^i)$, $a_j^i =$ mean, and $b_j^i =$ standard deviation. The procedure is the same for the remaining components of e_i ;
- The remaining *Q*-*t* chromosomes of the initial population are chosen at random, each chromosome in its respective interval;

Fitness function. The fitness function used in this GA is composed by three different criteria:

High frequency value: the frequency of a fuzzy rule, R_i , on the set of examples E_p , is defined as:

$$\Psi_{E_{p}}(R_{i}) = \frac{1}{p} \sum_{l=1}^{p} R_{i}(e_{l})$$
(10)

Where: $R_i(e_l)$ is the compatibility degree between R_i and e_l ; p is the size of E_p .

High average covering degree on positive examples: the set of positive examples for R_i with compatibility degree greater than or equal to *w* is defined as:

$$E_{w}^{+}(R_{i}) = \left\{ e_{i} \in E_{n} / R_{i}(e_{i}) \ge w \right\}$$
(11)

The average covering degree on $E_w^+(R_i)$ is defined as:

$$G_{w}(R_{i}) = \sum_{e_{i} \in E_{w}^{+}(R_{i})} \left(R_{i}(e_{i}) / n_{w}^{+}(R_{i}) \right)$$
(12)

Where:
$$n_w^+(R_i) = |E_w^+(R_i)|$$
 and $w \in [0, 1]$ (13)

Small negative example set: the set of the negative examples for R_i is defined as:

$$E_{w}^{-}(R_{i}) = \left\{ e_{i} \in E_{p} / R_{i}(e_{i}) = 0 \quad and \quad A_{i}(ex^{i}) > 0 \right\}$$
(14)

Where: $\mu_{Ai}(ex^{i}) = \min(\mu_{Ai1}(ex_{1}^{i}),...,\mu_{Ain}(ex_{n}^{i}));$

An example is considered negative for a fuzzy rule when it better matches some other fuzzy rule with the same antecedent (if-part), but a different consequent. The negatives examples are always considered on the complete training set of examples (in this case, the set E_p). Let $n_{R_i}^- = |E^-(R_i)|$, the penalty function on the negative examples set is:

$$g_{n}(R_{i}^{-}) = \begin{cases} 1, & se \quad n_{R_{i}}^{-} \le kn_{w}^{+}(R_{i}) \\ 1 \\ \frac{1}{n_{R_{i}}^{-} - kn_{w}^{+}(R_{i}) + \exp(1)} &, & otherwise \end{cases}$$
(15)

A number of negative examples per fuzzy rule without any penalty is tolerated, if this number isn't up to a percentage of the number of positive examples, $kn_w^+(R_i)$. This percentage is determined by the parameter $k \in [0, 1]$.

Thereby, the fitness function is defined in the equation (16) and the goal is to maximize the fitness function.

$$Z(R_i) = \Psi_{E_i}(R_i) \cdot G_w(R_i) \cdot g_n(R_i^-)$$
(16)

With this criterion, the AG looks for a fuzzy rule having a high frequency value, a high covering degree, and a small negative example set.

3.2 Genetic Algorithm Specification

The genetic algorithm toolbox (*gaot*) of Matlab® was used to implement this work.

Each chromosome represents a rule (like in the equation (8)), and each membership function is represented by a two-tuple. If the goal model's is the

fitting of a function with two inputs and an output, then six genes compose each chromosome, as is demonstrated in the equation (17):

$$C_i = (a_{i1} \ b_{i1} \ a_{i2} \ b_{i2} \ a_i \ b_i)$$
 (17)

The genetic operators are used to create new solutions based on existed solutions in the population. There are two basic kinds of genetic operators: crossover and mutation. The crossover operator combines the features of two parent chromosomes to form two similar offspring. It is applied at a random position with a probability P_c . The mutation operator arbitrarily alters one or more components of a selected chromosome in order to increase the population structural variability. Each position of each chromosome vector in the population undergoes a random change according to a probability defined by a mutation rate, the so-called mutation probability P_m . The genetic operators used were: non-uniform mutation with $P_m = 0.07$ (7% of population), and arithmetic crossover with $P_c = 0.6$ (60% of population). The selection procedure used was the ranking method with size equal $(Q)^{1/2}$.

3.3 Module of Fuzzy Rule Base Reduction and Simplification

The complexity of a rule base is determined by the number of rules, and the number of different fuzzy sets belonging to each systems variable (Roubos and Setnes, 2001).

The number of membership functions belonging to each variable is not determined a priori. It may result in concentric membership functions generating a high sized rule base. This implementation uses the similarity driven rule base simplification method, proposed in (Babuska et al., 1998) to reduce this rule base. A similarity measure is used to quantify the redundancy among fuzzy sets in the rule base. Similar fuzzy sets, representing compatible concepts, are merged in order to obtain a generalized concept. The new fuzzy set replaces the similar ones in the rule base. This reduces the number of different fuzzy sets (linguistic terms) used in the model. The similarity measure is calculated as:

$$S(A,B) = \frac{|\min(A,B)|}{|\max(A,B)|} = \frac{\sum_{k=1}^{d} (\min(\mu_A(x_k),\mu_B(x_k)))}{\sum (\max(\mu_A(x_k),\mu_B(x_k)))}$$
(18)

Where |.| denotes cardinality of a set. The membership functions $\mu_A(x_k)$ and $\mu_B(x_k)$ are defined on a discretized universe of discourse $D = \{x_k \mid k = 1, 2, ..., d\}$, where *d* is number of discrete values.

The simplification algorithm merges similar fuzzy sets by use of a threshold: $\eta \in (0,1)$. For each variable, it is calculated the membership functions similarity

measure, *s*, and the membership functions with $s \ge \eta$ are merged. The rule base is updated by replacing the new membership function for the ones merged. The new membership function parameters are the similar membership functions parameters mean. Then, the new membership parameters are adjusted to cover all the universe of discourse.

This merging process needs a modification in the fuzzy rule representation. Now, a membership function is represented by a vector with three elements: (**a b c**), as in figure 2.



Fig. 2 – The new membership function representation

Similarity driven rule base simplification differs from other reduction methods in that its main objective is to reduce the number of fuzzy sets used in the model. It does not necessarily alter the number of rules. Reduction of the number of rules might follow from rule base simplification if the rules become equal as a result of the merging process. If no rules are combined, simplification is still achieved by reducing the number of fuzzy sets (Babuska et. al., 1998).

The number of generated rules is high due to the size of the examples set E_p , and it may exist redundant fuzzy rules, i.e., fuzzy rules with the same antecedent part and different consequent. To solve this conflict, it is defined a true degree to each one of these conflicting rules. The rule with highest degree is kept and the others are eliminated. The result is a more compact rule base with the completeness property kept. The true degree of a rule is defined as follows (Babuska et. al., 1997):

$$TrueDegree(R_i) = \prod_{l=1}^{p} R_i(e_l)$$
⁽¹⁹⁾

Where: $R_i(e_l)$ is the compatibility degree between R_i and e_l ; p is the size of E_p .

4. IDENTIFICATION ALGORITHM

Let E_p the training example set, the genetic fuzzy identification method is composed by the following steps:

Step 1: Establish the training data set and the test data set, where these sets are disjunct. It is used 1/3 of available example data to training and 2/3 to the testing;

Step 2: Run the module to fuzzy rules construction; **Step 2.1:** While $|E_p| \neq 0$ do the steps 2.2 to 2.6: **Step 2.2:** Run the GA. The resulting individual will be a candidate rule to belong the rule base;

Step 2.3: Calculate the *CF* (see the equation (7)) of the candidate rule;

Step 2.4: If the *CF* of this rule is greater than δ , then this rule is inserted into the rule base;

Step 2.5: Calculate the covering value $CV_R(.)$ (see the equation (5) and equation (6)), of this new rule;

Step 2.6: Remove all examples $e_l \in E_p$ with covering value greater than a priori defined constant τ ;

Step 3: Run the module of fuzzy rule base reduction and simplification (steps 3.1 to 3.4):

Step 3.1: Modify the rule representation;

Step 3.2: Apply the similarity driven rule base simplification (see the equation (18));

Step 3.3: Line the membership functions to ensure the completeness property in each universe of discourse;

Step 3.4: Calculate the true degree (equation (19)) of conflicting rules and remove the inconsistent ones;

Step 4: Verify the error rate in the use of obtained fuzzy model. It is made with the testing data set;

Step 5: If the error rate is lower than the acceptable, then is shut up, and the rule base and database are kept.

5. EXPERIMENTAL RESULTS

This section describes the obtained results with the developed algorithm applying to inverted pendulum example, described by Hui et al. (1993).

Let consider a fuzzy model with three inputs (y(t-1), u(t-1), u(t-2)) and an output (y(t)). Then, the goal is to identify the membership functions describing these four variables, and the rules that describe the system behavior. For this, 201 samples from the system were collected, and 1/3 of these samples was used to training and 2/3 to testing. The performance index used to result validation are:

$$MSE = \frac{1}{p} \sum_{l=1}^{p} (y(l) - y_{f}(l))^{2}$$
(20)

$$NRMSE = \frac{p}{\sum_{l=1}^{p} y(l)} \cdot \sqrt{MSE}.100\%$$
(21)

Where: MSE is the mean-square error between the expected and obtained (fuzzy) output, *NRMSE* is the normalized root mean-square error between the expected and fuzzy outputs, p is the number of data, y(l) is the expected output and $y_f(l)$ is the fuzzy output.

5.1 Obtained Fuzzy Model

The obtained results are shown in the tables 1, 2, and 3. The table 1 contains the error description (MSE and NRMSE), the number of rules in the rule base (r), and the number of membership functions, to each

system variable (*nmi1*), (*nmi2*), (*nmi3*) and (*nmo*), for the input 1, input 2, input 3 and the output, respectively.

Table 1 - Obtained Results

mse	nrmse	r	nmi1	nmi2	nmi3	nmo
0.0011	5.23%	8	4	2	2	4

Table 2 presents a numerical description of the universe of discourse partition for each variable in the model. The first column contains the fuzzy set linguistic identification, the second one assigns each variable to a membership, and the third one presents the parameters of these sets.

Table 2 – Universes of Discourse Partition

£				
luzzy set	variable belonging	set parameters		
E11	y(t-1)- input 1	[-0.0900, 0.010, 0.1129]		
E12	y(t-1) - input 1	[0.0108, 0.1600, 0.6450]		
E13	y(t-1) - input 1	[0.1600, 0.6450, 0.9100]		
E14	y(t-1) - input 1	[0.8737, 0.9100, 1.2000]		
E21	u(t-1) - input 2	[-0.2222, 0, 1.0000]		
E22	u(t-1)- input 2	[0, 1.0000, 1.2634]		
E31	u(t-2) - input 3	[-0.2518, 0, 1.0000]		
E32	u(t-2) - input 3	[0, 1.0000, 1.2093]		
S1	y(t) - output	[-0.0343,0.0200,0.2800]		
S2	y(t) - output	[0.0200, 0.2800, 0.4695]		
S 3	y(t) - output	[0.4164, 0.6400, 0.8500]		
S4	y(t) - output	[0.6400, 0.8500, 1.200]		

The obtained rule base is showed in table 3. Each line corresponds to a rule with three antecedents (y(t-1), u(t-1), u(t-2)) and a consequent (y(t)).

Table 3 – Obtained Rule Base

E11	E22	E31	S1
E11	E22	E32	S1
E12	E22	E32	S2
E13	E21	E31	S3
E13	E21	E32	S3
E13	E22	E31	S3
E13	E22	E32	S 3
E14	E22	E32	<u>S</u> 4

Figures 3, 4, 5, and 6 present the universe of discourse partition of each model variable.



Fig. 3 – Membership functions to input variable 1 (y(t-1)) before(a) and after (b) the reduction and simplification phase



Fig. 4 – Membership functions to input variable 2 (u(t-1)) before(a) and after (b) the reduction and simplification phase



Fig. 5 – Membership functions to input variable 3 (u(t-2)) before(a) and after (b) the reduction and simplification phase



Fig. 6 – Membership functions to output variable (y(t)) before(a) and after (b) the reduction and simplification phase

The figure 7 shows the obtained fuzzy model output by the identification algorithm (continuous line) and the expected output (doted line).



Fig. 7 – Obtained fuzzy output by the simplified rule base

6. CONCLUSIONS

Model identification is very difficult and nontrivial in high-dimensional nonlinear problems. So a tool that make possible the use of both expert knowledge and input-output data is a good solution for a lot of complex problems. This paper has presented a fuzzy model identification methodology with this characteristic. Two parts compose the proposed method: a module based on GA, that generates the fuzzy model rule base, and a rule base reduction and simplification module that ensures the obtained fuzzy model completeness property. This is an advantage of the method: it allows to manage the trade-off between fuzzy model accuracy and complexity in accordance with the user choices.

The method validity was verified by inverted pendulum simulated identification. The obtained model reproduces adequately the system behavior. However, it is still necessary to realize comparatives studies with another fuzzy identification methods to verify the method validity.

7. REFERENCES

- Arruda, L. V. R.; Fertig, C. S.; Freitas, A. A. and Kaestner, C. A. (1999). Fuzzy beam-search rule induction algorithm. 3rd European Conference on Principles and Practice of Knowledge Discovery in Database, Praga, Rep. Tcheca, pp. 341 – 347.
- Babuska, R.; Kaymak, U.; Lenke, H. R. N. and Setnes, M. (1998). Similarity Measures in Fuzzy Rule Base Simplification. *IEEE Trans. on* Systems, Man, and Cybernetics – Part B, Vol. 28, pp.376-386.
- Cordón, O. and Herrera, F. (1997). Identification of linguistic fuzzy models by means of genetic algorithms. In H. Helledoorn e D. Driankov (Eds.), *Fuzzy Model Identification: Selected Approach*, pp. 215 – 250, ed. Springer Verlag, Germany.
- Hollatz, J. (1997). Fuzzy identification using methods of intelligent data analysis. In H. Helledoorn e D. Driankov (Eds.), *Fuzzy Model Identification: Selected Approach*, pp. 165 – 191, ed. Springer Verlag, Germany.
- Hui, S.; Kuschewski, J.G. and Zak, S.H. (1993). Application of feedforward neural networks to dynamical system identification and control. *IEEE Trans. on Control Systems Technology*, Vol. 01, pp. 37 – 49.
- Lee, C. C. (1990). Fuzzy logic in control systems: fuzzy logic controller, Part I and II. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 20, pp.404-435.
- Quinlan, J. R. (1987). Generating production rules from decision trees. *Proc. Int. Joint Conf. AI* (*IJCAI*-87), pp. 304 – 307.
- Roubos, H. and Setnes, M. (2001). Compact and transparent fuzzy models and classifiers through iterative complexity reduction. *IEEE Trans. on Fuzzy Systems*, Vol. 09, pp. 516-524.