Identification of Recurrent Fuzzy Systems with Genetic Algorithms

Alexandre G. Evsukoff  
COPPE/Federal University of Rio de Janeiro,  
Rio de Janeiro RJ, Brazil  
E-mail: evsukoff@coc.ufrj.br

Nelson F. F. Ebecken  
COPPE/Federal University of Rio de Janeiro,  
Rio de Janeiro RJ, Brazil  
E-mail: nelson@ntt.ufrj.br

Abstract - This work presents an algorithm for identification of fuzzy recurrent models of non-linear dynamic systems. The identification algorithm is based on a general purpose genetic algorithm. The resulting recurrent fuzzy system is encoding into a fuzzy finite state automaton in which the linguistic terms of the fuzzy model are the states and rule base weights are transition possibilities. The identification algorithm is tested against benchmark identification problems found in literature.

I. INTRODUCTION

Recurrent fuzzy (and neuro-fuzzy) systems can represent dynamic non-linear systems and have been widely used in many different applications [5], [8], [9], [10], [11], [15], [16]. Generally, their identification algorithms rely on gradient-based methods for neural networks version of fuzzy rule base representation. Recently, Genetic Algorithms (GAs) have also been used for parameters’ estimation [9], [15].

In recurrent fuzzy systems, the time-dependent relationship among input and output variables can be difficult to interpret in a linguistic fuzzy rule representation. Recently, Adamy and Kempf [1] have shown that recurrent fuzzy systems can have automata-like (or chaotic) behavior depending on their rule base configuration. Fuzzy finite state automaton (FFA) can be used to represent a dynamic system and its relationships with RNN have been studied [7], [2]. Such relationship is useful, since the representation capabilities and theoretical properties derived from the FFA can be used to analyze and understand the dynamics of the identified model.

In this work, an identification algorithm based on GA is proposed to estimate rule base output parameters of recurrent fuzzy systems and their corresponding FFA representation, in which the linguistic terms are the states and rule base weights are transition possibilities.

The paper is organized as follows; next section presents recurrent fuzzy systems and their representations. Section three present the GA identification algorithm. Results of the proposed methodology are discussed in section four. Finally conclusions and future work are highlighted.

II. RECURRENT FUZZY SYSTEMS

Consider the state space representations of discrete time, non-linear dynamic systems:

\[ x(t+1) = f(x(t), u(t)) \]  
(1)

\[ y(t) = g(x(t), u(t)) \]  
(2)

where \( x(t) \) are the state variables, \( u(t) \) the input variables and \( f \) and \( g \) are respectively the transition and output functions. In this work, recurrent fuzzy systems are used to model the transition function, while the output function is considered to be a linear function of states, such that \( y(t) = Cx(t) \).

Two types of recurrent fuzzy systems are discussed in this section. The first one is the TSK-type recurrent fuzzy system, which has been introduced by Gorrini and Bersini [8]. The second one is called linguistic recurrent fuzzy system, is an extension of the model proposed by Adamy and Kempf [1], by introducing rule base weights. The later model can be represented by a FFA, in which linguistic terms are the states and rule base weights are transition possibilities. Both models are equivalent and a conversion procedure from the TSK-type to the linguistic type is proposed in section III.

A. TSK-type Recurrent Fuzzy Model

The Sugeno (or TSK) fuzzy model is probably the most popular fuzzy system in literature. The TSK-type recurrent fuzzy system is an extension of the original one in which the rules are written as:

\[ \text{If } x(t) \text{ is } A_{i} \text{ and } u(t) \text{ is } B_{j} \text{ then } x_{0}(t+1) = \theta_{r} \]  
(3)

where \( x(t) = (x_{0}(t), \ldots, x_{n-1}(t)) \) is the vector of \( n \) state variables. The fuzzy sets \( A_{i} \) and \( B_{j} \) are defined respectively on the state and input variables domains and \( \theta_{r} \) is the output parameter for each one of the \( r = 1 \ldots M \) rules.

All the state variables are considered in the rule premises, but only the value of the state variable \( x_{0}(t) \) is computed in the output. If more that one state is necessary to adjust the dynamics of the actual system, additional state variables are introduced as copies of the delayed values of state variable \( x_{0}(t) \) as:

\[ x_{i}(t) = x_{0}(t-i), i = 1 \ldots n-1. \]  
(4)

The set of fuzzy sets \( B = \{ B_{j}, j=1\ldots m \} \) defines a fuzzy partition in the input variable domain. Since all state

---

1 The presentation focus the single variable case, the extension to multivariable is straightforward.
variables are delayed copies of the state variable \( x_0(t) \), the same fuzzy partition \( \mathbf{A} = \{ A_i, i = 1 \ldots p \} \) can be used to compute the fuzzification of all state variables. For a given rule, the membership \( \mu_{A_i}(x(t)) \) is computed as the Cartesian product of all components as:

\[
\mu_{A_i}(x(t)) = \prod_{j=0,n-1} \mu_{A_{ij}}(x_j(t))
\]  

(5)

where \( A_{ij} \) is the projection of the fuzzy set \( A_i \) into the domain of the state variable \( x_j(t) \).

The fuzzy membership functions are considered to be triangular-shaped and normalized, such that the following conditions are satisfied:

\[
\sum_j \mu_{A_i}(x_j) = 1, \forall x \\
\forall i, \exists x_j / \mu_{A_i}(x) = 1.
\]  

(6)

(7)

Normalized and triangular-shaped membership functions are completely defined by the vector \( \mathbf{a} = (a_1, \ldots, a_p) \) that defines number and location (prototype) of membership functions vertices.

The value of the state variable \( x_0(t) \) in the next time sample is computed as:

\[
x_0(t+1) = \mathbf{w}(t) \mathbf{a}^T
\]  

(8)

where \( \mathbf{w}(t) = (\theta_1, \ldots, \theta_M) \) is the output parameters vector and \( \mathbf{a} \) is the fire strength vector.

Each rule \( r = 1 \ldots M \) produces a component of the vector \( \mathbf{w}(t) \) by the conjunction of the fuzzy sets in the premise, computed by the product t-norm:

\[
w_r(x) = \mu_{A_i}(x(t)) \mu_{B_j}(u(t)).
\]  

(9)

The number of rules \( M \) are defined by the number of linguistic terms (fuzzy sets) defined in the state and input variables’ fuzzy partitions. Considering \( n \) state variables, each one described by \( p \) fuzzy sets, and that the input variable is described by \( m \) fuzzy sets, the number of rules \( M \) is equal to \( p^m \).

The TSK-type recurrent fuzzy systems can be extended to more complex type by using polynomial functions in rules outputs [9]. The simple model with constants values in rules’ output is equivalent to the linguistic recurrent fuzzy model presented next.

### B. Linguistic Recurrent Fuzzy Model

In linguistic recurrent fuzzy systems, rules outputs are fuzzy expressions defined on the state variable. In the model presented by Adamy and Kempf [1], fuzzy rules are written as:

If \( x(t) \) is \( A_i \) and \( u(t) \) is \( B_j \) then \( x_0(t+1) \) is \( A_k \)

(10)

where \( A_i \) and \( B_j \) are fuzzy sets like above, and \( A_k \in \mathbf{A} \) is an element of the state variables’ fuzzy partition, in such a way that recurrence is achieved.

The rule base can be represented by the fuzzy relation matrix \( \Phi \), whose components \( \phi_{rk} \) represent the relationship between the premise and the output expressions. There is one rule \( r = 1 \ldots M \) for each combination \( (A_i, B_j) \) of the terms in the premise, in such a way that \( \phi_{rk} = 1 \) if the output of the rule \( r \) is the fuzzy set \( A_k \) and \( \phi_{rk} = 0 \) otherwise. The matrix \( \Phi \) has thus \( M \) lines and \( p \) columns.

The output of a recurrent fuzzy system is computed as an ordinary fuzzy system. Choosing the appropriate inference and defuzzification operators, the state variable value in the next time sample can be computed as:

\[
x_0(t+1) = \mathbf{w}(t) \Phi \mathbf{a}^T
\]  

(11)

where \( \mathbf{w}(t) \) is computed like above, \( \mathbf{a} = (a_1, \ldots, a_p) \) is the state variable prototype vector and the superscript \( ^T \) indicates the transpose.

Comparing the equations (8) and (11), the models will be equivalent if \( \theta_k = a_k \), where \( a_k \) is the numeric value associated with the term \( A_k \), as it was originally proposed by Gorrini and Bersini [8].

A more flexible model can be obtained if confidence weights are associated to each rule. In this case, each component of the fuzzy relation matrix \( \Phi_{rk} \in [0,1] \) and the rules are written as:

If \( x(t) \) is \( A_i \) and \( u(t) \) is \( B_j \) then

(12)

\[
x_0(t+1) = (A_1 / \phi_{r1}, \ldots, A_p / \phi_{rp})
\]

Comparing the equations (8) and (11), the linguistic fuzzy model weights described by rules like (12) are related to the TSK output parameters as:

\[
\theta = \Phi \mathbf{a}^T
\]  

(13)

Linguistic recurrent fuzzy systems can be directly mapped into FFA, as discussed in next section.

### C. FFA Representation

A FFA is an extension of the finite automaton, where more than one state can be simultaneously activated with different degree of certainty. Formally, a FFA is a triple [13]

\[
\mathcal{F} = (Q, S, R),
\]

where \( Q \) is a nonempty finite input alphabet; \( S \) is a finite set of states and \( R \) is the transition mapping \( Q \times S \times Q \to [0,1] \).

The model defined by a set of weighted recurrent fuzzy rules (12) can be encoded into a FFA \( \mathcal{F} = (Q, S, R) \), referred as a “linguistic automaton” by Adamy and Kempf [1], where the input set \( Q = \{B_j, j = 1 \ldots m\} \) is the set of linguistic terms associated to the fuzzy sets in the input variable’s fuzzy partition as so as the set of states \( S = \{A_i, i = 1 \ldots n\} \) is
The objective of parameters estimation and the input remains defined by Table I. The resulting FFA is shown in Fig. 1.

This system can be represented by weighted recurrent linguistic fuzzy system when \( \alpha \in [0,1] \) and the input remains within a bounded interval. The minimum and the maximum values allowed for input variable define the fuzzy partition \( B = \{B_1, B_2\} \) on the input variable domain and also the fuzzy partition \( A = \{A_1, A_2\} \). The fuzzy relation matrix in Table I define a set of rules like (12).

<table>
<thead>
<tr>
<th>( x(t) )</th>
<th>( u(t) )</th>
<th>( A_1 )</th>
<th>( A_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_1 )</td>
<td>( B_1 )</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>( A_1 )</td>
<td>( B_2 )</td>
<td>( \alpha )</td>
<td>( 1-\alpha )</td>
</tr>
<tr>
<td>( A_2 )</td>
<td>( B_1 )</td>
<td>( 1-\alpha )</td>
<td>( \alpha )</td>
</tr>
<tr>
<td>( A_2 )</td>
<td>( B_2 )</td>
<td>0.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Whenever the input variable remains within its limits, the model output computed as the output of the linguistic recurrent fuzzy system (11) will be equal to (14).

This fuzzy model can be represented as a FFA, where \( S = \{A_1, A_2\}; Q = \{B_1, B_2\} \) and the transition mapping \( R \) is defined by Table I. The resulting FFA is shown in Fig. 1.

The example described above was chosen to illustrate a particular relationship between a linear system, its recurrent fuzzy model and the corresponding FFA. In this case, the coefficients of the state space equation are equal to the fuzzy rules weights and thus equal to the transition mapping of the FFA. This equivalence is no longer valid for a more general model. Nevertheless, appropriate weights can be identified from an input/output data set.

More complex non-linear dynamics can be obtained by a recurrent fuzzy system with more complex structures. The corresponding relation matrix can be identified by the algorithm presented in next section.

III. IDENTIFICATION ALGORITHM

In general, the identification of complex processes follows a methodology composed of three steps:
1. Structure identification,
2. Parameter estimation from a data set representative of system behavior called training set.
3. Model validation, considering modeling objectives: prediction, simulation, diagnosis, etc.

The structure identification includes the model selection and transformation. For a fuzzy model, the structure identification consists of the determination of the model type and the number and location of fuzzy sets in the domain of each variable. The model parameters are generally associated with the rule conclusions. The model validation must check the model precision, but also certifies that the model is readable by domain experts.

In this work, the structure of the recurrent fuzzy system will not be considered. The number of fuzzy sets associated to each variable is fixed and the corresponding prototypes are located equally spaced within the variable domain.

The output parameters’ estimation is computed considering a training set \( T \) composed by \( N \) samples \( \{u(t), y(t)\} \), where \( y(t) = f(u(t)) \). The objective of parameters’ estimation is to build a fuzzy model to compute a model \( \hat{y}(t) = \hat{f}(u(t)) \) that minimizes the Mean Squared Error (MSE) criterion:

\[
J = \frac{1}{N} \sum_{t=1}^{N} (y(t) - \hat{y}(t))^2.
\]  

For a non-recurrent fuzzy system, the minimization of the criterion (16) is done by solving a linear system of equations [4], since output parameters occur linearly in the output estimate \( \hat{y}(t) \). In the case of a recurrent fuzzy system, the values of the state variables depend on their values in the past time sample, as shown by equations (8) and (9), and the output estimate becomes non-linear with respect to the output parameters.

The most common approach for the parameter estimation problem in recurrent fuzzy (or neuro-fuzzy) systems is to use a gradient based algorithm [5], [8], [11], [10]. Recently the use of genetic algorithms for the identification of recurrent fuzzy systems has been proposed [9], [15] with good results.

The idea of the methodology proposed in this work is to identify first a TSK-type recurrent fuzzy system and then translate it to a linguistic recurrent fuzzy system using equation (13). The identification algorithm is based on a general purpose genetic algorithm.

A. Genetic Algorithms

The main goal of the identification algorithm is to produce a FFA encoding from a linguistic recurrent fuzzy system. The identification algorithm was developed to attain this goal.
Genetic Algorithms (GAs) have been widely used in a number of applications of fuzzy systems [3]. GAs are simple, robust and many implementations are available in the internet [14]. Moreover, GAs can overcome local minima problems found in most of gradient-based optimization algorithms.

The identification algorithm proposed in this work employs GAs in two steps. First, a GA is used to estimate the rules’ output parameters of a TSK-type recurrent fuzzy system from a training data set. In the second step, another GA is used to convert rules’ output into rule base weights, which are the transition possibilities of the corresponding FFA.

The problem solving procedure of GAs is inspired in natural evolution. The candidate solutions for a given problem are encoded into an individual. An initial population of randomly generated individuals is evolved towards the solution. At each iteration (called generation) the individuals are rated by their fitness to the solution and a selection mechanism assures that only the best individuals are retained for the next generation by applying recombination operators such as mutation and crossover. Randomly driven recombination operators allow the individuals to explore the all solution space, avoiding local minima.

Although there are many variants, a typical GA structure is sketched in Fig. 2.

Procedure GA

\[
\begin{align*}
& t = 0; \\
& \text{initialize } P(t); \\
& \text{evaluate structures in } P(t); \\
& \text{while (not termination)} \quad \begin{align*}
& t = t + 1; \\
& \text{select } P(t) \text{ from } P(t-1); \\
& \text{recombine structures in } P(t); \\
& \text{evaluate structures in } P(t); \\
& \end{align*} \\
& \text{end.}
\end{align*}
\]

At each iteration (called generation) the individuals are rated by their fitness to the solution and a selection mechanism assures that only the best individuals are retained for the next generation by applying recombination operators such as mutation and crossover. Randomly driven recombination operators allow the individuals to explore the all solution space, avoiding local minima.

When applying a GA to a new problem, two main issues must be defined by the user: the codification of the variables into individuals and the fitness function. These issues are discussed in the following for the identification of recurrent fuzzy models.

B. Recurrent Fuzzy Model Identification

The GENESYS 5.0 [6] source code was used in this work to implement the GA. Using GENESYS, only the evaluation function must be coded by the user. Additional information is provided by the user in separate files.

The identification algorithm of the recurrent fuzzy model is the GA implemented by GENESYS, in which the fitness function is defined by the MSE criterion (16).

The output estimate \( \hat{y}(t) \) is computed by a linear function of the state variables:

\[
\hat{y}(t) = Cx(t)
\]  

where \( x(t) = (x_0(t), \ldots, x_{n-1}(t)) \) is the vector of \( n \) state variables and \( C = [c_1 \ldots c_n] \) the output matrix, which must also be computed by the identification algorithm.

The codification of variables into individuals imposes bounds onto the state and input variables values. Codification schemes vary according to the GA implementation. The GENESYS source code allow floating point representation of variables in such a way that only the minimum and maximum bounds of each variable must be informed by the user in a separate file.

The identification algorithm starts with the normalization of the input/output samples in the training set into the \([-1,1]\) interval. The minimum and the maximum values of the actual system output are saved to convert the model output into the actual system scale.

In the first step of the identification algorithm, an approximation of the transition function of the actual system is computed by the TSK-type recurrent fuzzy system as defined in equation (8). Since all state variables are delayed versions of state variable \( x_0(t) \) only one set of rules’ output parameters are required for each output.

The individual is coded as a vector, in which each component is one of the parameters to be identified. In the single output case, the individual vector is composed by the rules’ output parameters \( \theta = (\theta_1, \ldots, \theta_M) \) plus the output parameters \( C = [c_1 \ldots c_n] \).

In the second step, the solution \( \theta \) computed in the first step is used to compute the equivalent linguistic recurrent fuzzy model from equation, defined by rules like (12).

The linguistic fuzzy model will be equivalent to the TSK fuzzy model when equation (13) holds. Nevertheless, equation (13) defines an underdetermined linear system of equations, which must be solved in the Least Squares sense by minimizing the norm of the residual as:

\[
\min_{\Phi} \| \theta - \Phi a^T \|. 
\]

The solution of the problem (18) must be subjected to the constrain that the components of the rule base weights matrix components \( \phi_{rk} \in [0,1] \). This defines a constrained and underdetermined Least Squares problem which has a global minimum solution, which may not be unique. This problem is solved using a second GA, where individuals represent the components \( \phi_{rk} \) of the rule base weights matrix \( \Phi \). The fitness function is defined from (18).

IV. RESULTS AND DISCUSSION

The algorithm presented above was tested with benchmark identification problems found in the literature. The first example is the classic Box and Jenkins data set. The second example is the non-linear dynamic system first presented by Narendra and Parthasarathy [12], but also tested in many other works [8], [9], [10], [11].
The examples’ results are discussed in different points of view. In the first example, the interest is on the resulting FFA while in the second example the numeric performance is compared with results presented in the literature.

A. Box & Jenkins Data Set

This example is the well-known Box & Jenkins data set, where the system to be modeled is a gas furnace. The original data set is composed of 296 pairs.

This data set has been used to evaluate several system identification methods, but the data have not always been used in the same way. In this work, the model was built using delayed values of the system input \( u(t-3) \) to compute an estimation \( \hat{y}(t) \). Each training sample is thus expressed in the form \( (u(t-3), y(t)) \).

The fuzzy system structure to be identified used \( n = 2 \) state variables with \( p = 2 \) fuzzy sets for their fuzzification and \( m = 2 \) fuzzy sets for the fuzzification of the input variable. This structure results in \( M = 8 \) rules that are defined by 16 parameters. The total number of parameter to be identified is then 18, considering the two parameters of the output matrix. The plot of the model output and the actual system output is shown in Fig. 3.

The rule base weights are shown in Table II and the corresponding FFA is shown in Fig. 4.

| TABLE II |
| Fuzzy Relation Matrix |
|-----------------|-----------------|-----------------|
| \( x(t-1) \) | \( x(t) \) | \( u(t) \) | \( A_1 \) | \( A_2 \) |
| \( A_1 \) | \( A_1 \) | \( B_1 \) | \( \varphi_{11} \) | \( \varphi_{12} \) |
| \( A_1 \) | \( A_1 \) | \( B_1 \) | \( \varphi_{21} \) | \( \varphi_{22} \) |
| \( A_1 \) | \( A_2 \) | \( B_1 \) | \( \varphi_{31} \) | \( \varphi_{32} \) |
| \( A_1 \) | \( A_2 \) | \( B_1 \) | \( \varphi_{41} \) | \( \varphi_{42} \) |
| \( A_2 \) | \( A_1 \) | \( B_1 \) | \( \varphi_{51} \) | \( \varphi_{52} \) |
| \( A_2 \) | \( A_1 \) | \( B_2 \) | \( \varphi_{61} \) | \( \varphi_{62} \) |
| \( A_2 \) | \( A_2 \) | \( B_1 \) | \( \varphi_{71} \) | \( \varphi_{72} \) |
| \( A_2 \) | \( A_2 \) | \( B_2 \) | \( \varphi_{81} \) | \( \varphi_{82} \) |

The first rule, for instance, of the rule base may be read as: “if the state \( x(t-1) \) is \( A_1 \) and the state \( x(t) \) is \( A_1 \) and the input is \( B_1 \), then \( x(t+1) \) is \( A_1 \) with weight \( \varphi_{11} \) and \( x(t+1) \) is \( A_2 \) with weight \( \varphi_{12} \)”. The remaining rules can be read in the same way.

B. Nonlinear Dynamic System

In this example, the actual system output is governed by the following difference equation:

\[
y(t+1) = \frac{y(t)y(t-1)y(t-2)u(t-1)(y(t-2)-1)+u(t)}{1+y^2(t-1)+y^2(t-2)} \tag{19}\]

The training data set for the identification of a TSK-type recurrent fuzzy system was computed from an input \( u(t) \) generated as follows: 400 samples of an independent and identically distributed uniform sequence over the \([-1,1]\) interval and 400 samples of a sinusoidal signal \( u(t) = 1.05 \sin(\pi t / 45) \). The testing data set was generated considering 1000 samples of the following signal:

\[
u(t) = \begin{cases} 
\sin(\pi t / 25) & t < 250 \\
1 & 250 \leq t < 500 \\
-1 & 500 \leq t < 750 \\
0.3 \sin(\pi t / 25) + 0.1 \sin(\pi t / 32) + 0.6 \sin(\pi t / 10) & \frac{750}{\pi} \leq t < \frac{1000}{\pi} 
\end{cases} \tag{20}
\]

The fuzzy system structure to be identified used \( n = 2 \) state variables with \( p = 3 \) fuzzy sets for their fuzzification and \( m = 3 \) fuzzy sets for the fuzzification of the input variable. This structure results in \( M = 27 \) rules and, consequently, 29 parameters must be identified. The plot of the model output and the actual system output is shown in Fig. 5. It can be seen that the model can track adequately the actual systems except when \( u(t) = -1 \).
The results obtained with the proposed method are compared with the ones reported in the literature in Table III. Nevertheless, most of the models in the literature use a delayed value of the actual output as an additional input in a prediction model. The model identified in the present work (in both examples) is a pure simulation model where actual outputs are not used as input.

**TABLE III**

<table>
<thead>
<tr>
<th>Model</th>
<th>MSE</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRFN [9]</td>
<td>0.0084</td>
<td>33</td>
</tr>
<tr>
<td>RSOFIN [9]</td>
<td>0.0780</td>
<td>36</td>
</tr>
<tr>
<td>RFNN [9]</td>
<td>0.0575</td>
<td>112</td>
</tr>
<tr>
<td>DFNN [11]</td>
<td>0.0025</td>
<td>39</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.0082</td>
<td>29</td>
</tr>
</tbody>
</table>

**IV. CONCLUSIONS**

This work has presented an identification algorithm for estimating the parameters of a TSK-type recurrent fuzzy system and then converts it into a linguistic type recurrent fuzzy system. The resulting linguistic recurrent fuzzy system can be encoded into a FFA. The identification algorithm is based on a general purpose GA, whose source code is available in the internet.

The results have shown that the proposed strategy can have reasonable performance comparing with other methods found in the literature. The main contribution of this work is that the identified fuzzy system can be encoded into a FFA. This is an important issue, since FFA can be used to further analyze the identified model. Furthermore this method can be a powerful tool for modeling systems with hybrid (discrete and continuous) behavior.

In the present work the structure of the fuzzy model was fixed off-line and was not optimized. Several methods are reported in the literature in which a GA is used to identify the fuzzy systems structure. An optimized structure can certainly achieve a better numerical performance. This is the direction of future works in this research project.

**ACKNOWLEDGMENT**

The authors are grateful to the Brazilian Research Council, CNPq for the financial support for this research.

**REFERENCES**