A Genetic-Algorithm Based Approach For Generating Fuzzy Singleton Models

MIGUEL RAMIREZ
Universidad de los Andes
Facultad de Ingeniera
La Hechicera - Merida
VENEZUELA
Ramirezmb@pdvsa.com

ELIEZER COLINA
Universidad de los Andes
Facultad de Ingeniera
La Hechicera - Merida
VENEZUELA
Ecolina@ula.ve

Abstract: Methods for generating fuzzy singleton models from input-output data have been proposed by many authors. This paper introduces a genetic algorithm (GA) based method to generate a fuzzy singleton model taking into account all the necessary constraints to guarantee an analytically inverted representation of the process dynamics which may be used as a fuzzy controller in Internal Model Control (IMC) strategy. A major advantage of this sort of models is its high interpretability compared to first-order Takagi-Sugeno fuzzy models generated from fuzzy clustering techniques [15]. The proposed method is applied to a liquid level control problem in an oil production separator based upon real input-output data, where obtaining an adequate fuzzy model is of crucial importance.

Key–Words: Genetic algorithm, internal model control, fuzzy inverse control

1 Introduction

Inverse fuzzy model based control has been widely used in many applications and its succesful implementation depends on the exactness of the fuzzy model obtained from input - output data [1][2][3][4]. As a matter of fact, the simplest way to control a process using a fuzzy model is to invert it and use it as a controller but it is imperative to have an ideal model in order to guarantee a perfect control, which is very difficult to achieve because an exact inversion of the process can only be found in special situations and the model is never identical to the process, resulting in model-plant mismatches.

In general, the output of a singleton fuzzy model is linear in the consequent parameters. This fact allows a straightforward application of standard recursive least-squares algorithms for estimating the consequent parameters from data. In this sense, a formal methodology was proposed in [5] based on least squared techniques. Fuzzy singleton models obtained from this methodology have a good generalization characteristic and acceptable exactness while keeping a high interpretability.

In order to enhance the exactness of this sort of models which facilitates calculating its inverse, the use of genetic algorithms (GA) is proposed in this work. Genetic algorithms are randomized search algorithms that are based on the mechanics of natural selection and genetics [6][7]. They combine the principles of natural selection based on the survival of the fittest with a randomized information exchange in order to form a search and optimization algorithm. A significant number of papers have proposed different methods to build a fuzzy rule base using GA [8][9][10][11].

Although genetic algorithms can be used for a variety of purposes, their most important application is in the field of optimization, because of their ability to seek efficiently e in large no convex search spaces, which makes them more suitable with respect to more conventional optimization techniques. In fact in many engineering fields, optimization is the basic concept behind the application of genetic algorithms (GAs) or any other evolutionary algorithm.

This paper is organized as follows. Problem formulation is stated in section 2, where all the constraints of fuzzy singleton models that guarantees its exact inversion are described. The generation of the singleton fuzzy model using GA will be presented in section 3. In section 4, the fuzzy singleton model and its inverse are applied in Internal Model Control strategy. Finally, conclusions are given in section 5.

	$u(\tau)$			
$x(\tau)$	U_1	U_2	. . .	U_M
A_1	c_{11}	c_{12}	. . .	c_{1M}
A_2	c_{12}	c_{22}	. . .	c_{2M}
⋮	⋮	⋮	⋱	⋮
A_N	c_{N1}	c_{N2}	. . .	c_{NM}

Figure 1: Antecedents and Consequents of the fuzzy singleton model

2 Problem Formulation

It is important to know the characteristics of the fuzzy singleton model. A general fuzzy singleton model has the following form:

$$\textbf{If } x(\tau) \text{ is } A \textbf{ and } u(\tau) \text{ is } U \textbf{ Entonces } y(\tau+1) = c \tag{1}$$

Following the procedure in [13], let N denote the number of different fuzzy sets A_i defined for the state $x_(\tau)$ and M the number of different fuzzy sets U_j defined for the input $u(\tau)$. If the rule base consists of all possible combinations of A_i and U_j then the rule base is complete and the total number of rules is $K = N * M$. The entire rule base may be represented as a table which is showed in Fig 1.

The logical connective and is assumed to be represented by the product t-norm operator. This is a necessary condition to perform the inversion and the degree of fulfillment of the rule antecedent $\beta_{ij}(\tau)$ is calculated as:

$$\beta_{ij}(\tau) = \mu_{A_i}(x(\tau)).\mu_{U_j}(u(\tau)) \tag{2}$$

Where $\mu_{A_i}(x(\tau))$ is the membership degree of a particular state $x(\tau))$ in the fuzzy set A_i and $\mu_{U_j}(u(\tau))$ is the membership degree of an input $u(\tau)$ in the fuzzy set U_j. The predicted output $\hat{y}(\tau+1)$ of the model is computed by the fuzzy-mean defuzzification, where an average of the consequents c_{ij} is weighted by the degrees of fulfillment β_{ij}, so that:

$$\hat{y}(\tau+1) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\beta_{ij}(\tau)c_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{M}\beta_{ij}} \tag{3}$$

The rule base of equation (3) corresponds to a nonlinear regression model expressed as:

$$\hat{y}(\tau+1) = f(x(\tau), u(\tau)) \tag{4}$$

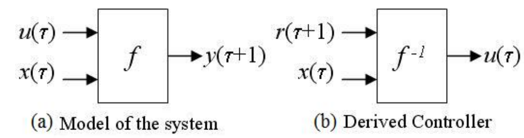


Figure 2: Fuzzy model and a controller based on the model inverse

Which is shown schematically in Fig 2-a., where, the model inputs are the current state $x(\tau)$ and the current input $u(\tau)$ and the output is the system's predicted output at the next sampling instant $\hat{y}(\tau+1)$. Given the current system state $x(\tau)$ and the desired system output reference at the next sampling time $r(\tau+1)$, in order to calculate the inverse of the model, it is neccesary to find $u(\tau)$, such that the system output $y(\tau+1)$ is as close as possible to the desired output $r(\tau+1)$. This can be achieved by inverting the plant model, as indicated in Fig 2-b., substituting the reference $r(\tau+1)$ for $\hat{y}(\tau+1)$ in the following static function:

$$u(\tau) = f^{-1}(x(\tau), r(\tau+1)) \tag{5}$$

This tecnique was proposed in [12]. Equation (4) can be reduced to the univariate mapping $\hat{y}(\tau+1) = f_x(u(\tau))$ by making use of the model structure. The subscript x denotes that f_x is obtained for the particular state $x(\tau)$. If the model is invertible, the inverse mapping $u(\tau) = f_x^{-1}(r(\tau+1))$ can be obtained. The concept of invertibility and the respective conditions for the fuzzy model are related to the monotonicity of the model's input-output mapping. A fuzzy model f given by the rule base from equation (1) and the defuzzification method of equation (3) is invertible if $\forall x$ and $\forall y$, a unique u exists such that $y = f(x, u)$. In terms of the parameters of the model, the monotonicity is translated into the following conditions:

$$card(core(U_j)) = 1, \forall j = 1, ..., M, \text{and} \tag{6}$$

$$core(U_1) < ... < core(U_M) \rightarrow c_{i1} < c_{i2} < ... < c_{iM}, \text{or}$$

$$core(U_1) > ... > core(U_M) \rightarrow c_{i1} > c_{i2} > ... > c_{iM}, \tag{7}$$

With $i = 1, ..., N$. Here, $card(.)$ denotes the cardinality of a set.

Theorem 1 Inversion of the singleton fuzzy model. *Let the process be represented by the singleton fuzzy model of equation(1) with the weighted-mean defuzzification method from equation (3). Furher, let the antecedent membership functions form a partition, i.e.,*

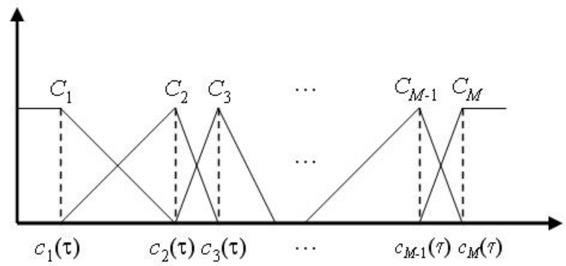


Figure 3: Partitions of fuzzy sets C_j buit using the cores c_j.

let $\sum_{i=1}^{N} \mu_{A_i}(x) = 1, \forall x$, *and* $\sum_{j=1}^{M} \mu_{U_j}(u) = 1, \forall u$. *At a certain time* τ *the system is at the state* $x(\tau)$*, and the inverse of the singleton model is given by the fuzzy rules:*

$$\textbf{If } r(\tau+1) \textit{ is } C_j(\tau) \textbf{ Then } u(\tau) = U_j \qquad (8)$$

Where C_j *are fuzzy sets which form partitions as in Fig 3. The cores* $c_j(\tau)$ *of the fuzzy sets* C_j *are given by:*

$$c_j(\tau) = \sum_{i=1}^{N} \mu_{A_i}(x(\tau))c_{ij}, j = 1, ...M. \qquad (9)$$

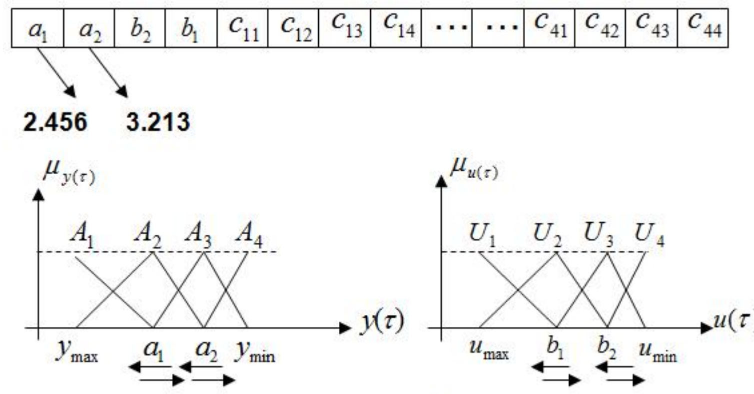
The inference and defuzzification of the rules in equation (8) is accomplished by the fuzzy-mean method:

$$u(\tau) = \sum_{j=1}^{M} \mu_{C_j}(r(\tau+1)).core(U_j) \qquad (10)$$

3 Generation of a Fuzzy Singleton Model For An Oil Production Separator Using GA

In this section, based on a set of 750 input-output data set of an oil production separator, a genetic algorithm is applied to generate a fuzzy singleton model that can satisfy two important requirement: it fullfills the constraints established in equations (6) and (7), thus the inversion is guaranteed and the index of performance of the resultant fuzzy model must be less than 0,05 (measured by the root mean squared).

The use of genetic algorthm requires the determination of six fundamental issues: chromosone representation, the fitness function, the genetic operators making up the reproduction function, the creation of the initial population, termination criteria and the evaluation function.



$y(\tau)$	U_1	U_2	U_3	U_4
A_1	c_{11}	c_{12}	c_{13}	c_{14}
A_2	c_{11}	c_{12}	c_{13}	c_{14}
A_3	c_{11}	c_{12}	c_{13}	c_{14}
A_4	c_{11}	c_{12}	c_{13}	c_{14}

Figure 4: Chromosone structure, Membership functions and fuzzy rules.

Chromose representation: For any GA, a chromosone representation is needed to describe each individual in the population of interest. Each individual or chromosone is made of a sequence of genes from a certain alphabet. An alphabet could consist of binary digits (0 and 1), floating point numbers, integers, symbols, matrices, etc. In Holland's original design, the alphabet was limited to binary digits. However, problem representation has been a subject of much investigation. One usefull representation of a chromosone for function optimization involves genes or variables from an alphabet of floating point numbers with values withing the varaibles upper and lower bounds. Michalewicz in [14] has done extensive experimentation comparing real-valued and binary digits representations and shows that the real-value representation is an order of magnitude more efficient in terms of CPU time. In this paper, a floating-point (real-valued) representation chromosone is used as depicted in figure 4.

The structure of the chromosone is composed of twenty parameters: a_1 and a_2 are the cores of the membership functions A_2 and A_3 respectively (the cores of A_1 and A_4 are y_{min} and y_{max} which belongs to the minimum and maximum value of $y(\tau)$ taken from the set of data), b_1 and b_2 are the cores of the membership functions U_2 and U_3 respectively (the cores of U_1 and U_4 are u_{min} and u_{max} which belongs to the minimum and maximum value of $u(\tau)$ taken

ISBN: 978-960-474-257-8

from the set of data), and sixteen fuzzy singleton consequents corresponding to the sixteen rules. Each parameter is a real value with decimal precision of three digits. Five conditions (Conditions two to five correspond to the monotonocity of the fuzzy model) have to be fulfilled:

- $a_1 < a_2$ and $b_1 < b_2$
- $c_{11} < c_{12} < c_{13} < c_{14}$
- $c_{21} < c_{22} < c_{23} < c_{24}$
- $c_{31} < c_{32} < c_{33} < c_{34}$
- $c_{41} < c_{42} < c_{43} < c_{44}$

Fitness evaluation:The second task in the implementation of a GA is to specify the principles for the fitness evaluation. In the approach presented here, the objective funcion to be minimized is the root mean squared error (RMSE) given by the following equation:

$$RMSE = \bar{J} = \sqrt{\frac{y_k - \hat{y_k}}{N}} \tag{11}$$

Where y_k is the real output and $\hat{y_k} = \hat{y}(\tau + 1)$ is the output of the singleton fuzzy model. As the GA operators are designed to maximize the fitness function, the above minimization has to be transformed into a maximitation one. However, at the same time the monotonocity conditions of the fuzzy model has to be satisfied. In order to do this, a function called f_m is included in equation (11) and the following transformation is considered:

$$J = \frac{f_m}{\bar{J} + 1} * C_{cm} \tag{12}$$

Where f_m is equal to 1 if the constraint established in equation (7) is true; otherwise f_m = 0. On the other hand, if $C_{cm} = 1$ then the firts condition established for the cores of the membership functions A_2, A_3, U_2 and U_3 is fulfilled; otherwise $C_{cm} = 0$.

Genetic operator: genetics operators are used to create new solutions based on existing solutions in the population. There are two basic operators: crossover and mutation. Crossover takes two individuals and produces two new individuals while mutation alters one individual to produce a single new solution. The application of these two basic operators and their derivatives depends on the chromosone representation used. Operators for floating-point representations, were developed by Michalewicz [14]. Some of these operators are: uniform mutation, non-uniform mutation, multi-non-uniforms mutation, boundary mutation, simple crossover, and heuristic crossover. In this particular work, we have used simple crossover and uniform mutation; the crossover rate was set to 0.95 and the mutation rate was set to 0.05. In order to choose the bests individuals in each generation a roulette wheel approach was adopted as the selection procedure; which belongs to the fitness-proportional selection and is able to select a new population with respect to the probability distribution based on fitness values.

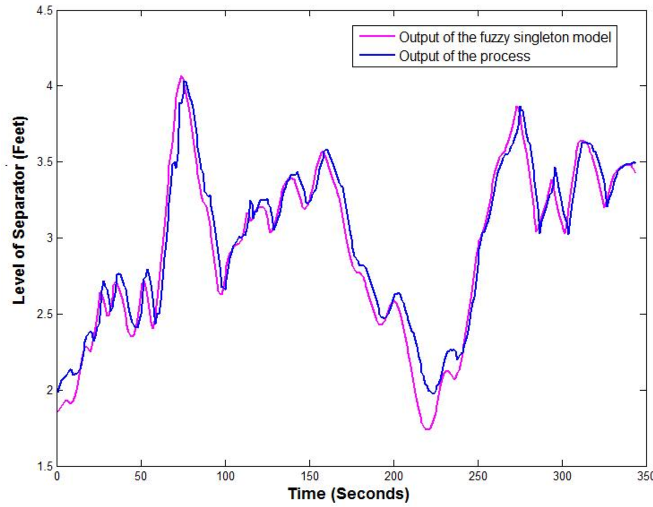


Figure 5: Output of the process vs Output of the optimal fuzzy singleton model

Initialization, termination and evaluation: the GA must be provided with an initial population.The most common method is to randomly generate solutions for the entire population.The GA moves from generation to generation selecting and reproducing potential solutions until a termination criterion is met. The most frequently used stopping criterion is a specified maximum number of generations. In this case, we have defined a maximum number of generations equal to 500 or when J is greater or equal to 0.9524 which means that $RMSE < 0.05$.

The GA was applied using Matlab, specifically the Genetic Algortihms For Optimization Toolbox. The initial population was 16 fuzzy singleton models. In generation 257 $J = 0.9578$ and the optimized fuzzy singleton model obtained had the following parameters: $a_1 = 2.446$, $a_2 = 3.278$, $b_1 = 41.660$, $b_2 = 70.82$, $c_{11} = 1.729$, $c_{12} = 1.801$, $c_{13} = 1.915$, $c_{14} = 2.100$, $c_{21} = 2.181$, $c_{22} = 2.423$, $c_{23} = 2.490$, $c_{24} = 2.619$, $c_{31} = 3.182$, $c_{32} = 3.231$, $c_{33} = 3.334$, $c_{34} = 3.431$, $c_{41} = 3.650$, $c_{42} = 3.921$, $c_{43} = 4.023$ y $c_{44} = 4.182$. The model was validated using a new

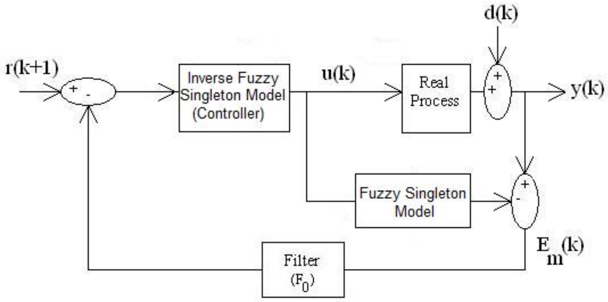


Figure 6: Internal Model Control Based on Fuzzy Singleton Model

set of 343 data. Figure 5 shows the generalization capability of the obtained optimal fuzzy singleton model obtained.

4 Inverse Fuzzy Control

In order to apply the internal model control (ICM) strategy four elements has to be determined. The process model and the controller (The fuzzy model and its inverse) may be derived considering the premises studied in section 2 and 3. A third element is the real process. In this case the valve, the pneumatic actuator and the separator constitute the whole process, for which an approximate mathematical model has to be determined. The last element is a feedback filter which is introduced in order to filter out the simulated measurement noise and to minimize the instability introduced by the feedback of the modeling error. Figure 6 shows the scheme of the ICM based on fuzzy singleton model.

Finally, figure 7 shows the response of the output of the plant with the ICM architecture implemented. As it can be observed, a second order response is generated.

5 Conclusion

This paper has proposed a method to generate satisfactory GA-Fuzzy Singleton Models. This methodology was applied to the fuzzy modeling of an oil production separator using real input-output data. A serie of constraints were carefully considered so that the analitycal model inversion is guaranteed. In order to apply the ICM strategy, the mathematical model of the valve, pneumatic actuator and the separator was taken into account as a unique process.

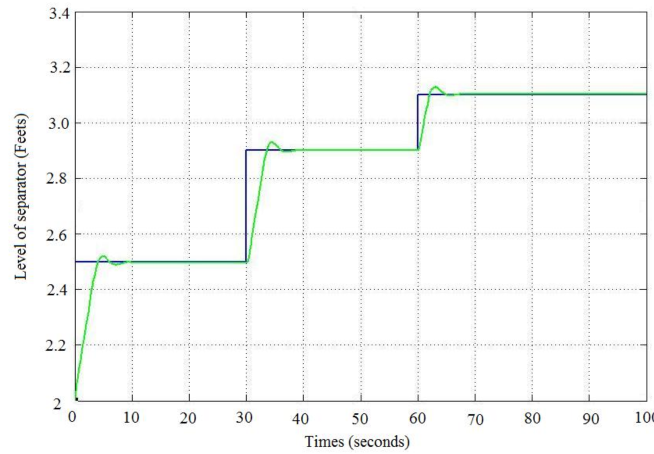


Figure 7: Response of $y(\tau)$ based on the changes in the reference $r(\tau+1)$

References:

[1] R. Boukezzoula, S. Galichet and L. Foulloy, Nonlinear Internal Model Control:Application of Inverse Model Based Fuzzy Control, *IEEE Transactions on Fuzzy System* Vol 11, No 6, 2003, pp. 814–829.

[2] J. Abonyi, L. Nagy and F. Szeifert, Indirect Model Based Control Using Fuzzy Model Inversion, *Department of Chemical Engineering, University of Vezprem, Hungary* 2005.

[3] M. Ramirez, E. Colina, Fuzzy Model Based Control: Application to an Oil Production Separator, *IEEE Proceedings on Hybrid Intelligent Systems, Barcelona Spain* 2005.

[4] J. Roubos, S. Mollov and R. Babuska, Fuzzy Model Based Predictive Control By Using Takagi Sugeno Fuzzy Models, *International Journal Of Approximate Reasoning* 22:3-30.

[5] J. Contreras, R. Misa and L. Urueta, Algoritmos Para Identificacin de Modelos Difusos Interpretables, *IEEE Latin American Transactions* Vol 5, No 5, September, 2007.

[6] L. Davis, The Handbook Of Genetic Algorithm, *Van Nostrand Reingold* New York 1991.

[7] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, *Addison-Wesley* 1989.

[8] K. Chan, V. Lee and H. Leung, Generating fuzzy rules for target tracking using a steady-state genetic algorithm, *IEEE Transactions on Evolutionary Computation* Vol 1, 1997, pp. 189–200.

[9] C. Cheng and C. Wong, Self-generating rule-mapping fuzzy controller design using genetic algorithm, *IEEE Transactions on Fuzzy System* Vol 3, pp. 129–139, May 1995.

[10] A. Homaifar and E. Mccormick, Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithm, *IEEE Proceedings on Control Theory Application* Vol 149, No 2, March 2002.

[11] W. Wang, T. Yeng and C Sung, GA-Based Fuzzy Rules Generation, *Department of Electrical Engineering, National Central University* , 2002.

[12] R. Babuska, Fuzzy Modeling For Control, *Kluwer Academic Publishers* , Boston, 1998.

[13] J. Sousa and U. Kaymak, Fuzzy Decision Making In Modeling And Control, *World Scientific Series in Robotics and Intelligent Systems* , Vol 27, 2003.

[14] Z. Michalewicz, Genetic Algorithm + Data Structure = Evolution Programs, *AI Series Springler New York* 1994.

[15] M. Ramirez and E. Colina, Fuzzy Clustering Based Models Applied to Petroleum Processes, *WSEAS Transactions on Systems and Control* , Issue 3, Vol3, March 2008.