# Sparse Fuzzy Systems Generation and Fuzzy Rule Interpolation: A Practical Approach

Chong[1], T.D. Gedeon[1], Sz. Kovacs[2], L.T. Koczy[1,3,4]

[1]School of Information Technology
Murdoch University
Western Australia
Email: {cchong,gedeon}@murdoch.edu.au

[2]Department of Information Technonology
University of Miskolc
HUNGARY
Email: szkilv@gold.uni-miskolc.hu

[3]Department of Telecom & Telematics
Budapest University of Technology and Economics
HUNGARY
Email: koczy@ttt.bme.hu

[4]Institute of Information Technology and Electrical Engineering, Széchenyi István University, Győr
HUNGARY

***Abstract*-In this paper, we explore the use of a sparse fuzzy system generation technique in conjunction with simple projection-based fuzzy rule interpolation, to generate sparse fuzzy systems with relatively few rules whilst still achieving reasonable system accuracy. Through setting a parameter value, the user is able to control, to some extent, the number of rules generated by the rule extraction technique. The rule interpolation approach enables the sparse fuzzy system to maintain a reasonable accuracy. The effectiveness of this approach is validated experimentally.**

## I. Introduction

Fuzzy Systems suffer from rule explosion. To model a system with $k$ variables and maximum $T$ fuzzy terms in each dimension, the number of necessary rules is:

$$\text{Number of rules} = O(T^K) \qquad \text{Eqn 1}$$

Which will be very large if $k$ is not very small. Because of this, fuzzy systems are limited to only very few variables. To widen the range of problems controllable by fuzzy rule-bases, it is essential to reduce $T$, $K$, or both in (Eqn 1). Hierarchical fuzzy systems [1] are proposed to reduce $K$. On the other hand, decreasing $T$ leads to sparse fuzzy systems, i.e. fuzzy rule-bases with "gaps" between the rules. In such fuzzy systems, there often exist observations (inputs) that do not match any of the rule antecedents. Fuzzy rule-interpolation [2] is proposed to infer the conclusions for such observations.

Since the pioneering work of fuzzy rule-interpolation [2], much research has been carried out in improving and extending the technique [3-8]. Some techniques are designed for both interpolation as well as extrapolation [4, 9] [2]. Apart from [4], which has been used to control an automatically guided vehicle, most of the work in the literature discusses rule interpolation at a theoretical level. There has been little report on the application of the technique in fuzzy rule-bases. In [10], a rule extraction technique for sparse fuzzy system generation has been designed. It was shown in the paper that the technique is able to generate relatively few rules to achieve a reasonable accuracy. The potential of the technique to be used in conjunction with fuzzy rule interpolation was however not explored. This issue is investigated in this paper.

As part of the study, a straightforward projection-based approach to fuzzy rule interpolation is also proposed.

In the next section, some theoretical discussion of fuzzy rule-interpolation is presented. Section III reports on sparse fuzzy systems generation. The fuzzy Rule Interpolation based on Conservation of Fuzziness (RICF) is discussed in section IV. The technique is extended to a projection-based approach in the subsequent section. Section VI reports on the experiments performed to validate the effectiveness of the sparse fuzzy system generation technique used in conjunction with the proposed projection-based fuzzy rule interpolation. The conclusion is presented in the last section.

## II. Fuzzy Rule Interpolation

A fuzzy rule $R_i$ comes in the following form:
If $X$ is $A_i$ then $Y$ is $B_i$

Where $X = \{x_1, x_2, \ldots, x_n\}$ is the input, $Y$ is the output, $A_i$ and $B_i$ are fuzzy sets of the antecedent and consequent of the rule respectively. Each rule defines a patch in the ($X$ x $Y$) space. A fuzzy system with $r$ rules is a sparse fuzzy system if (Eqn 2) holds:

$$X - \bigcup_{i=1}^{r} Supp\ (A_i) \neq \theta \qquad \text{Eqn 2}$$

Where Supp is the support and $\theta$ is the empty set. That is, there are "gaps" in the universe of discourse where no rule antecedent is present to any positive degree. The advantage of such sparse systems is the reduced complexity. If $T$ in (Eqn 1) is reduced in every dimension at least by a factor of $s$, then the number of rules is $|R| = O((t/s)^k)$. The reduced complexity comes with a price – loss of information. With the absence of rules, results for data that falls in the gaps cannot be inferred directly. To cope with this situation, a family of new reasoning techniques for sparse fuzzy systems, known as fuzzy rule interpolation, has been introduced [2]. Some theoretical discussion of fuzzy interpolation techniques are presented next.

This research was supported by the National Scientific Research Fund OTKA T034233 and T034212, a Main Research Direction Grant 2002 by Széchenyi István University, and the Australian Research Council.

The basic idea of rule interpolation is formulated in the Fundamental Equation of Rule Interpolation (FERI) [2]:

$$D(A^*,A_1) : D(A^*,A_2) = D(B^*, B_1) : D(B^*, B_2) \quad \text{Eqn 3}$$

Where $A^*$ and $B^*$ denote the observation and the corresponding conclusion, $R_1 = A_1 \rightarrow B_1$, $R_2 = A_2 \rightarrow B_2$ are the rules to be interpolated (flanking rules), such that

$$A_1 < A^* < A_2 \quad \text{Eqn 4}$$

$$B_1 < B_2. \quad \text{Eqn 5}$$

Here, $D$ denotes some distances or degree of similarity between two fuzzy sets and the symbol '<' denotes a partial ordering between fuzzy sets (more details in [2]). Different choices of the distance $D$ results in different types of interpolation. In [2], the distance was introduced as the set of lower and upper α–cut distances, describing the relative position of two comparable convex and normal fuzzy sets (CNF-sets) unambiguously. The conclusion, after decomposing (Eqn 3) to every $\alpha \in [0,1]$, it can be solved for $B^*$ as:

$$\inf\{B^*_\alpha\} = \frac{\frac{1}{D_{\alpha L}(A_{1\alpha}, A^*_\alpha)}\inf\{B_{1\alpha}\} + \frac{1}{D_{\alpha L}(A_{2\alpha}, A^*_\alpha)}\inf\{B_{2\alpha}\}}{\frac{1}{D_{\alpha L}(A_{1\alpha}, A^*_\alpha)} + \frac{1}{D_{\alpha L}(A_{2\alpha}, A^*_\alpha)}} \quad \text{Eqn 6}$$

$$\sup\{B^*_\alpha\} = \frac{\frac{1}{D_{\alpha U}(A_{1\alpha}, A^*_\alpha)}\sup\{B_{1\alpha}\} + \frac{1}{D_{\alpha U}(A_{2\alpha}, A^*_\alpha)}\sup\{B_{2\alpha}\}}{\frac{1}{D_{\alpha U}(A_{1\alpha}, A^*_\alpha)} + \frac{1}{D_{\alpha U}(A_{2\alpha}, A^*_\alpha)}} \quad \text{Eqn 7}$$

Where *inf* is the infimum and *sup* is the supremum. The main disadvantage of this method is that (Eqn 6) and (Eqn 7) often result in abnormal membership functions for $B^*$ that need further transformation to obtain a useful result. There are many ways to alleviate this problem [3-6]. In this paper, we choose to investigate [3] for its simplicity.

A majority of the rule-interpolation research focuses on the generation of a conclusion using two flanking rules that are closest to the observation and at the same time satisfy the conditions (Eqn 4) and (Eqn 5). In general, the closest flanking rules can be found by trying each possible pair of rules and selecting the closest pair. The time complexity involved in finding the closest flanking rules among $R$ rules in the multi-dimensional space is $O(R^2)$.

In [2] the possible extension of (Eqn 6) and (Eqn 7) to use $N$ rules instead of two is reported. The extended formulae for the new interpolation method are obtained by using the weighted average:

$$\min\{B^*_\alpha\} = \frac{\sum_{i=1}^{N}\frac{1}{D_{\alpha L}(A_{i\alpha}, A^*_\alpha)}\inf\{B_{i\alpha}\}}{\sum_{i=1}^{N}\frac{1}{D_{\alpha L}(A_{i\alpha}, A^*_\alpha)}} \quad \text{Eqn 8}$$

$$\max\{B^*_\alpha\} = \frac{\sum_{i=1}^{N}\frac{1}{D_{\alpha U}(A_{i\alpha}, A^*_\alpha)}\sup\{B_{i\alpha}\}}{\sum_{i=1}^{N}\frac{1}{D_{\alpha U}(A_{i\alpha}, A^*_\alpha)}} \quad \text{Eqn 9}$$

The use of all rules for interpolation relieves the fuzzy system from the task of flanking rules identification. Apart from that, it can be observed from (Eqn 8) and (Eqn 9) that even extrapolation is possible. That is, given an observation that does not fire any rule, a conclusion can be inferred from the existing rules without requiring the conditions (Eqn 4) and (Eqn 5) to be satisfied.

## III. Sparse Fuzzy System Generation

In the earlier works, the idea of complexity reduction using rule interpolation was mainly in the direction of reducing a dense rule base to a sparse one. The main idea is that for a fuzzy system with $r$ rules, $F = \{R_i: i = 1, \ldots, r\}$, a reduced sparse fuzzy system $F' = \{R_{ik}: i_k \in I \subset \{1, \ldots, r\}, |I| < r\}$ is created by elimination of some less important rules. Here, $I$ is the index set. The aim is to produce a proper compression of $F$ such that keep $|RI_{R'}(A_j) - B_j| \leq \varepsilon$ (the error tolerance) where $RI_{R'}(A_j)$ denotes the conclusion generated by applying interpolation method $RI$ on the rule subset $R'$ and substituting the observation $A_i$ into the approximation function. The rule elimination process is not straightforward. There is generally no monotonicity in the omission of rules, that is, having three sets of rules: $R'' \subset R' \subset R$ maybe $R''$ is a proper compression of $R$ while $R'$ is not. Thus, no computationally effective serial search algorithm for the best reduction can be constructed. The reduction of a dense fuzzy system to a sparse fuzzy system is often performed on a heuristic trial and error basis. Sparse fuzzy system generation by dense fuzzy system reduction is ineffective with high dimensional data since the real problem lies in the generation of the dense rule base itself. Therefore this technique is only suitable for complexity reduction given an existing dense fuzzy system.

In [10], a rule extraction technique for sparse fuzzy system generation has been designed (details later). In this paper, we explore the potential of the technique to be used in conjunction with fuzzy rule interpolation. Given a set of training data, the technique [10] first clusters the output space. Data points from each output cluster are projected back to each input dimension forming one-dimensional clusters. The clusters from different dimensions are then merged to form fuzzy rules. The algorithm is as follow:

1. Perform Fuzzy c-Means clustering [11] on the output space. The algorithm iteratively searches for a set of cluster centers that represent the structure of the data as best as possible by minimizing (Eqn 10).

$$J_m(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{c}(U_{ik})^m \| x_k - v_i \|^2, \ 1 \leq m \leq \infty$$

Eqn 10

where $J_m(U,V)$ is the sum of squared error for the set of fuzzy clusters represented by the membership matrix $U$, and the associated set of cluster centers $V$. Here, $\|x_k - v_i\|^2$ represents the Euclidean distance between the data $x_k$ and the cluster center $v_i$. At each iteration, the cluster centers are calculated using (Eqn 11) and (Eqn 12).

$$U_{ik} = \left( \sum_{j=1}^{c} \left( \frac{\| x_k - v_i \|}{\| x_k - v_j \|} \right)^{2/(m-1)} \right)^{-1} \forall i, \forall k, \quad \text{Eqn 11}$$

and

$$v_i = \frac{\sum_{k=1}^{n} (U_{ik})^m x_k}{\sum_{k=1}^{n} (U_{ik})^m}, \quad \text{Eqn 12}$$

The optimal number of clusters in the data is determined by means of the FS index [12] as follows:

$$S(c) = \sum_{k=1}^{n} \sum_{i=1}^{c} (U_{ik})^m (\|x_k - v_i\|^2 - \|v_i - \bar{x}\|^2) \quad 2<c<n$$

Eqn 13

The number of clusters, $c$, is determined so that $S(c)$ reaches a local minimum as $c$ increases.

2. For each fuzzy output cluster $B_i$ approximated, all the points belonging to the cluster are projected back to each of the input dimensions. For each dimension, fuzzy clustering is again applied to the projection of the points. In this step, the FS index (Eqn 13) is used in conjunction with the merging index [13]:

$$P(v) = \sum_{j=1}^{n} e^{-4\left\| (v - x_j) / \frac{(v_i - v_j)}{2} \right\|^2} \quad \text{Eqn 14}$$

For each pair of cluster centers $v_i$ and $v_j$, the index (Eqn 14) is calculated and the clusters are merged if $p(v_m)$ is smaller than both $p(v_i)$ and $p(v_j)$, where $v_m$ is the middle point $(v_i + v_j)/2$.

3. The previous step results in multiple 1D fuzzy clusters in each input dimension. For each fuzzy cluster, a trapezoidal cluster is approximated. We refer the reader to [14] for a simple trapezoidal cluster approximation technique. The partition is converted to a Ruspini partition [15] for the convenience of the latter steps.

4. Each of the $n$ clusters $(C_{d1} - C_{dn})$ in the input dimension $d$, is a projection of the multi-dimensional input cluster to that input dimension. Next, the clusters from individual dimensions are combined to form the multi-dimensional input cluster. The merging process involves the use of a threshold $t$, which governs the degree of sparseness in the rule-base to be generated (more details on this later).

   The cluster in the multi-dimensional space is determined to be the region where the number of projected points in the region exceeds $t$. A point $p$ is contained in the cluster $C_i$ if $\mu_{Ci}(p) > \mu_{Cj}(p)$ for all $j \neq i$. The process has three steps:

   a. Find one of the multi-dimensional clusters $C$ where the number of points that falls into its projection exceeds the threshold $t$ using the following algorithm in Figure 1.
   b. Remove all data points that are contained in the cluster $C$ approximated.
   c. Repeat steps a – b until no more clusters can be found.

5. For each of the multi-dimensional clusters identified, a rule can be formed. For example, if a multi-dimensional cluster is formed with $[C_{11}, C_{23}, C_{34}]$ for the points projected from output cluster $B_i$, we obtain the following rule:
   *If* $x1$ *is* $C_{11}$ *and* $x_2$ *is* $C_{23}$ *and* $x_3$ *is* $C_{34}$ *then* $y$ *is* $B_i$

   Where $C_{dn}$ is the $n^{th}$ cluster identified at input dimension $d$.

6. The completed fuzzy rule-base then goes through a parameter identification process where each trapezoidal cluster in the input and output space is adjusted to improve the overall performance. The parameter identification is described in [14]. An alternative technique is proposed in [16].

```
PROCEDURE find_MD_cluster

Let U_i be the set of 1D clusters in dimension i
Let mdCluster = [ ]
for i = 1 to k
  for each unit u ∈ U_i
    utemp = mdCluster x u
    if utemp is dense
      denseunit = utemp
      break
    end if
  end for
end for
```

**Figure 1: Algorithm for finding multi-dimensional cluster**

The threshold $t$ in step 4 governs the sparseness of the fuzzy system to be generated. The higher the threshold, the fewer multi-dimensional clusters will be obtained and consequently fewer rules are generated. When $t$=0, all data involved in the rule extraction (training data) will be covered by at least 1 rule. We remark, however, that the parameter tuning process may create more gaps among rules and causes some of the training data to fall in the gaps.

The main aim of this paper is to validate the effectiveness of this technique when used with fuzzy rule interpolation to construct fuzzy systems with minimal number of rules without loosing significant accuracy.

## IV. Fuzzy Rule-Interpolation Based on Conservation of Relative Fuzziness

In this section, the rule interpolation based on conservation of fuzziness (RICF) is investigated. The technique considers only the flanks closer to the observation (see Figure 2) instead of the entire shapes of the rules. The main motivation is that often there are cases where the size of the fuzzy sets is not comparable with the observation, typically in a hierarchical fuzzy rule base [1, 3].

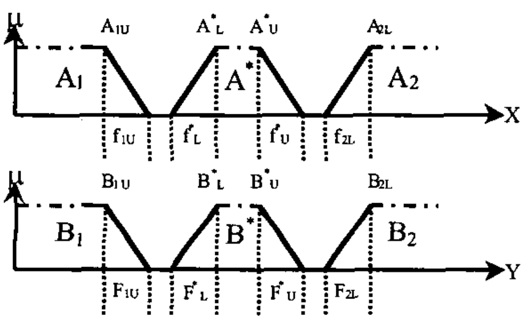


**Figure 2: Notation of input and output trapezoidal fuzzy sets involved in the fuzzy rule interpolation**

This technique can only be used for trapezoidal fuzzy sets (and its special case, triangular). From Figure 2, the $i^{th}$ input trapezoidal set has an upper and lower fuzziness denoted as $f_{iU}$ and $f_{iL}$. The fuzziness in the output space is denoted analogously by $F_{iU}$ and $F_{iL}$. The main idea of RICF is that given an observation $A^*$, the fuzziness of the output $B^*$ can be calculated based on the following ratio:

$$F_L^* : f_L^* = F_{1U} : f_{1U} \qquad \text{Eqn 15}$$

$$F_U^* : f_U^* = F_{2L} : f_{2L} \qquad \text{Eqn 16}$$

From (Eqn 15) and (Eqn 16), the upper and lower fuzziness of the output fuzzy set can be obtained:

$$F_L^* = f_L^* \frac{F_{1U}}{f_{1U}} \qquad \text{Eqn 17}$$

$$F_U^* = f_U^* \frac{F_{2L}}{f_{2L}} \qquad \text{Eqn 18}$$

To perform the interpolation, the core of $B^*$ ($B^*_L$ and $B^*_U$) is calculated using the FERI (Eqn 3):

$$B_L^* = \frac{\frac{1}{D_1(A_{1U}, A_L^*)} B_{1U} + \frac{1}{D_1(A_{2L}, A_L^*)} B_{2L}}{\frac{1}{D_1(A_{1U}, A_L^*)} + \frac{1}{D_1(A_{2L}, A_L^*)}} \qquad \text{Eqn 19}$$

$$B_U^* = \frac{\frac{1}{D_1(A_{1U}, A_U^*)} B_{1U} + \frac{1}{D_1(A_{2L}, A_U^*)} B_{2L}}{\frac{1}{D_1(A_{1U}, A_U^*)} + \frac{1}{D_1(A_{2L}, A_U^*)}} \qquad \text{Eqn 20}$$

$$D_1(A_i, A^*) = \begin{cases} A_L^* - A_{iU} & \text{if } A_i \prec A^* \\ A_{iL} - A_U^* & \text{if } A^* \prec A_i \end{cases} \qquad \text{Eqn 21}$$

It can be readily observed from (Eqn 21) that the distance ($D_1$) between neighboring fuzzy sets considers only the closest flanks from the two trapezoids.

## V. The Proposed Projection-Based Interpolation By Conservation of Fuzziness

In practical application, it is sometimes difficult to guarantee for each observation the existence of flanking rules that satisfy (Eqn 4) and (Eqn 5). It can therefore be argued that the multiple rules approach is more useful in practice. In this section, the use of multiple rules for RICF is investigated. We restrict our discussion for the singleton case and leave the generalization to non-singleton case for future research.

When the input $A^*$ is a singleton, the input fuzziness $f^*_L = f^*_U = 0$. It follows from (Eqn 17) and (Eqn 18) that $F^*_L = F^*_U = 0$, that is, the output will also be a singleton. Since in this case, $A^*_L = A^*_U$, it is simpler to leave out the subscript and use just $A^*$ for the rest of our discussion. The simplification is done analogously to the output $B^*$. In the one-dimensional case, by adapting (Eqn 8) and (Eqn 9) we obtain:

$$B^* = \frac{\sum_{i=1}^{N} \frac{1}{D_1(A_{ir}, A^*)} B_{ir}}{\sum_{i=1}^{N} \frac{1}{D_1(A_{ir}, A^*)}} \qquad \text{Eqn 22}$$

where $A_{ir}$ and $B_{ir}$ are chosen as $A_{iU}$ and $B_{iU}$ respectively if $A_i < A^*$; otherwise, $A_{iL}$ and $B_{iL}$ are used. The partial ordering between $A_i$ and $A^*$, which the selection of $A_{ir}$ and $B_{ir}$ depends on, is often ambiguous in the multi-dimensional case. Therefore, (Eqn 22) cannot be used for multi-dimensional data.

A simple way to alleviate this problem is to adopt a projection-based approach. That is, the interpolation is performed with the individual dimensions separately using (Eqn 22) and the results obtained are combined by averaging to form the final result. In this case, the term $1/D_1(A_{ir}, A^*)$ in (Eqn 22) can be zero, since it is possible for the multi-dimensional rules to overlap with the observation in some (but not all) dimensions. In such situations, we adopt the following:

$$\lim_{\frac{1}{D_1(A_p, A^*)} \to 0} B^* \to defuzz(B_i) \quad \text{Eqn 23}$$

Where defuzz(.) is a defuzzification method (e.g. Center of Area). If in any dimension *d*, there is more than one rule antecedent ($A_1 \ldots A_n$) that overlaps with the observation, then the average is used as the conclusion $I_d$ for that dimension:

$$I_d = \frac{1}{n}\sum_{i=1}^{n} defuzz(B_i) \quad \text{Eqn 24}$$

Finally, the system output is computed by averaging the conclusion from each of the *D* dimensions:

$$B^* = \frac{1}{D}\sum_{d=1}^{D} I_d \quad \text{Eqn 25}$$

## VI. Experiments

In this section, artificially generated as well as the box-jenkins [17] data have been used to validate the effectiveness of the projection-based fuzzy rule extraction technique used in conjunction with fuzzy rule interpolation. The mean square error is used as the performance index (*PI*) to evaluate the effectiveness of the fuzzy rule bases:

$$PI = \sum_{i=1}^{m} (y^i - \hat{y}^i)^2 / m \quad \text{Eqn 26}$$

where m is the number of data, $y^i$ is the $i^{th}$ actual output and $\hat{y}^i$ is the $i^{th}$ model output. The lower the performance index, the more accurate the fuzzy rule base.

### *A. Artificially Generated Data*

In the first experiment, the method was used to model a two-variable quadratic function:

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 0.8 \le x_1, x_2 \le 1.8 \quad \text{Eqn 27}$$

Two hundred input-output data pairs were generated from *y*. The output range was 3.15 - 13.72. The fuzzy rule extraction technique was applied multiple times to the data using different choices of the threshold *t*. Table 1 shows the thresholds used and the corresponding number of rules generated.

| Threshold *t* | Number of rules |
|---|---|
| 0.00 | 9 |
| 0.04 | 7 |
| 0.08 | 7 |
| 0.12 | 4 |
| 0.16 | 3 |
| 0.20 | 3 |

**Table 1: Number rules vs. threshold *t* used in the fuzzy rule extraction technique**

In Figure 3, we compare the errors of the interpolative system with the trivial average-based system that uses the average of the output domain as the default output whenever a observation cannot find rules to fire.

It can be observed that the error of the interpolative system is always lower. Picking the threshold $t = 0.12$, four fuzzy rules were generated and the PI of the system was 3.17. After 20 iterations of parameter tuning, the error decreased to 0.17, which can be considered reasonable.

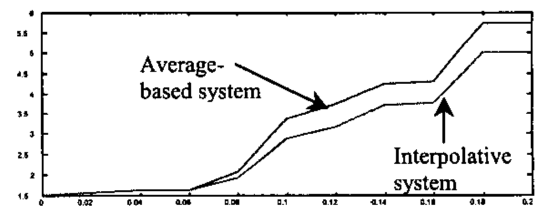


**Figure 3: Error comparison of the average and the interpolated results for the quadratic function**

In the inference process, ten data points cannot find rules to fire. Three out of the ten data points do not have a suitable pair of flanking rules. Table 2 compares the interpolated values from the projection-based n-rule interpolation and the traditional 2-rule interpolation. The symbol 'x' indicates data points that do not have flanking rules. It can be seen that even for data that has flanking rules, the projection-based approach out-performs the 2-rule approach in the majority number of points.

| Data Point | Projection | Two Rules | Desired |
|---|---|---|---|
| 1 | 9.0243 | x | 9.3272 |
| 2 | 8.9436 | x | 9.0339 |
| 3 | 7.7117 | 7.0010 | 8.0541 |
| 4 | 9.1674 | 7.9606 | 8.1745 |
| 5 | 9.3621 | x | 9.3280 |
| 6 | 10.2831 | 9.6010 | 9.3471 |
| 7 | 7.1934 | 6.6837 | 8.0829 |
| 8 | 9.2453 | 8.0331 | 8.1411 |
| 9 | 7.1519 | 6.5074 | 7.5747 |
| 10 | 6.8722 | 6.0185 | 7.1309 |

**Table 2: Comparison between projection-based n-rule interpolation and the traditional 2-rule interpolation.**

### *B. Box-Jenkins Data*

The data given by Box and Jenkins [17], often used as a benchmark in research papers, is used in this experiment. There are 290 data points, where the input variable *u* is the methane concentration and the output variable *y* is the $CO_2$ concentration. The input $y(t)$ and $u(t)$ are the output and input at time step *t* respectively. Figure 4 shows the error comparison in a way similar to Figure 3.

Table 3 summarizes the performance of the proposed technique as well as other models that use the data in a similar way as benchmark. It can be observed that the rules extraction technique is able to produce very few rules to achieve a reasonably high accuracy. The choices of the threshold $t$ provide the user with some amount of control over the number of rules generated.

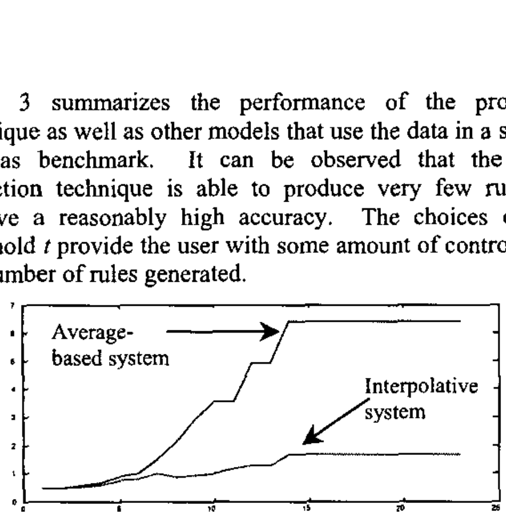


**Figure 4: Error comparison of the average and the interpolated results for the Box-Jenkins Data**

| Method | M | PI |
|---|---|---|
| Tong [18] | 19 | 0.469 |
| **Proposed ($t$ = 0.26)** | **5** | **0.435** |
| Xu et al [19] | 25 | 0.328 |
| Pedryc [20] | 81 | 0.320 |
| **Proposed ($t$ = 0.20)** | **9** | **0.250** |
| **Proposed ($t$ = 0.10)** | **10** | **0.139** |
| **Proposed ($t$ = 0.06)** | **15** | **0.129** |
| **Proposed ($t$ = 0.00)** | **17** | **0.121** |

**Table 3 Number of rules, M, and performance index, PI, of models using $y(t-1)$ and $u(t-4)$ as input variables**

## VII. Conclusion

In this paper, we have investigated the use of the projection-based fuzzy rule interpolation in conjunction with a sparse fuzzy system generation technique. By setting a parameter value, the user is able to generate fuzzy systems with a desirably small number of rules. The rule interpolation approach enables the system to maintain a reasonable accuracy. It has been shown in the experiments that the use of rule interpolation improves the system performance.

## References


[1] L. T. Koczy and K. Hirota, "Approximative inference in hierarchical structured rule bases," presented at Fift IFSA World Congress, Seoul, 1993.

[2] L. T. Koczy and K. Hirota, "Interpolation in fuzzy rule bases," Tokyo Institute of Technology, Yokohama TR 93-94/401, 1993.

[3] L. T. Koczy, K. Hirota, and T. D. Gedeon, "Fuzzy rule interpolation by the conservation of relative fuzziness," Hirota Lab, Nagatsuta 97-2, 1997.

[4] S. Kovács, "New Aspects of Interpolative Reasoning," presented at Proceedings of the 6th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Spain, 1996.

[5] D. Tikk and P. Baranyi, "Comprehensive analysis of a new fuzzy rule interpolation method," *IEEE Transactions on Fuzzy Systems*, vol. 8, pp. 281-296, 2000.

[6] T. D. Gedeon and L. T. Koczy, "Conservation of fuzziness in rule interpolation," presented at International Symposium on New Trends in Control of Large Scale Systems, Herlany, 1996.

[7] W. Z. Qiao, M. Masaharu, and Y. Shi, "An improvement to Koczy and Hirota's interpolation reasoning in sparse fuzzy rule bases," *International journal of Approximate Reasoning*, vol. 15, pp. 185-201, 1996.

[8] D. Dubois and H. Prade, "On fuzzy interpolation," presented at Proceedings of the 3rd International Conference on Fuzzy Logic & Neural Networks, Japan, 1994.

[9] S. Jenei, "A new approach for interpolation and extrapolation of compact fuzzy quantities," presented at Proceedings of the 21th Linz Sem. on Fuzzy Sets theory, 2000.

[10] A. Chong, T. D. Gedeon, and L. T. Koczy, "Projection Based Method for Sparse Fuzzy System Generation," presented at 2nd WSEAS Int. Conf. on Scientific Computation and Soft Computing, Crete, 2002.

[11] J. C. Bezdek, *Pattern Reconition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.

[12] Y. Fukuyama and M. Sugeno, "A new method of choosing the number of clusters for fuzzy c-means method," presented at Proceedings of the 5th Fuzzy Systems Symposium, 1989.

[13] A. Chong, T. D. Gedeon, and L. T. Koczy, "A Hybrid Approach for Solving the Cluster Validity Problem," presented at submitted to 14th International Conference on Digital Signal Processing, Greece, 2002.

[14] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Transactions on Fuzzy Systems*, vol. 1, pp. 7-31, 1993.

[15] E. H. Ruspini, "A new approach to clustering," *Information and Control*, vol. 15, pp. 22-32, 1969.

[16] D. Tikk, T. D. Gedeon, L. T. Koczy, and G. Biro, "Improvements and critique on Sugeno and Yasukawa's qualitative modelling," *IEEE Transactions on Fuzzy Systems*, 2002.

[17] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco: Holden Day, 1970.

[18] R. M. Tong, "The evaluation of fuzzy models derived from experimental data," *Fuzzy Sets and Systems*, vol. 4, pp. 1-12, 1980.

[19] C. W. Xu and Y. Z. Lu, "Fuzzy model identification and self-learning for dynamic systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, pp. 683-689, 1987.

[20] W. Pedrycz, "An identification algorithm in fuzzy relational systems," *Fuzzy Sets and Systems*, vol. 13, pp. 153-167, 1984.