Self-tuning Fuzzy Inference based on Spline-membership Function

Koji SHIMOJIMA, Toshio FUKUDA, and Fumihito ARAI
Dept. of Mechano-Informatics and Systems, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya 464-01, Japan
email: koji@mein.nagoya-u.ac.jp
fukuda@mein.nagoya-u.ac.jp
arai@mein.nagoya-u.ac.jp

Abstract
Recently, fuzzy systems are used in many fields and places. In order to apply the fuzzy system to wider fields, it is necessary to study the tuning methods of the fuzzy system. Some self-tuning methods were proposed so far. However these conventional self-tuning methods do not have sufficient capability of generalization. In this paper, we propose new self-tuning fuzzy inference. The fuzzy inference consist of membership functions that are expressed by spline function. Descent method is applied to tune the membership functions and consequent parts. The effectiveness of the proposed methods is shown by some numerical examples.

Key Word: Self-tuning fuzzy inference, Spline-membership function, descent method

1 Introduction

In recent years, fuzzy systems such as fuzzy reasoning, fuzzy modeling, and fuzzy logic controllers are utilized in many fields as engineering, medical engineering, and even social science. Some fuzzy control system already can be seen in home appliance, transportation system, and so on. We also have been studying about the sensor integration system applied fuzzy inference.[1, 2, and 3]

Fuzzy system has a characteristic to represent human knowledge by some fuzzy rules. However the fuzzy system has some problems. In most fuzzy systems, the shape of membership functions of the antecedent, the consequent, and fuzzy rules were decided and tuned through trial and error by operators and their experienced knowledge, therefore it takes many times to decide and tune them, and it is very difficult to design the optimal fuzzy system in detail. This problem is more serious, when the fuzzy controller is applied to the more complex system.

In order to solve this problem, some self-tuning methods have been proposed such as Fuzzy Neural Network[4] that is applied the neural network method[5], fuzzy learning controller applying Radial basis function[6 and 7], utilizing the genetic algorithm for deciding the shapes of membership functions and fuzzy rules [8], and so on [9].

These methods can learn faster than neural networks. However operator must decide the number and shapes of membership functions before learning, and the learning ability and accuracy of approximation are related to the number or shape of membership functions. Fuzzy inference with much membership function and fuzzy rules has high learning ability, however there are some redundant rules or unlearned rules. The number of rules is product of the number of membership function for each input, and the number of rules is increased as exponential with increase of the input number. Therefore operators must pay attention to decide the structure of the membership functions. For this problem, the hierarchical fuzzy inference has been proposed to reduce fuzzy rules. However this method also has two problems. One is that the number of rules is increased with increase of the input number. Other is that the operator must design the fuzzy inference considering the relation of each input because inputs are classified into some groups and the relation of each input is cut into off.

[7 and 8] utilize Radial Basis Function and make a fuzzy inference adding a new rule for the maximal error point through learning process. Therefore, fuzzy rule depends on the learning data set and if the learning data is biased, there are some unlearning area. These methods also have the increasing fuzzy rule problem and adding of fuzzy rules are cause of consuming the calculation time and memory. These methods do not integrate or delete a fuzzy rule, only add a new fuzzy rule.

Self-tuning fuzzy inference based on B-spline[10] has been proposed[11]. The characteristic of this method is that each membership function is utilized as B-spline and three membership functions is fired for one input and then the output is C[1] function. However for n input system, fired rules are increased as 3^n and it needs much calculation. Learning is only carried out for the consequent part, and the learning ability is depends on the initial state because of no adding knot.

In this paper, we propose a new type of self-tuning fuzzy inference. The membership function of the antecedent is expressed by the spline function. In[11], a membership function (B-spline) covers a part of the input space. On the contrary, the input space of each membership function covers the whole space of each input variable, thus this fuzzy inference can be constructed by less membership function and fuzzy rules and the initial state problem does not arise and it is able to learn in short time. In order to enhance the learning ability, this fuzzy inference adds/deletes a new knot that can make the spline function more complex shape. The added knot affects little to the calculation time of inference.

We describe the structure of the fuzzy inference based on spline function, it's learning method, the knot addition/deletion method, and show the results of numerical experiments.
2 Fuzzy inference based on spline function

2.1 Construction

The fuzzy inference based on spline function proposed here have the consequent with numerical value. Note that input variables are defined as \( x_1, x_2, \ldots, x_n \) and estimated result as \( y \), then the i-th rule of fuzzy inference is expressed as follows:

\[
\text{Rule}_i: \text{If } x_1 \text{ is } M_{i1} \text{ and } x_2 \text{ is } M_{i2} \text{ and } \ldots \text{ and } x_n \text{ is } M_{in}, \text{ then } y \text{ is } W_i \ (i = 1, 2, \ldots, m)
\]  

where \( W_i \) and \( M_{in} \) means numerical value of the consequent of the i-th fuzzy rule and the membership function for input variable \( x_n \) of the antecedent respectively.

The membership function of the antecedent \( M_{ij} \) is expressed by the natural cubic spline function, and each fuzzy rule consists of some membership functions individually. The natural cubic spline function can be expressed by the value, here the value is the grade of membership, and the second derivative on the knot with the simple equation (2) [12]. Thus grade of membership of the i-th rule \( M_{ij} \) against input \( x_j \) is the output \( u_{ij} \) of the natural cubic spline function which expresses \( M_{ij} \).

\[
\begin{align*}
\mu_{ij} &= m_{ijk-1} \frac{(x_j - x_{ijk-1})^2 (x_j - x_{ijk+1})}{h_{ijk}^3} + m_{ijk} \frac{(x_j - x_{ijk+1})^2 (x_j - x_{ijk-1})}{h_{ijk}^3} \\
&+ m_{ijk} \frac{(x_j - x_{ijk})^2 (x_j - x_{ijk+1}) + h_{ijk}}{h_{ijk}^3} \\
&+ m_{ijk} \frac{(x_j - x_{ijk+1})^2 (x_j - x_{ijk}) + h_{ijk}}{h_{ijk}^3} \\
&= x_j - x_{ijk-1}
\end{align*}
\]

where \( x_{ijk}, \mu_{ijk}, \text{ and } m_{ijk} \) means position of knot, grade of membership at \( x_{ijk} \), and the second order derivative at \( x_{ijk} \) respectively. Here, the grade of membership is limited as 0 to 1. Therefore a fitness of the antecedent of the i-th rule \( \mu_i \) is given by eq.(3).

\[
\mu_i = \mu_{i1} \cdot \mu_{i2} \cdots \mu_{in}
\]

then the result of estimation \( y \) is calculated by eq.(4).

\[
y = \frac{\sum_{i=1}^{m} \mu_i \cdot W_i}{\sum_{i=1}^{m} \mu_i}
\]

In ordinary fuzzy system, each rule does not cover the whole input space, because each membership function takes a part of the input domain. Our proposed membership function takes the whole input space. Therefore each rule of the fuzzy inference covers the whole input space.

Membership function of each rule implies the probability distribution map about the rule. Therefore it is difficult to express the probability distribution map like parity bit problem, because each learning data has contradictory learning data. The exclusive-or problem with 2-input and an output is an example. Now we select the learning data \((x_1, x_2)\) as \((0, 0), (0, 1), (1, 0), \text{ and } (1,1)\) and we use two rules (the consequent value of rule 1 is 0, rule 2 is 1. When we think about input \( x_1 \), the value is 0 at \( (0, 0) \), however the value is 1 at \( (0, 1) \) and they are contradictory. This contradiction can be seen on all learning point, therefore the fuzzy inference cannot make a membership function for each rule.

In order to express the parity bit problem, proposed fuzzy inference has two more membership functions that express the position information of input data. One membership function is about the angle \( \theta \) for 4-input given by eq.(5), and the other is about the distance \( D \) that is given by eq.(6). For these membership functions, the proposed fuzzy inference can learn the parity bit problem.

\[
\theta = \tan^{-1}(\tan^{-1}(x_1, x_2), \tan^{-1}(x_3, x_4))
\]

\[
D = \sqrt{\sum_{i=1}^{n} (x_{ic} - x_i)^2}
\]

where \( x_{ic} \) means the center value of the input variable \( x_j \). Membership function for \( \theta \) is expressed by the cubic periodic spline function, others are expressed by the natural cubic non periodic spline function. Figure 1 shows the structure of the fuzzy inference based on the spline function and Fig. 2 shows the calculation way of the angle information \( \theta \).
2.2 Learning law

Learning of membership functions of the antecedent and values of the consequent are conducted by the descent method. Eq.(7) defines the error function.

\[ E_p = \frac{1}{2} (y_p - y_p^*)^2 \]  

(7)

where \( y_p \) and \( y_p^* \) means the output of the fuzzy inference for the p-th learning data and the p-th learning data.

The consequent of the i-th rule \( W_i \) is refined by the partial differential of eq.(7) by \( W_i \) as follows,

\[ \Delta W_i = -\alpha \frac{\partial E_p}{\partial W_i} = -\alpha \frac{\partial E_p}{\partial y_p} \frac{\partial y_p}{\partial x_{ij}} \frac{\partial x_{ij}}{\partial W_i} = -\alpha (y_p - y_p^*) \frac{\mu_i}{\sum_{i=1}^{m} \mu_i} \]  

(8)

where \( \alpha \) means the learning rate of the consequent.

Learning of the antecedent is conducted by refining the knots of the spline function which satisfy \( x_{ijk} \leq x_j \leq x_{ijk+1} \). The learning equations of \( \mu_{ijk} \) and \( \mu_{ijk+1} \) are defined by partial differentiate eq.(7) by \( \mu_{ijk} \) and \( \mu_{ijk+1} \) as follows,

\[ \Delta \mu_{ijk} = \beta \frac{\partial E_p}{\partial \mu_{ijk}} = \beta \frac{\partial E_p}{\partial y_p} \frac{\partial y_p}{\partial \mu_{ijk}} \]  

\[ \Delta \mu_{ijk+1} = \beta (y_p - y_p^*) \frac{W_i - y_p}{\sum_{i=1}^{m} \mu_i} \mu_{ij} \frac{x_j - x_{ijk}}{h_{ijk}} \]  

(9)

\[ \mu_{ijk+1} = \mu_{ijk} \]  

\[ \mu_{ijk+1} = \mu_{ijk} - \beta (y_p - y_p^*) \frac{W_i - y_p}{\sum_{i=1}^{m} \mu_i} \mu_{ij} \frac{x_j - x_{ijk}}{h_{ijk}} \]  

(10)

The learning equations of \( x_{ijk} \) and \( x_{ijk+1} \) are defined as partial differentiate eq.(7) by \( x_{ijk} / x_{ijk+1} \) as follows,

\[ \Delta x_{ijk} = \gamma (y_p - y_p^*) \frac{W_i - y_p}{\sum_{i=1}^{m} \mu_i} \mu_{ij} \]  

\[ \Delta x_{ijk+1} = \gamma (y_p - y_p^*) \frac{W_i - y_p}{\sum_{i=1}^{m} \mu_i} \mu_{ij} \]  

\[ x_{ijk+1} = x_{ijk} - \frac{x_j - x_{ijk}}{h_{ijk}} \left( \mu_{ijk} - \mu_{ijk+1} \right) \]  

\[ x_{ijk+1} = x_{ijk} - \frac{x_j - x_{ijk}}{h_{ijk}} \left( \mu_{ijk} - \mu_{ijk+1} \right) \]  

(11)

(12)

where \( \beta \) and \( \gamma \) are the learning late of \( \mu_{ijk} / \mu_{ijk+1} \) and \( x_{ijk} / x_{ijk+1} \).

2.3 Knot addition/deletion method

The proposed fuzzy inference consists of the spline function. The shapes of spline function depend on knots. The learning law of the section 2.2 is applied for changing the value and the position of the knots. However the number of knots limits the ability of the expression, thus the learning ability also is limited.

For this problem, the fuzzy inference adds or deletes knots of spline function in order to improve the ability of expression of membership function and the learning ability, not adds a new fuzzy rule.

Knot addition is carried out when the changing of mean square error (eq.(13)) is small, i.e. eq.(14) is satisfied for \( k \) times continuously. The position of the additional knot is the maximal error of the learning data and the value is the output of the spline function.

\[ E_t = \frac{1}{P} \sum_{p=1}^{P} (y_p - y_p^*)^2 \]  

(13)

\[ \left| \frac{E_{t-1} - E_t}{E_{t-1}} \right| \leq e \]  

(14)

where \( t \) means iteration time, \( k \), and \( e \) are the constant value set before the learning.

By the learning process or the additional knot process, if there is a knot which cannot be carried out the learning, i.e., the knot is not used through a learning cycle, and the deletion of the knot is not affect the mean square error, the knot is deleted except knots of both ends.

2.4 Learning algorithm

In this paper, we propose two different initial conditions of Fuzzy Neural Networks as follows;

(1) Shape of membership functions of the antecedent is initialized as the grade is 0.5 in any point as Fig. 3 and define the consequent value of each rule's.

(2) Shape of membership function is initialized as Fig. 5. The consequent is initialized as 0.0.

The number of rules of initial state(1) is that of the operator decided before learning. The minimal number is two from eq.(4). For initial state(2), the number of rules is product of each divided number of input space. For example, for 2-input system and each input space is divided into three parts, the number is nine.

Learning is carried out by two steps. For initial state (1), the antecedent is tuned using eqs. (9)-(12) as the first step. When the total error almost converges or the total error is smaller than the objective value of the first step, both the antecedent and the consequent is tuned simultaneously as the second step. For state (2), the first step is tuning of the consequent using eq.(9). The second step is the same as in case of initial state(1). Figure 5 shows the learning algorithm.
3. Simulation results

In order to show the effectiveness of the proposed fuzzy inference, we apply the inference to identify the 2-input 1-output nonlinear function described by eq.(15) and shown in Fig.6. The learning data consist of 11 by 11 points at intervals on the input space, and the evaluation data consist of 21 by 21 points at intervals. Learning process repeats till the mean square error becomes smaller than 0.001.

The proposed inference which consist of 2 rules, where each rule has 4 membership functions with 5 knots respectively, was set to the initial state (1) and leaning was carried out 12 times' iteration. The mean square error for the learning data is $8.67 \times 10^{-4}$, the maximum square error is $8.79 \times 10^{-2}$. Fig.7 shows the learning completed membership function and Fig.8 shows the output of the proposed inference for the evaluation data. For the initial state(2), 4 rules with 2 membership functions consist of 5 knots was employed and 35 times' iterations was carried out. The mean square error was $9.79 \times 10^{-4}$ and the maximum error was $8.43 \times 10^{-2}$. Fig.9 shows one of the learning completed membership function and Fig.10 shows the output of the inference for the evaluation data. There appears few differences in among Fig.6, Figs.8, and 10. We also apply the inference to identify the 2-input 1-output function described by eqs.(16) to (21), the 3-input 1-output function described by eqs.(22) to (24) compared with Fuzzy Neural Networks (FNN) [5] and the Gaussian function based fuzzy inference. FNN is tuned by changing parameters of the sigmoidal function (the antecedent part) and the consequent values. The Gaussian fuzzy inference is
also tuned by changing parameters of the Gaussian function (the antecedent part) and the consequent values.

At first, in order to examine the basic learning ability of each fuzzy inference, we carried out the identification and estimation test by eqs.(16) to (18). We use sequential and random set data for the identification and estimation test data. The number of identification data with random set is 50 and that of estimation data with random set is 300. The identification data with sequential set consists of 7 by 7, the estimation data consists 21 by 21. The input range is 0.0 to 1.0.

The proposed fuzzy inference consists of two rules, two membership functions with three knots (18 parameters). FNN consists 3 by 3 rules (25 parameters). Gaussian fuzzy inference consists 3 by 3 rules (21 parameters). Learning rates of the proposed fuzzy inference are $\alpha = 0.2$, $\beta = 0.02$, and $\gamma = 0.02$, and the additional / deletion knot process is not carried out. Those of FNN are 0.02 for the antecedent and 0.2 for the consequent, and of Gaussian fuzzy inference are 0.001 and 0.1.

Iteration is carried out 500 times. Table 1 shows the results of the identification and estimation test after 500 iterations and value of AIC[13]. In table, MSE means a mean square error. Equation (25) expresses the AIC, where $E$, $N$, and $k$ means mean square error, the number of data set, and the number of controllable parameters, respectively.

Table 1 shows that the proposed fuzzy inference is the smallest AIC's value for each equation except eq.(18) in case of the random set. For eq.(18), FNN is the best value and Gaussian fuzzy inference and the proposed fuzzy inference are almost same. In case of sequential data, the learning ability of the proposed method is superior to the Gaussian fuzzy inference. Comparing with FNN, the learning ability of the proposed method is equal or over FNN. From the viewpoint of mean square error, the proposed method is smaller than others. Thus, the proposed method has high learning ability with a few controllable parameters.

In order to show the effectiveness of the knot addition / deletion, we used more complex learning model expressed eqs. (19) to (24). In this case, we use two types of learning sets, i.e., sequential data set and random data set. In case of the random set, the number of data set is 20 (2-input equation) and 30 (3-input). Sequential data set for the 2-input equation consists of 11 by 11 data points, 9 by 9 by 9 data points for the 3-input equation. Learning is carried out until the mean square error converges under 0.001 for the random data set, 0.0001 for the sequential data set, or iteration reaches 500 times.

The proposed fuzzy inference with initial state(1) consists of two rules, eight membership functions. The number of knots for each input is 6, that of the distance and angle information is 11(2-input), 16(3-input). For initial state(2), the input space is divided into 3, and each membership consists of 5 knots. The membership function for the distance and angle information consists of 9 knots (2-input), 13 knots (3-input). FNN and the Gaussian fuzzy inference consist 4 by 4 rules. Learning rates of the proposed fuzzy inference are $\alpha = 0.2$, $\beta = 0.02$, $\gamma = 0.02$, $e = 0.1$, and $k = 5$. Those of FNN are 0.02 for the antecedent and 0.2 for the consequent, and of the Gaussian fuzzy inference are 0.001 and 0.1.
In this paper, we proposed a new fuzzy inference that consists of some membership function expressed by the cubic spline function. We showed its structure, and its learning methods as follows:

1. Before learning, the value of the consequent is set to the operator's desired value, then fuzzy inference makes the shapes of membership function through the learning process.
2. Before learning, the operator decides the shapes of each membership function. Then fuzzy inference modifies the consequent and the antecedent through the learning process.

We also show the effectiveness of the proposed method as follows:

1. For the fuzzy inference with initial state (1), the number of rules is free from the number of input. The inference can learn an object quickly except discontinuous function as eq. (21) or (24).
2. For the fuzzy inference with initial state (2), the operator can design the fuzzy inference based on his/her knowledge. The inference also can learn fast and its learning ability is higher than other fuzzy inference with the same number of rules.
3. The knot addition / deletion process improves the learning ability, higher and more accurate.

We also showed a part of learning ability through the simulation experiments.

References


### Table 1: Basic Learning Ability

<table>
<thead>
<tr>
<th>Equation</th>
<th>Data set</th>
<th>Random</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RBF</td>
<td>FNN</td>
<td>Proposed</td>
<td>RBF</td>
<td>FNN</td>
<td>Proposed</td>
<td>RBF</td>
<td>FNN</td>
<td>Proposed</td>
<td>RBF</td>
<td>FNN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\times 10^{-4}$</td>
<td>AIC</td>
<td>$\times 10^{-4}$</td>
<td>AIC</td>
<td>$\times 10^{-4}$</td>
<td>AIC</td>
<td>$\times 10^{-4}$</td>
<td>AIC</td>
<td>$\times 10^{-4}$</td>
<td>AIC</td>
<td>$\times 10^{-4}$</td>
</tr>
<tr>
<td>6.16</td>
<td>Identity</td>
<td>72.4</td>
<td>204.4</td>
<td>39.2</td>
<td>227.1</td>
<td>40.9</td>
<td>238.9</td>
<td>135</td>
<td>169.0</td>
<td>64.2</td>
<td>197.4</td>
<td>59.8</td>
</tr>
<tr>
<td>6.17</td>
<td>Identity</td>
<td>1.84</td>
<td>-378.9</td>
<td>0.88</td>
<td>-416.9</td>
<td>0.65</td>
<td>-446.0</td>
<td>2.35</td>
<td>-367.4</td>
<td>0.00</td>
<td>-1755</td>
<td>0.88</td>
</tr>
<tr>
<td>6.18</td>
<td>Identity</td>
<td>1.90</td>
<td>-386.4</td>
<td>0.67</td>
<td>-430.5</td>
<td>3.82</td>
<td>-357.5</td>
<td>2.44</td>
<td>-365.6</td>
<td>0.00</td>
<td>-1191</td>
<td>1.96</td>
</tr>
<tr>
<td>6.16</td>
<td>Estimate</td>
<td>88.0</td>
<td>46.3</td>
<td>-</td>
<td>42.0</td>
<td>-</td>
<td>116</td>
<td>-</td>
<td>42.5</td>
<td>-</td>
<td>56.7</td>
<td>-</td>
</tr>
<tr>
<td>6.17</td>
<td>Estimate</td>
<td>2.24</td>
<td>3.67</td>
<td>-</td>
<td>0.70</td>
<td>-</td>
<td>2.00</td>
<td>-</td>
<td>4.07</td>
<td>-</td>
<td>0.55</td>
<td>-</td>
</tr>
<tr>
<td>6.18</td>
<td>Estimate</td>
<td>3.16</td>
<td>2.05</td>
<td>-</td>
<td>4.05</td>
<td>-</td>
<td>2.17</td>
<td>-</td>
<td>3.04</td>
<td>-</td>
<td>1.23</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 2: Learning Results of the Proposed Method for Random Data Set

<table>
<thead>
<tr>
<th>Equation</th>
<th>Initial state (1)</th>
<th>Initial state (2)</th>
<th>Without knot addition/deletion</th>
<th>Initial state (1)</th>
<th>Initial state (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iterations</td>
<td>MSE</td>
<td>MAE</td>
<td>Iterations</td>
<td>MSE</td>
</tr>
<tr>
<td>6.16</td>
<td></td>
<td>54</td>
<td>9.92</td>
<td>0.0499</td>
<td>39</td>
</tr>
<tr>
<td>6.17</td>
<td></td>
<td>32</td>
<td>9.14</td>
<td>0.0430</td>
<td>26</td>
</tr>
<tr>
<td>6.18</td>
<td></td>
<td>112</td>
<td>9.92</td>
<td>0.0617</td>
<td>42</td>
</tr>
<tr>
<td>6.19</td>
<td></td>
<td>45</td>
<td>9.99</td>
<td>0.0277</td>
<td>99</td>
</tr>
<tr>
<td>6.20</td>
<td></td>
<td>36</td>
<td>9.96</td>
<td>0.0531</td>
<td>61</td>
</tr>
<tr>
<td>6.21</td>
<td></td>
<td>500</td>
<td>-</td>
<td>-</td>
<td>98</td>
</tr>
<tr>
<td>6.22</td>
<td></td>
<td>22</td>
<td>9.61</td>
<td>0.1782</td>
<td>46</td>
</tr>
<tr>
<td>6.23</td>
<td></td>
<td>46</td>
<td>6.43</td>
<td>0.0346</td>
<td>59</td>
</tr>
<tr>
<td>6.24</td>
<td></td>
<td>500</td>
<td>-</td>
<td>-</td>
<td>96</td>
</tr>
</tbody>
</table>

### Table 3: Learning Results of the Proposed Method for Sequential Data Set

<table>
<thead>
<tr>
<th>Equation</th>
<th>Initial state (1)</th>
<th>Initial state (2)</th>
<th>Without knot addition/deletion</th>
<th>Initial state (1)</th>
<th>Initial state (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iterations</td>
<td>MSE</td>
<td>MAE</td>
<td>Iterations</td>
<td>MSE</td>
</tr>
<tr>
<td>6.16</td>
<td></td>
<td>38</td>
<td>9.60</td>
<td>0.0343</td>
<td>56</td>
</tr>
<tr>
<td>6.17</td>
<td></td>
<td>36</td>
<td>9.96</td>
<td>0.0531</td>
<td>61</td>
</tr>
<tr>
<td>6.18</td>
<td></td>
<td>500</td>
<td>-</td>
<td>-</td>
<td>98</td>
</tr>
<tr>
<td>6.19</td>
<td></td>
<td>22</td>
<td>9.61</td>
<td>0.1782</td>
<td>46</td>
</tr>
<tr>
<td>6.20</td>
<td></td>
<td>46</td>
<td>6.43</td>
<td>0.0346</td>
<td>59</td>
</tr>
<tr>
<td>6.21</td>
<td></td>
<td>500</td>
<td>-</td>
<td>-</td>
<td>96</td>
</tr>
<tr>
<td>6.22</td>
<td></td>
<td>44</td>
<td>6.43</td>
<td>0.0346</td>
<td>59</td>
</tr>
<tr>
<td>6.23</td>
<td></td>
<td>500</td>
<td>-</td>
<td>-</td>
<td>96</td>
</tr>
<tr>
<td>6.24</td>
<td></td>
<td>500</td>
<td>-</td>
<td>-</td>
<td>96</td>
</tr>
</tbody>
</table>

### Table 4: Learning Results of the Gaussian fuzzy inference and FNN

<table>
<thead>
<tr>
<th>Equation</th>
<th>Data set</th>
<th>RBF</th>
<th>FNN</th>
<th>RBF</th>
<th>FNN</th>
<th>RBF</th>
<th>FNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iterations</td>
<td>MSE</td>
<td>MAE</td>
<td>Iterations</td>
<td>MSE</td>
<td>MAE</td>
<td>Iterations</td>
</tr>
<tr>
<td>6.19</td>
<td></td>
<td>14</td>
<td>9.91</td>
<td>0.0742</td>
<td>14</td>
<td>9.02</td>
<td>0.0497</td>
</tr>
<tr>
<td>6.20</td>
<td></td>
<td>9</td>
<td>9.82</td>
<td>0.0982</td>
<td>11</td>
<td>9.94</td>
<td>0.0397</td>
</tr>
<tr>
<td>6.21</td>
<td></td>
<td>281</td>
<td>9.66</td>
<td>0.0977</td>
<td>16</td>
<td>8.98</td>
<td>0.0898</td>
</tr>
<tr>
<td>6.22</td>
<td></td>
<td>24</td>
<td>9.55</td>
<td>0.0836</td>
<td>26</td>
<td>9.66</td>
<td>0.0860</td>
</tr>
<tr>
<td>6.23</td>
<td></td>
<td>31</td>
<td>9.94</td>
<td>0.1008</td>
<td>17</td>
<td>9.98</td>
<td>0.1074</td>
</tr>
<tr>
<td>6.24</td>
<td></td>
<td>124</td>
<td>9.51</td>
<td>0.1028</td>
<td>16</td>
<td>8.38</td>
<td>0.0894</td>
</tr>
</tbody>
</table>