Fuzzy Classifier System and Genetic Programming on System Identification Problems

Jose Aguilar
CEMISID, Dpto. de Computación,
Facultad de Ingeniería.
Universidad de Los Andes
Mérida, VENEZUELA.

Mariela Cerrada
CEMISID, Dpto. de Control,
Facultad de Ingeniería.
Universidad de Los Andes
Mérida, VENEZUELA.

Abstract

In this work, two techniques of Computational Intelligence, Fuzzy Classifier System and Genetic Programming, are compared on system identification problems. By using a Fuzzy Classifier System, we pretend to find an input-output identification fuzzy model (composed of fuzzy rules). The Fuzzy Classifier System uses a genetic algorithm in order to adapt an initial population of fuzzy rules. In Genetic Programming, a set of analysis trees (the nodes are a set of mathematical symbols: constants, functions, variables, operators, etc.) is the population manipulated by the evolutionary algorithm. These analysis trees describe the possible different identification models. In both cases, the initial population is generated based on intuitive knowledge about the dynamic of the system. A set of historical data about input and output signals is used to adapt that population

1 INTRODUCTION

In processes control are required models which describe the dynamic behavior of the system in order to carry out control tasks [7,11,13]. Identification techniques propose an approximated model of a real system, based on linguistic or mathematical expressions, or an algorithm. Identification models that only manipulate input and output signals is one of the possible identification schemes (Input-Output Identification Models). In control theory, there are many techniques to solve this problem [10]. In this work, two intelligent mechanisms based on Evolutionary Computation (EC) are proposed in order to solve the input-output identification problem of dynamical system, one of these based on Genetic Programming (GP) and the other one based on Fuzzy Classifier System (FCS). In the case of GP, this approach proposes the evolution of a set of possible models that characterize the system. In specific, the evolutive process manipulates a population of analysis trees, which describe the possible models. In the case of FCSs, an input-output identification fuzzy model is generated from an initial population of fuzzy rules. Genetics Algorithms (GAs) are used to propose a new population of rules through an iterative cycle of states, until minimizing the identification error.

2 SYSTEM IDENTIFICATION (SI)

In control tasks, it is necessary to known the system model that describes the behavior of the system [7, 10, 11, 13]. The identification methods develop models which are capable of describe the essential properties of a system, taking into account its static and dynamic behavior during an interval of time. Such models can be used in control tasks, fault tolerance, etc.

There are many identification methods, several of them based on the control theory [10], or on the computational intelligence [1, 2, 12]. The identification models can be defined as a non-linear function of the current input (u(t)) and previous inputs (u(t-1), u(t-2) and so forth) and outputs (y(t-1), y(t-2) and so forth) (these models are called input-output identification models) [10]. The classical scheme for system identification is shown in the figure 1. The error signal between the real output and the estimated output is used to update the model parameters.

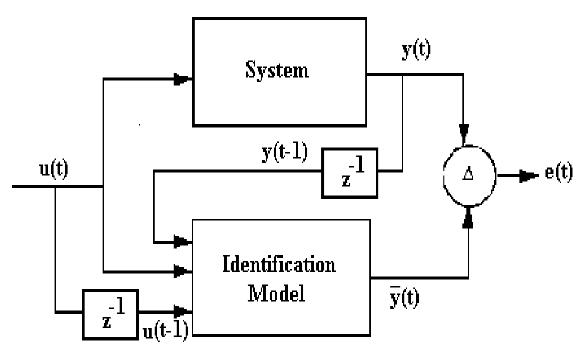

Figure 1: System identification scheme.

3 FCS AND GP ON IDENTIFICATION PROBLEMS

3.1 FCS-BASED IDENTIFICATION MECHANISM

In a previous work [3], a FCS for fault tolerance in industrial processes has been designed. Some ideas of that work have been used in order to design our FCS approach for system identification problems. The FCS generates an input-output identification fuzzy model, which is obtained from historical data about the input and output variables of the system. Our identification scheme based on FCS is shown in the figure 2.

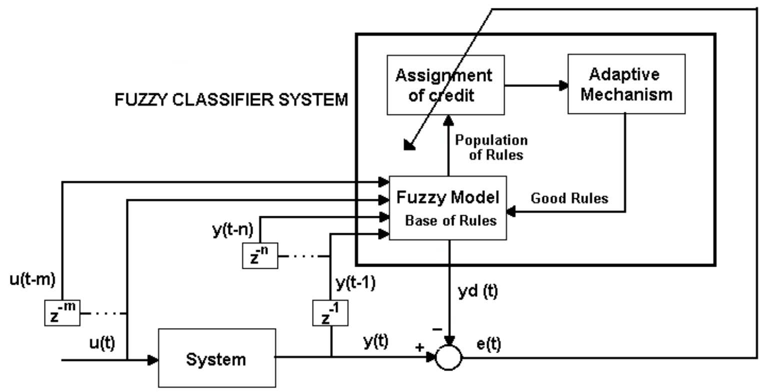

Figure 2: Our identification scheme based on FCS

In this design, we suppose that both the generic structure of the fuzzy rules and the membership functions of the fuzzy sets are known. Then, the FCS only finds the best instances of this generic structure.

3.1.1 Algorithm of the FCS.

For each training pattern, according to the historical data of the system and a population of "n" fuzzy rules, we follow the next steps:

1. Compute the activation grade of each rule.
2. Compute the credit of each activated rule.
3. Defuzzification of the output fuzzy set obtained by the fuzzy inference mechanism.
4. Compute the identification error *er*.
5. Compute the average error *ep*, when all patterns have been processed.
6. If average error is bigger than the error limit given by the user, then the FCS uses the adaptive mechanism based on GAs.
    - 6.1 Choose the parents (rules with high credit value).
    - 6.2 Apply the genetic operators (mutation and crossover).
    - 6.3 Replace the olds individuals for the new individuals, according to some replacement mechanism.

This procedure is repeated until that the identification average error reaches a minimum value given by the user or a maximum number of iterations have been accomplished.

3.1.2 The identification error calculation

The equation (1) is used to calculate the identification error associated to each pattern. The average error for all training patterns is given by the equation (2).

$$er = |(y_s - y_d)/y_s| \tag{1}$$

$$ep = \Sigma_{i=1}^{m} er/m \tag{2}$$

where y_s is the output of real system, y_d is the output of the fuzzy model and m is the number of patterns.

3.1.3 The fitness function definition.

The credit value of each fuzzy rule is computed based on the fitness function given by the equation (3):

$$S_i(t+1)=S_i(t)+Act_i(t)*\mu y_i/ea \tag{3}$$

where $S_i(t)$ is the credit value of the fuzzy rule i at time t, $Act_i(t)$ is the activation grade of the fuzzy rule i at time t, ea is the absolute error ($ea=y_s-y_d$) and μy_i is the membership grade of the crisp value of the fuzzy model output. This fitness function permits the evaluation of the weight of the output fuzzy set of a rule into the crisp value given by the fuzzy model. So, a good credit value is obtained for those rules which give a minor identification error.

3.1.4 The adaptive mechanism

Each rule is codified as a vector of finite length, as it is shown in the figure 3.

Ve_1	CD_1	....	Ve_3	CD_3	Vs	CDs

Figure 3: Codification of a rule as an individual

where Ve_i is the input variable i, CD_i is the fuzzy set of the input variable Ve_i , Vs is the output variable and CDs is the fuzzy set of the output variable Vs.

In this work, we propose a set of changes into the fuzzy sets of the input and output variables in order to create new rules. The genetic operators of crossover and mutation are used in order to accomplish this task [6]. At the end, the new population is composed of $n+k$ rules (individuals), where n is the number of rules of the previous population and k is the number of new rules. In order to have n rules, we must eliminate k rules. We eliminate a rule according to its probability of elimination, given by the equation (4):

$$Pr(R_t)= Fr(R_t)/ \sum_{i=1}^{m} Fr(R_i) \qquad (4)$$

where P_r is the replacement probability of the rule R_i, F_r is the replacement factor of the rule R_i and m is the number of rules of the population (m=n+k). The replacement factor is given by the equation (5):

$$Fr(R_i)=1-FA_i/\sum_{j=1}^{m} FA_i \qquad (5)$$

where FA_i is the credit value of the rule R_i.

3.2 GP-BASED IDENTIFICATION MECHANISM

In this section it is proposed a method based on PG to develop identification models. In our approach, each individual is defined by a Multiple Interaction Programs (MIP) model. In the MIP model, each node is one equation, which is represented by an analysis tree. The identification mechanism proposes a simultaneous evolution of each analysis tree [1].

In our model, the terminal set of each node has input variables, constants or outputs from some precedent equations. In the figure 4.b is shown an analysis tree for T3, where *In1* y *In2* are input values of the problem. *T1* y *T2* are the outputs of these equations, which precede T3 (see figure 4.a). This model is easy to implement in GP, through the utilization of the ADF (Automatic Definition Function) technique. This extension of GP permits to define functions to evolve in parallel with the main procedure. These functions can be called by other functions, or by the main procedure, during the evolution. In our case, the MIP model defines the relationship among the functions. The population evolution follows the next algorithm:

1. Define a given MIP model for the individuals.
2. Generate, randomly, a population of individuals. Each one of the individuals is defined by a set of analysis trees according to the MIP model.
3. Evaluate each individual in order to determine its performance. The evaluation function is the average error between the historical output of the system and the output of the identification model (individual).
4. Select the parents (individuals with the smallest average error).
5. Apply the genetic operators to these parents in order to reproduce new individuals.
6. Replace the old worst individuals for the new individuals.

4 EXPERIMENTS

In this section, we present an example in order to compare both proposed identification methods. The example is a distillation system that uses a distillation column in continuous operation of multiple stages.

4.1 SYSTEM DESCRIPTION

The objective of a distillation system is to separate a mixture in two or more fractions with different boiling points. The function of the continuous distillation system can be seen with details in [7]. In the figure 5 is shown the structure of this distillation column. The feeding input (composed by benzene and toluene) is introduced in the second plate, and the distilled product is obtained in the first plate on the top of the column.

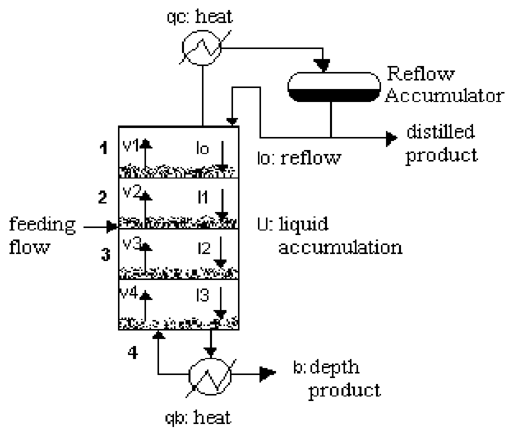

Figure 5: Distillation column

The constant input signal (feeding rate) is modeled with a step function with amplitude equal to ten (U(t) = 10). The theoretical model of this system is given by the equation (6) [7]:

X(t) =1.1148*X(t-1) + 0.2525*X(t-2) – 0.3823*X(t-3) + 0.3294e-4*U(t-1) (6)

where X(t) represents the output of the system. The output is the concentration of benzene on the top. The output signal from this model is shown in the figure 6.

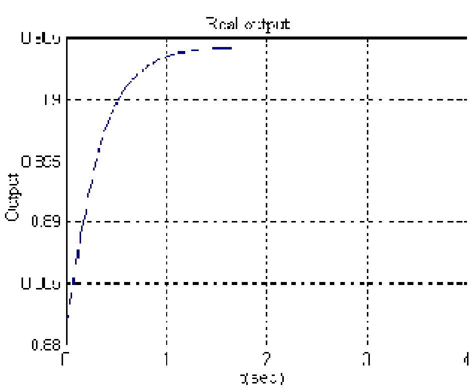

Figure 6: Output signal from theoretical model

4.2 IDENTIFICATION MODEL BASED ON GP

In order to develop the computational program, we have used the "The Genetic Programming Kernel" library designed by A. Fraser en 1994 [5]. This library permits the utilization of ADFs.

In this experiment, the MIP model is composed by two equations (M1 y M2), where M2 represents the ADF and M1 represents the main program (main tree), which can depend of M2 or not. The function set used by M1 and M2 is {+,-,*, %, *sin, cos}*. The terminal set of the main tree is composed by St(M1)={*u, xa1, xa2, xa3, xa4, xa5, s_M2*}, where *u* is the input signal at the time *t, xa1* is the output signal at the time t-1 *(X(t-1)), xa2* is the output at the time t-2 *(X(t-2)),* and so on, and s_M2 is the output of the ADF. The terminal set of the ADF only has two elements St(M2)={*xa1, xa2*}. The trigonometric functions are supposed with input values given in radians.

The historical values of the input and output signals have been obtained using the theoretical model defined by the equation (6). The aptitude of each individual was determined based on the average error between the output historical values and the outputs of the model proposed by the individual for the same set of input signals. A population of 300 individuals has been evolved through 50 generations. Finally, the individual with the smallest average error is selected. In the table 1 is shown the models obtained (the best individuals) using our identification method, for different terminal sets.

Table 1: Identification Models

CASES	**IDENTIFICATION MODEL**	**THEORETICAL MODEL**	**ERROR**
AP: C_T={u, a1, xa2, xa3, s_M2}	M1 = 2*xa1 – xa2*s_M2	Equation (6)	1.59254e-4
ADF: C_T={x1, x2}	M2 = $(xa1)^2$ / xa2		
AP: C_T={u, xa, xa2, s_M2}	M1 = 2*xa1 – xa2* s_M2	Equation (6)	2.3965e-4

ADF: C_T={x1, x2}	M2 = Equation (7)		
AP: C_T={u, xa1, a2, xa3, xa4, s_M2}	M1 = (xa1 / xa2)*xa1	Equation (6)	2.86043e-4
ADF: C_T={x1, x2}	M2 = xa1+xa2-xa3		
AP: C_T={u, xa1, xa2, xa3, xa4, xa5, s_M2}	M1 = 2*xa1 – xa2 *s_M2	Equation (6)	1.59254e-4
ADF: C_T={x1, x2}	M2 = $(xa1)^2$ / xa2		

where:

M2=xa1-sin(sin(sin(sin(sin(sin(sin(sin(sin(sin(sin(sin(sin(sin((xa2-xa1))))))))))))))) (7)

The identification models obtained in the cases 1 and 4 are similar, and they are the best models. In the second case, the ADF model is different to the previous ones, but the value of the error is acceptable. In the case 3, the identification model do not depends of the ADF. In general, in all cases the best individual depends of the output signal at the times (*t-1*) and (*t-2*), and it does not depend of the input signal. In the second case, the identification model is more complex.

The identification error signal obtained by the model proposed in the case 2 is shown in the figure 7. The input signal is a constant function U(t)=10, and the initial conditions for the variables xa1 y xa2 was randomly selected near to the real initial conditions. At t=2 sec., the identification error converges to zero.

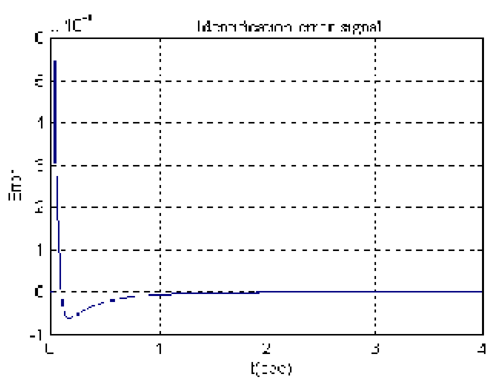

Figure 7: Identification error using the second model

4.3 IDENTIFICATION MODEL BASED ON FCS

In our approach, we suppose the following generic structure for the fuzzy rules:

If U(t) and Y(t-1) then Y(t) (8)

where U(t) denotes the input variable at time t, Y(t-1) denotes the output variable at time t-1 and Y(t) denotes the output variable at time t. For such variables, we previously define their fuzzy sets according to their historical data values. The membership functions of these fuzzy sets are shown in the figure 8.

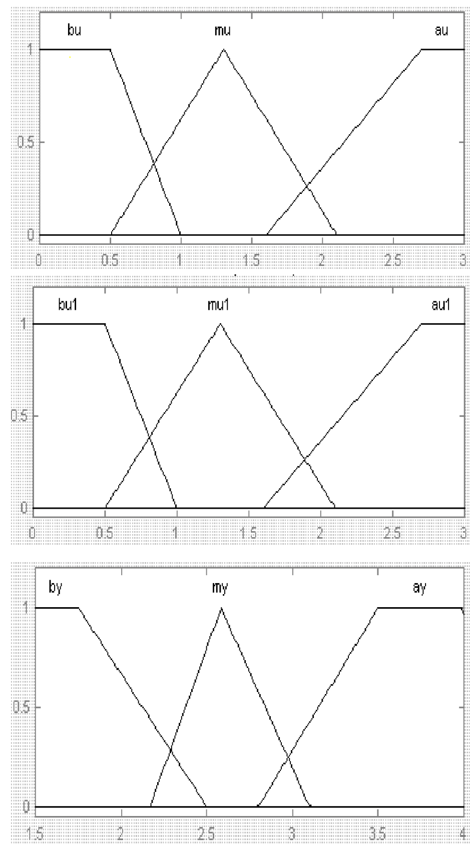

Figure 8: Membership functions of the fuzzy sets for U(t), Y(t-1) y Y(t).

Different experiments have been made from an initial population of fuzzy rules and 800 training patterns. The best fuzzy model according to the identification average error is the following:

If U(t) is mu and Y(t-1) is bu1 then Y(t) is ay

If U(t) is au and Y(t-1) is mu1 then Y(t) is by

If U(t) is mu and Y(t-1) is au1 then Y(t) is my

If i U(t) is au and Y(t-1) is au1 then Y(t) is ay

If i U(t) is mu and Y(t-1) is mu1 then Y(t) is ay

If i U(t) is au and Y(t-1) is au1 then Y(t) is my

If i U(t) is mu and Y(t-1) is bu1 then Y(t) is my

If i U(t) is bu and Y(t-1) is mu1 then Y(t) is my

If i U(t) is mu and Y(t-1) is mu1 then Y(t) is by

This fuzzy model has been found in the iteration number 87, with an average training error of 0.13. The output of this fuzzy identification model, for the input signal U(t)=10, is shown in the figure 9.

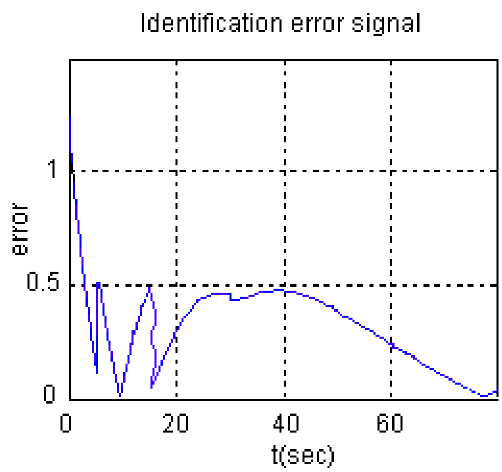

Figure 9: Identification error signal based on FCS.

In the figure 9, we can observe the identification error signal of the fuzzy model. The accuracy is not very good. We remark that the membership functions have not been adjusted. We can reach a good accuracy if these membership function are suitably adjusted. If we compare the identification error for both approaches (see figures 7 and 9), we can see that the identification error based on PG is better than the other one based on FCS.

5 CONCLUSIONS

This work shows the capabilities of the GP and the FCS in system identification problems. This application is very useful when the knowledge about the system is poor and when we not have the expert knowledge about the relationships between the system variables. The identification models that we have obtained by using these approaches are suitable.

FCS has been "training" out of line, therefore, the fuzzy model is an universal generic model. It is necessary to test our identification mechanism with different structures of the fuzzy rules in order to give more information at the FCS (more delayed input and outputs signal as inputs variables) . We can observe that there are many repeated fuzzy rules into the model, then the elimination algorithm must be improved. In the future, we will incorporate a membership function adaptive mechanism.

In the case of the GP, it depends of the function and terminal sets that are used, and the relationship established in the MIP model. In the future, we are going to test one extension of our approach where the MIP model evolves such that the evolution determines the optimal relation between the equations/variables.

Based on the experimental results, the GP-based identification mechanism is more efficient than the FCS-based mechanism, but we must remark that the FCS have not the membership function adaptive mechanism. This is a serious limitation that we must improve. Finally, other experiments will be tested in order to determine the efficiency of each proposed technique in different types of problems.

Acknowledgment

This work was partially supported by CONICIT-CONIPET grant *97003817* and CDCHT-ULA grant *I-621-98-02-A*.

References

[1] J. Aguilar, M. Cerrada, " GP-Based Approach for System Identification", accepted for publication, WSES International Conference on Evolutionary Computation, Tenerife, Spain, February 2001.

[2] P. Angeline, D. Fogel, "An Evolutionary Program for the Identification of Dynamical System", Technical Report, Natural Selection Company, 1998.

[3] M. Cerrada, J. Aguilar. "Fuzzy Classifier System: An Applications for Fault Tolerance in Industrial Processes", Proceedings of IASTED International Conference of Artificial Intelligence and Soft Computing, pp. 453-456, 1998.

[4] D. Fogel, "Evolutionary Computation: Toward a New Philosophy Through Simulated Evolution", IEEE Press, 1995

[5] A. Fraser, "Genetic programming in C++", Technical Report, University of Salfrod, Cybernetics Research Institute, 1994.

[6] D. Goldberg, "Genetic Algorithms in Optimization and Machine Learning", Addison-Wesley Publishing Company, 1989.

[7] C. Holland, "Fundamentos y Modelos de Procesos de Separación", Prentice Hall, 1997.

[8] J. R. Koza, "Genetic Programming: On the Programming of Computers by Means of Natural Selection", MIT Press, 1992.

[9] J. R. Koza, "Genetic Programming II: Automatic Discovery of Reusable Programs". MIT Press, 1994.

[10] L. Lennart, "System Identification, Theory for the User", Prentice Hall, 1997.

[11] K. Ogata, "Ingeniería de Control Moderna", McGraw Hill, 1992.

[12] P. Sastry, G. Santharam, K. Unnikrishnan, "Memory neuron networks for identification and control of dynamical system", IEEE Transaction on Neural Networks, Vol. 5, No. 2, p.p. 306-319, 1994.

[13] A. Wittenmark. "Computer Control System", Addison Wesley, 1989.