GA-optimized neuro-fuzzy approach for nonlinear system modeling

Behnam Mehrkian 1, Arash Bahar 1, Ali Chaibakhsh 2
1. Department of Civil Engineering, University of Guilan, Rasht, Guilan, Iran
2. Department of Mechanical Engineering, University of Guilan, Rasht, Guilan, Iran
behnam_mehrkian@webmail.guilan.ac.ir
bahar@guilan.ac.ir
chaibakhsh@guilan.ac.ir

Abstract
In order to characterize the behavior of nonlinear dynamic systems many different approaches have been proposed in recent years. One of the best black-box models employed to deal with system nonlinearities is the combination of artificial neural network (ANN) and fuzzy logic system (FLS), which is known as neuro-fuzzy system. However, the gradient-based nature of this combination causes some deficiencies. Therefore, in this paper, an optimization approach which utilizes genetic algorithm (GA) as a derivative-free optimizer is proposed for both designing the structure of neuro-fuzzy model and assessing the model parameters. The whole proposed approach is applied to approximate: first, a nonlinear plant; next, nonlinear dynamic behavior of magneto-rheological (MR) damper, which is widely used in semi-active control of structures and its identification is significantly difficult due to inherent hysteretic and highly nonlinear behavior of the device. Comparisons between the responses of the models and the reference data show high accuracy and feasibility of the proposed approach.

Keywords: fuzzy, genetic algorithm, backpropagation learning, MR damper, earthquake record.

1. INTRODUCTION
Soft computing platform emerged as a simple but effective technique for describing wide variety of plant behaviors with and without a priori knowledge about the plants. Fuzzy logic system (FLS) is an intelligent tool imitating the logical thinking of human and then is capable of approximating any continuous function. However, there is no systematic method to design and examine the number of rules, input space partitioning and membership functions (MFs). Meanwhile, artificial neural network (ANN), as a derivative-based approach mimics the biological information processing mechanisms. This technique modifies its behavior in response to the environment, and is ideal in case that the expected mapping algorithm is un-known and the tolerance to faulty input information is required. But because of its derivative nature, some deficiencies may occur. The other exploitation for adopting and optimizing fuzzy systems is employing derivative-free optimization methods such as genetic algorithm (GA) [1, 2], simulated annealing (SA) [3], the random optimization method [4], and so on. Although the randomness and stochastic nature of these algorithms are advantages, some unique features of derivative-based algorithms are not employed. In this paper, neural network capabilities and genetic algorithm attributes are tuned to fuzzy structure to predict the behavior of some nonlinear plants in a high level of accuracy. The paper is organized as follows: in section two, Takagi–Sugeno (T–S) fuzzy model and its learning paradigm are reviewed; next, backpropagation (BP) learning as a well-known part of the learning procedure is studied in section three; how genetic algorithm (GA) is employed in this paper is introduced in section four; proposed approach is explained in section five; finally, in order to evaluate the proposed approach, application examples (AE) are brought in section six.

2. TAKAGI–SUGENO FUZZY MODEL
The Takagi–Sugeno (T–S) fuzzy model is a system described by fuzzy IF–THEN rules which can give local linear representation of the nonlinear system by decomposing the whole input space into several partial fuzzy spaces and representing each output space with a linear equation. Such a model is capable of approximating a wide class of nonlinear systems. For the reason that it employs linear model in the consequent part, conventional linear system theory can be applied for the system analysis and synthesis accordingly. And hence, the T–S fuzzy models are becoming powerful engineering tools for modeling and control of complex dynamic systems [5]. To construct a fuzzy model choosing a strategy to partition the input space is needed. If we use the grid partitioning strategy, which is the commonest method, then the structure of fuzzy model could be expressed by the following fuzzy IF–THEN rules that, is shown in Figure 1.

Rule i: IF $x_1$ is $A_{i_1}^1$ and ⋯ and $x_{n_i}$ is $A_{i_n}^i$, THEN $y_i = a_{0_i} + a_{1_i} + ⋯ + a_{n_i}^i x_{n_i}$,  

(1)
where \( i = 1, \ldots, N_r \), \( N_r \) is the number of IF-THEN rules, \( x_i = [x_{i1}, \ldots, x_{iN_i}] \) the premise input variable, \( N_i \) the number of input variable, \( a_{ij}, j = 1, \ldots, N_i \), the consequent parameters, \( y_i \) an output from the \( ith \) IF–THEN rule, and \( A_{ij} \) is a fuzzy variable. Given the input \( x_i = [x_{i1}, \ldots, x_{iN_i}] \), the final output of the fuzzy model is inferred by a weighted mean defuzzification as follows:

\[
\hat{y} = \frac{\sum_{i=1}^{N_r} w_i y_i}{\sum_{i=1}^{N_r} w_i},
\]

where the weight \( w_i \) implies the overall truth value of the premise of the \( jth \) implication for the input, and is calculated as

\[
w_i = \prod_{j=1}^{N_i} \mu(A_{ij}),
\]

(if algebraic product is accepted as a T-norm operator for intersection (and), and fuzzy implication function) where \( \mu(A_{ij}) \) is the grade of the membership function (MF) of \( A_{ij} \). But the big question is how many membership functions should be assigned to each input.

The aforementioned architecture is also the form of adaptive neuro-fuzzy inference system (ANFIS), where fuzzy compartments constitute the nodes of the adaptive networks. In this sense, FLS utilizes the architecture to be trained. Hybrid learning procedure combining BP learning and least-squares estimator (LSE) is the most common method to train FLS. More specifically, in the forward pass of the hybrid learning algorithm, nodes outputs go forward until last layer and the consequent parameters are identified by LSE, while in the backward pass, the error signals propagate backward and the premise parameters are updated by BP learning. The whole procedure is summarized in Table 1. BP learning is based on gradient descent (GD) method, which needs the gradient information of the problem. Since it is gradient-based it causes some deficiencies, which is studied in the next section.

**Table 1- passes in the hybrid learning procedure for ANFIS [6]**

<table>
<thead>
<tr>
<th></th>
<th>Forward pass</th>
<th>Backward pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Premise parameters</td>
<td>Fixed</td>
<td>Backpropagation learning</td>
</tr>
<tr>
<td>Consequent parameters</td>
<td>LSE</td>
<td>Fixed</td>
</tr>
<tr>
<td>Signals</td>
<td>Node outputs</td>
<td>Error signals</td>
</tr>
</tbody>
</table>

### 3. BACKPROPAGATION LEARNING

BP learning [7] is the principle and complicated part of optimizing a neuro-fuzzy system. The method of steepest descent method, also known as gradient method, is one of the oldest techniques for minimizing a given function, which could be employed in BP learning. Although it has slow convergence, the method is the most frequently used nonlinear optimization technique due to its simplicity. In general, a given objective function may have a nonlinear form with respect to an adjustable parameter \( \theta \). Then the method by its iterative algorithm could update the current point \( \theta_{\text{now}} \) to next point \( \theta_{\text{next}} \) as follows:

\[
\theta_{\text{next}} = \theta_{\text{now}} - \eta g,
\]

Where \( \eta \) is some positive step size regulating to what extent to proceed in \(-g\) direction and \( g \) is the gradient of objective function. Since the well-known basic learning algorithm of the adaptive network is the steepest descent method, it would be instrumental to bring the other form of (4) that is obtained by normalizing the gradient:

\[
\theta_{\text{next}} = \theta_{\text{now}} - \kappa \frac{g}{\|g\|},
\]
where $\kappa$ is the real step size, indicating the Euclidean distance of the transition from $\theta_{\text{now}}$ to $\theta_{\text{next}}$:

$$\kappa = \|\theta_{\text{next}} - \theta_{\text{now}}\|.$$  \hfill (6)

There are two methods to employ $\kappa$: 1. Using a small fixed $\kappa$. 2. Updating $\kappa$. In the first method, a small value of $\kappa$ leads to an inefficient search unless it is already in the vicinity of the minimum, while a large value of $\kappa$ allows the search process to approach the minimum efficiently, but it then oscillates around the local minimum and cannot pinpoint it precisely as shown in Figure 2 (a). In contrast, the second method employs an adaptive strategy to adjust the step size dynamically. Based on empirical observations, the step size $\kappa$ can be updated according to the following two heuristic rules [8]:

1. If the objective function undergoes $m$ consecutive reductions, increase $\kappa$ by $p\%$.
2. If the objective function undergoes $n$ consecutive combination of one increase and one reduction, decrease $\kappa$ by $q\%$.

This updating strategy is incorporated in the hybrid learning in ANFIS. In particular, if we use the gradient vector in a simple steepest descent method, the resulting learning paradigm is often referred to as BP learning. Here, the important question is how much the $\kappa$, $p$, and $q$ would be? Are they fixed figures for all kind of problems? In fact, virtually no gradient-based descent algorithm is guaranteed to find the global optimum of complex objective function. In the other world, all descent methods are deterministic in the sense that they inevitably lead to convergence to the nearest local minimum [6]. Moreover, selecting initial position for deterministic methods clearly has a decisive effect on the final result. Figure 2 (b) shows how two close but different initial points (2, 3) and (2.5, 3) result in different minima. It means that if the initial point is suitable, we could employ the capability of gradient-based descent algorithm. Therefore, according to the mentioned deficiencies, a derivative-free optimization method is needed. One of the powerful methods is utilizing genetic algorithm, which is briefly introduced in next section.

4. **GENETIC ALGORITHM**

Genetic algorithm (GA), which was proposed for the first time by Holland from university of Michigan in 1975, is a free-derivative stochastic optimization method. Generally, there is no regular rule to adjust the GA’s operators [9]. In this paper GA is employed in a simple, effective and more general case. The following equation is used to estimate the population Size ($N$) [10]:

$$\frac{FE}{N} \log(1 + \frac{1}{N}) = -M - \log\left(\frac{l}{12}\right),$$  \hfill (7)

where $FE$ is the number of objective-function evaluation, $M$ is a constant ($M=3$ is suitable), and $l$ is the number of parameters. It is recommended that $FE$ can be assumed, because of time and memory limitation, from the range $[10^3, 10^4]$. Elite Operator=2 is employed; Uniform stochastic method is chosen for selection operator; for the crossover operator and the scattered method is used with crossover fraction = 0.8; and Gaussian with scale=1 and shrink=1 is used for the mutation operator. The Number of required generations to converge is chosen as the criteria to end GA, which is simply obtained by

$$g_{\text{converge}} = \frac{FE}{N}.$$  \hfill (8)

5. **PROPOSED APPROACH**

The proposed approach consists of two phases which are explained in the following.

- **Phase 1.** Assessing the structure of neuro-fuzzy system: In this phase, we know the number of inputs, and assume the maximum number of rules (maybe based on time limitation). The unknown parameters are: the number of MFs for any input individually (optimum number of rules); the optimum $\kappa$, $p$, and $q$ in BP learning method only for this phase.
Phase 2. Optimizing the resulted neuro-fuzzy structure: The premise and the consequent parameters of fuzzy rules should be adapted in this phase. Therefore, cooperation of GA and BP learning method results in a stochastic-deterministic approach.

In phase 1, the optimized number of rules is simply estimated based on grid partitioning strategy where we always encounter to the curse of dimensionality. Moreover, there is no guarantee in which the more MFs (rules) result in more accuracy. In fact, in this phase, when a fixed number of epochs (number of forward and backward passes for LSE and BP) is assumed, GA searches for the optimum number of MFs for any input individually and simultaneously the consistent properties of BP learning characteristics. Note that the number of epochs is in secondary place of importance because everything would be assigned adaptively. Since the BP learning parameters should be assessed consistently, phase 1 would not decrease the error just by increasing number of rules; it used the high capability of BP learning to gain the number of rules. Therefore, phase 1 can estimate the optimized number of rules with appropriate BP learning characteristics. In Phase 2, in fact, the parameters of MFs would be estimated twice: first, by GA; second, by BP learning. But, consequent parameters (the linear part) just would be estimated once by LSE. Since after all generations, thousands of initial MFs have been being checked, the procedure does not dependent on the gradient information of the problem. Moreover, since the initial points are estimated randomly, it means that they may be around the global minimum, so it would be in time to employ a GD method. The objective function has training and checking data error simultaneously, prevents the model to be over fitted. The other point is that it does not need to optimize the BP learning properties simultaneously anymore in phase 2, since the initial points are changing all the time. Finally, the resulted neuro-fuzzy structure is expected to have an excellent behavior, which is examined in the next section based on some well-known nonlinear systems.

6. APPLICATION EXAMPLES (AE)

6.1 FIRST AE

The first nonlinear equation, which has two input variables and one output, is as follow:

\[ z = \sin(x, y) = \frac{\sin x \sin y}{xy}. \]  

(9)

[6] employed this equation to compare the performance of ANFIS and a 2-18-1 multilayer perceptron (MLP) as an artificial neural network (ANN). In that work, from evenly distributed grid points of the input range \([-10,10] \times [-10,10]\) of the preceding equation, 121 training data pairs were obtained. The ANFIS structure, which [6] used, contained 16 rules with four MFs assigned to each input variable. Figure 3 shows the root mean squared error (RMSE) for both a 2-18-1 BP MLP and the ANFIS architecture used there.

Each curve is the result of averaging 10 error curves from 10 runs. For the MLP these 10 runs were started from different set of initial random weights. For ANFIS, these 10 runs correspond to 10 initial step size \(\kappa\) values ranging from 0.01 to 0.10. This, however, is validated implicitly the significant influence of the initial step size and initial points on the overall behavior of the system. On the other hand, there was not any explanation of why we should assume the initial step size in that way and, finally, contemplating the average amount. The BP MLP, which contained 73 fitting parameters (connected weights and thresholds), was trained with quick propagation [11], which considered one of the best learning algorithm for BP MLPs. The number of epochs in both methods was 250. Now we apply the proposed approach to this equation to compare with the aforementioned methods. Since [6] selected \(\text{MFs} = [4, 4]\), \(\text{gbellmf}\) as the MFs type, and first order of T-S fuzzy model, we assume 16 as the maximum number of rules and the same characteristics, in order to not to exploit any advantages in the proposed model. As the number of data pairs are small, number of epochs= 50 is assumed. A total five parameters, including two parameters for the number of MFs (for each input variable) and three parameters for initial step size \(\kappa\), \(p\), \(q\) would be estimated in phase 1. Due to the a few numbers of parameters, \(\text{FE}=10^3\) is assumed, so \(N=10\) and \(\text{gconverge}= 100\) are obtained from (7) and (8), respectively. \(\text{MFs} = [4, 3]\) is estimated from phase 1, so 21 parameters including premise nonlinear parameters should be assigned in phase 2. Therefore, assuming
FE=10^4 (due to more parameters), N=40 and g_{converge}= 250 are obtained. Table 2 contained the RMSE values for both the methods in [6] and the proposed approach. As shown in Table 2 the proposed approach with epoch= 50 and 12 rules has an excellent performance and it is superior to ANFIS with epoch= 250 and 16 rules. Figure 4 shows the initial and final MFs. In Figure 5 the best performance of the proposed approach in front of training data is shown. To compare ANFIS by proposed approach visually, their performance are shown in Figure 6 where data plotted as a function of samples. As it can be seen the proposed approach excellently mimics the training data while some errors are observable in ANFIS performance.

Figure 4. (a) and (b) MFs before learning, which are predicted by GA; (c) and (d) MFs after BP learning (1st AE)

Figure 5. (a) Training data output and (b) predicted output by the proposed approach; (c) training error changing during the whole epochs and (d) step size curve where $\kappa = 0.01$, $p= 0.9$ and $q = 1.1$ (1st AE)

It is interesting to bring the ANFIS results after more epochs. The original paper did not have it, so here we run ANFIS again. After some trial and error, epochs= 70000 (or exactly 69836) was found as the number of epochs after which the error not only doesn't decrease anymore but also becomes worse. On the other hand, the proposed model is also run for epochs= 250. The results are brought in Table 2. However, ANFIS stagnates but the proposed approach still has the capability to decrease the error. This is a significant feature of the proposed approach which is due to the randomness and derivative-free nature of GA.

### Table 2- Comparison the proposed approach with ANFIS (1st AE)

<table>
<thead>
<tr>
<th>Model</th>
<th>epochs</th>
<th>MFs</th>
<th>RMSE</th>
<th>epochs</th>
<th>MFs</th>
<th>RMSE</th>
</tr>
</thead>
</table>

### 6.2 SECOND AE

In this application example, it will be depicted that even if we omit BP learning from the proposed method, it still has well performance. To this end, we try to model MR damper, which is a semi-active control device that has recently received more attention by the vibration control community. But due to inherent hysteretic and highly nonlinear dynamics of MR fluid damper many parametric [12, 13] and nonparametric models [14], have been proposed.
In our simulation, data are generated by the base isolated benchmark (BM) program [15], which is used by the structural control community as a state-of-the-art model for numerical experience of seismic control attenuation. Therefore, four-column data consisting of displacement, velocity, voltage, and force of MR damper are produced by employing seven predefined earthquake records of the BM problem, and have been used as validation data. For training and checking data in the proposed approach, an Iranian earthquake record and a predefined earthquake record of the BM problem are employed. We used Nahavand earthquake record with peak ground acceleration (PGA) equal to 0.35, as the Iranian record for training data, and Newhall earthquake record, as a predefined BM record for checking data. Consequently, the model that is trained by Iranian earthquake will be validated by BM earthquakes records. The training and checking data consists of 9470 and 6000 data pairs. Training data are depicted in Figure 7. In both phases the number of epochs is equal to one. In phase 1, maximum number of rules= 25. MFs= [2 6 2] is obtained, which belongs to displacement, velocity, and voltage data, respectively. In phase 2, the resulted structure is introduced to the combination of GA and LSE. Consequently, 30 optimized nonlinear parameters and 96 optimized linear parameters would be obtained. Comparing the force which is predicted by the proposed approach (F), with the target force (FBM) from the BM program, is depicted in Figure 8 for checking data. This figure shows the excellent performance of the proposed approach.

In the relevant literatures, comparing nonparametric models with parametric ones rarely have been done. Although the parametric models have some deficiencies, because of employing strict and precise mathematical differential equation to describe the nonlinear and inherent dynamics of MR damper, they have high accuracy and are superior to nonparametric models. [13] and [16] are the two best parametric models for describing the behavior of MR damper, which they used normalized Bouc-Wen term for hysteresis phenomenon. In order to compare, we use the same performance index (PI) as [12] did. It is the 1-norm error (ε) which is defined as follows:

\[
\varepsilon = \frac{\| F_{BM} - F \|}{\| F_{BM} \|}, \quad \| F \| = \int_0^T \| f(t) \| dt.
\]

(10)

Here, we also modeled MR damper based on ANFIS method. The results are listed in Table 3.
Figure 7. (a) displacement; (b) velocity; (c) controller voltage; (d) generated force by applying Nahavand to BM program (2nd AE)

Figure 8. Comparison between the proposed approach-predicted force (F) and the target force (F_BM); force vs. (a) Time, (b) Displacement, (c) Velocity under Y direction of Newhall excitation (2nd AE)

Table 3 shows that the proposed model has better performance even in comparison with parametric models (the first two rows are from [13]). Therefore, although BP is omitted due to time limitation in large data, the proposed approach has the best performance. Furthermore, it should be noted that, though the Parametric 1 method has a good performance, it cannot be inversed. The model capability for being inversed is a vital issue when we want to apply the model to a control process, where an appropriate instant voltage signal should be sent to MR dampers during an earthquake. In contrast, the proposed model easily can be inversed just by interchanging the force and voltage location in the data that indicates the promising role of
the proposed approach in control process. Consequently, two preceding application examples indicate high accuracy of the proposed approach, which consists of GA and neuro-fuzzy system and can be categorized in soft computing techniques, to predict and mimic the plants behavior.

Table 3 - \( \epsilon \) (in terms of %), each cell contains X (top) and Y (bottom) direction (2nd AE)

<table>
<thead>
<tr>
<th>Method</th>
<th>Newhall</th>
<th>Sylmar</th>
<th>El Centro</th>
<th>Rinaldi</th>
<th>Kobe</th>
<th>Jiji</th>
<th>Erzinkan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric 1</td>
<td>6.47</td>
<td>5.67</td>
<td>7.78</td>
<td>7.12</td>
<td>6.52</td>
<td>3.61</td>
<td>4.88</td>
</tr>
<tr>
<td></td>
<td>3.84</td>
<td>8.44</td>
<td>7.90</td>
<td>5.67</td>
<td>7.85</td>
<td>4.02</td>
<td>5.35</td>
</tr>
<tr>
<td>Parametric 2</td>
<td>16.15</td>
<td>18.06</td>
<td>22.89</td>
<td>17.55</td>
<td>18.22</td>
<td>14.16</td>
<td>14.19</td>
</tr>
<tr>
<td></td>
<td>15.83</td>
<td>24.14</td>
<td>19.68</td>
<td>18.48</td>
<td>24.72</td>
<td>20.09</td>
<td>18.80</td>
</tr>
<tr>
<td>ANFIS</td>
<td>9.96</td>
<td>7.50</td>
<td>12.21</td>
<td>6.83</td>
<td>9.90</td>
<td>9.25</td>
<td>5.50</td>
</tr>
<tr>
<td></td>
<td>8.80</td>
<td>6.36</td>
<td>14.75</td>
<td>5.76</td>
<td>20.29</td>
<td>8.94</td>
<td>4.58</td>
</tr>
<tr>
<td>Proposed approach</td>
<td>4.36</td>
<td>3.04</td>
<td>5.60</td>
<td>2.92</td>
<td>5.66</td>
<td>5.70</td>
<td>2.58</td>
</tr>
<tr>
<td></td>
<td>4.13</td>
<td>2.83</td>
<td>7.66</td>
<td>2.52</td>
<td>12.86</td>
<td>4.53</td>
<td>2.12</td>
</tr>
</tbody>
</table>

7. REFERENCES


