Fuzzy Interpolation and Extrapolation: A Practical Approach

Huang, Zhiheng; Shen, Qiang

Published in:
IEEE Transactions on Fuzzy Systems

DOI:
10.1109/TFUZZ.2007.902038

Publication date:
2008

Citation for published version (APA):
Huang, Z., & Shen, Q. (2008). Fuzzy Interpolation and Extrapolation: A Practical Approach. *IEEE Transactions on Fuzzy Systems*, *16*(1), 13-28. https://doi.org/10.1109/TFUZZ.2007.902038

Fuzzy Interpolation and Extrapolation: A Practical Approach

Zhiheng Huang and Qiang Shen

Abstract—**Fuzzy interpolation does not only help to reduce the complexity of fuzzy models, but also makes inference in sparse rule-based systems possible. It has been successfully applied to systems control, but limited work exists for its applications to tasks like prediction and classification. Almost all fuzzy interpolation techniques in the literature make strong assumptions that there are two closest adjacent rules available to the observation, and that such rules must flank the observation for each attribute. Also, some interpolation approaches cannot handle fuzzy sets whose membership functions involve vertical slopes. To avoid such limitations and develop a more practical approach, this paper extends the work of Huang and Shen. The result enables both interpolation and extrapolation which involve multiple fuzzy rules, with each rule consisting of multiple antecedents. Two realistic applications, namely truck backer-upper control and computer activity prediction, are provided in this paper to demonstrate the utility of the extended approach. Experiment-based comparisons to the most commonly used Mamdani fuzzy reasoning mechanism, and to other existing fuzzy interpolation techniques are given to show the significance and potential of this research.**

Index Terms—**Fuzzy model simplification, fuzzy rule extrapolation, fuzzy rule interpolation, scale and move transformations, sparse rule base, transformation-based interpolation.**

I. Introduction

FUZZY rule interpolation helps reduce the complexity of fuzzy models and supports inference in systems that employ sparse rule sets [2], [3]. As argued in [1], with interpolation, fuzzy rules which may be approximated from their neighboring rules can be omitted from the rule base. This leads to the complexity reduction of fuzzy models. When given observations have no overlap with the antecedent values of the rules, classical fuzzy inference methods have no rule to fire, but interpolation methods can still obtain certain conclusions. Despite these significant advantages, earlier work in fuzzy interpolative reasoning does not guarantee the convexity of the derived fuzzy sets [4], [5], which is often a crucial requirement of fuzzy reasoning to attain more easily interpretable practical results.

Initially, the motivation of research on fuzzy interpolation is to eliminate the nonconvexity drawback. It soon goes beyond that to make fuzzy interpolation a more practical and general inference for both sparse and nonsparse rule bases. There has been considerable research reported in the literature in the last decade. For instance, Vas *et al.* have proposed an algorithm [6] that reduces the problem of nonconvex conclusions. Qiao *et al.* [7] have published an improved method which uses similarity-based reasoning to ensure the attainment of convex results. Hsiao *et al.* [8] have introduced a new interpolative method which exploits the slopes of the fuzzy sets. Baranyi*et al.* [9], [10] have proposed the work on general fuzzy interpolation and extrapolation techniques. Tikk *et al.* [11]–[13] have presented a modified α-cut-based method. Dubois *et al.* [14] have proposed a fuzzy interpolation technique using fuzzy relation in the Cartesian product of input and output space. Bouchon*et al.* [15] have created an interpolative method by exploiting the concept of graduality. Yam and Kóczy [16], [17] have proposed a fuzzy interpolative technique based on Cartesian representation, and Jenei *et al.* [18], [19] have introduced an axiomatic approach for fuzzy interpolation and extrapolation.

Nevertheless, some of the existing methods (e.g., [17]) involve complex computation. It becomes more difficult when they are extended to interpolation with multiple antecedents. Others (e.g., [8]) may only apply to simple fuzzy membership functions limited to triangular or trapezoidal. Apart from the work that uses different combinations of interpolation and inference schemes (e.g., [10]) and the approaches of [16] and [17] (that are able to generate multiple results but do not show how to make a choice amongst them), many existing techniques lack the flexibility to generate fuzzy results that meet different application requirements. In addition, practically, fuzzy sets used in a rule base may have vertical slopes. In fact, handling fuzzy sets with vertical slopes is crucial in prediction or classification problems as shown in Section V. However, some existing fuzzy interpolation methods cannot handle such cases.

The work of [1] proposes a novel interpolation method which avoids the problems mentioned above. It only considers interpolation between two adjacent rules. This paper further extends this method to deal with extrapolation as well as interpolation and which can both involve multiple rules, with each rule consisting of multiple antecedents. Although fuzzy interpolation techniques have been successfully applied to control problems [20]–[23], little has been reported for their use in performing tasks like prediction and classification. This extension helps bridge the gap between theory and prediction or classification applications, which may often require reasoning with multiple multiantecedent rules and extrapolation.

Incidentally, it is worth mentioning that a set of axioms have been proposed by Jenei [18], from a logical point of view, for fuzzy interpolation which include properties such as validity and compatibility amongst others. To uphold these properties,

Manuscript received April 10, 2006; revised February 8, 2007.

Z. Huang is with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720-1776 USA (e-mail: zhiheng@eecs.berkeley.edu).

Q. Shen is with the Department of Computer Science, University of Wales, Aberystwyth SY23 3DB, U.K. (e-mail: qqs@aber.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2007.902038

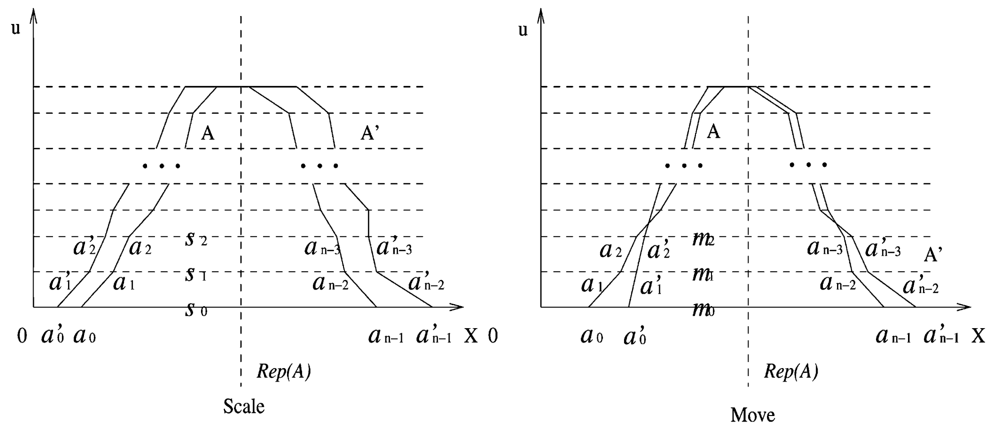


Fig. 1. Scale and move transformations.

a number of conditions are set. However, a particular constraint imposed is not realistic in practice, which requires that all intermediate fuzzy sets have a wider nuclear (interval of elements which have a full membership value) in the consequent part than those in the antecedent part. Violation of this constraint will cause the resultant fuzzy sets to lose normality (with the maximal fuzzy membership values of the result being less than 1). In addition, Jenei's method cannot handle the case where the membership functions of the intermediate fuzzy sets have vertical slopes. The interpolation method proposed here avoid these, and, hence, it does not necessarily require or satisfy properties such as the monotonicity as imposed in [18].

The rest of the paper is organized as follows. For completeness, Section II introduces the concept of general representative values for arbitrarily polygonal fuzzy sets, and Section III reviews the fuzzy interpolation method proposed in [1]. Section IV extends the method to handle the interpolation and extrapolation involving multiple rules, with each rule consisting of multiple antecedents. Section V gives two examples to illustrate the utility of this method, showing its potential in both model simplification and in performing interpolation and extrapolation inferences. Finally, Section VI concludes the paper.

II. General Representative Value

To facilitate the description of the proposed work, the concept of *representative value* (Rep) of a polygonal fuzzy set must be defined first. This value should capture important information such as the overall location of a fuzzy set, and will be used as the guide to perform scale and move transformations of given fuzzy sets. Consider an arbitrary polygonal fuzzy set with n odd points, $A = (a_0, \ldots, a_{n-1})$, as shown in Fig. 1. It has $\lfloor (n/2) \rfloor$ *supports* (horizontal intervals between every pair of odd points which have the same membership value) and $2(\lceil (n/2) \rceil - 1)$ *slopes* (nonhorizontal intervals between every pair of consecutive odd points). Note that two top points (of full membership value) do not have to be different. Although this figure explicitly assumes that evenly paired odd points are given at each α-cut level, this does not affect the generality of the fuzzy set representation as artificial odd points can be created to construct evenly paired odd points. Given such an arbitrary polygonal fuzzy set, its general Rep is defined by

$$\mathrm{Rep}(A) = \sum_{i=0}^{n-1} w_i a_i \tag{1}$$

where w_i is the weight assigned to point a_i.

Specifying the weights is necessary for a given application. The simplest case, which is called the *average Rep* hereafter, is that all points take the same weight value, i.e., $w_i = 1/n$.

An alternative definition named the *weighted average Rep* assumes that the weights increase upwards from the bottom support to the top support, to reflect the relative significance of the fuzzy membership values. For instance, assuming the weights increase upwards from $1/2$ to 1, such a Rep is defined by

$$\mathrm{Rep}(A) = \frac{\sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} \frac{1+\alpha_i}{2}(a_i + a_{n-1-i})}{\sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} \frac{1+\alpha_i}{2}} \tag{2}$$

where α_i is the membership value of a_i.

One of the most widely used defuzzification methods—the center of core can also be used to define the *center of core Rep*. In this case, the Rep is solely determined by those points with a fuzzy membership value of 1

$$\mathrm{Rep}(A) = \frac{1}{2}\left(a_{\lceil \frac{n}{2} \rceil - 1} + a_{n-\lceil \frac{n}{2} \rceil}\right). \tag{3}$$

Note that the general Rep definition can be simplified if the lengths of the $\lfloor (n/2) \rfloor$ supports $S_0, \ldots, S_{\lfloor (n/2) \rfloor - 1}$ (with the indices arranged in ascending order from the bottom to the top) are known. Indeed, as $a_{n-1-i} = a_i + S_i, i = 0, \ldots, \lfloor (n/2) \rfloor - 1$, the general form of (1) can be re-written as

$$\mathrm{Rep}(A) = \sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} a_i(w_i + w_{n-1-i}) + C \tag{4}$$

where $C = S_0 w_{n-1} + \cdots + S_{\lceil (n/2) \rceil - 1} w_{n-\lceil (n/2) \rceil}$ is a constant. Finally, it is worth indicating that the underlying scheme used to capture the representative value of a fuzzy set is the same as with the work of [12], that is, using a vector of characteristic points to represent a fuzzy set.

III. Overview of Interpolation

The method of [1] works by first constructing a new inference rule via manipulating two given adjacent rules, and then by using scale and move transformations to convert the intermediate inference results into the final derived conclusions. To be self-contained, a brief overview of this method is provided here. Note that all fuzzy sets involved are convex and normal polygonal fuzzy sets throughout this paper.

A. Construct the Intermediate Rule

To be concise, the simplest case is herein used to illustrate the underlying techniques for fuzzy interpolation. Given two adjacent rules (informally, where a rule does not exist whose antecedent value is between the antecedent values of these two rules) as follows:

$$\text{If } X \text{ is } A_1 \text{ then } Y \text{ is } B_1$$
$$\text{If } X \text{ is } A_2 \text{ then } Y \text{ is } B_2$$

where $A_i = (a_{i0}, \ldots, a_{i,n-1})$, $B_i = (b_{i0}, \ldots, b_{i,n-1})$, $i = 1, 2$, together with an observation $A^* = (a_0, \ldots, a_{n-1})$ which is located between fuzzy sets A_1 and A_2 (i.e., $\text{Rep}(A_1) < \text{Rep}(A^*) < \text{Rep}(A_2)$ if $\text{Rep}(A_1) < \text{Rep}(A_2)$, or $\text{Rep}(A_1) > \text{Rep}(A^*) > \text{Rep}(A_2)$ if $\text{Rep}(A_1) > \text{Rep}(A_2)$, this constraint will be disregarded in the extension, see Section IV), an interpolation is performed to achieve the fuzzy result B^*.

The transformation-based interpolation begins with constructing a new fuzzy set A' which has the same Rep as that of A^*. To support this, the distance between A_1 and A_2 is defined by

$$d(A_1, A_2) = d(\text{Rep}(A_1), \text{Rep}(A_2)) \tag{5}$$

where the actual scheme adopted to compute Reps is fixed for both A_1 and A_2, of course. A ratio λ_{Rep} $(0 \le \lambda_{\text{Rep}} \le 1)$ is introduced to represent the important impact of A_2 upon the construction of A' with respect to A_1

$$\lambda_{\text{Rep}} = \frac{d(A_1, A^*)}{d(A_1, A_2)}. \tag{6}$$

That is to say, if $\lambda_{\text{Rep}} = 0$, A_2 plays no part in constructing A', while if $\lambda_{\text{Rep}} = 1$, A_2 plays a full role in determining A'. Then,by using the simplest linear interpolation, the a'_i, $i = 0, \ldots, n-1$, of A' are calculated as follows:

$$a'_i = (1 - \lambda_{\text{Rep}})a_{1i} + \lambda_{\text{Rep}} a_{2i}. \tag{7}$$

Note that the resulting A' has the same representative value as A^* and that A' is convex and normal. Similarly, the consequent fuzzy set B' can be obtained by B_1, B_2 and λ_{Rep}. In doing so, a new rule $A' \Rightarrow B'$ is derived, which involves the use of only normal and convex fuzzy sets.

As $A' \Rightarrow B'$ is derived from $A_1 \Rightarrow B_1$ and $A_2 \Rightarrow B_2$, when A^* is given it is feasible to perform fuzzy reasoning with this new rule without further reference to its originals. Consider two extreme cases first. If $A^* = A_1$, then from (6) $\lambda_{\text{Rep}} = 0$, and according to (7), $A' = A_1$, and similarly $B' = B_1$, so the conclusion $B^* = B_1$. Likewise, if $A^* = A_2$, then $B^* = B_2$. Other than the extreme cases, *similarity* measures are used to support the interpolation that follows the intuition:

$$\text{The more similar } X \text{ to } A', \text{ the more similar } Y \text{ to } B'. \tag{8}$$

Suppose that a certain degree of similarity between A' and A^* is established, it is intuitive to require that the consequent parts B' and B^* attain the same similarity degree. The question is now how to obtain an operator which can capture the similarity degree between A' and A^*, and to allow transforming B' to B^* with the desired degree of similarity. To this end, the following two component transformations are proposed.

B. Scale Transformation

Consider applying scale transformation to an arbitrary polygonal fuzzy membership function $A = (a_0, \ldots, a_{n-1})$ (as shown on the left of Fig. 1) to generate $A' = (a'_0, \ldots, a'_{n-1})$ such that A and A' will have the same Rep, and $a'_{n-1-i} - a'_i = s_i(a_{n-1-i} - a_i)$, where s_i are scale rates and $i = 0, \ldots, \lfloor (n/2) \rfloor - 1$. In order to achieve this, $\lfloor (n/2) \rfloor$ equations $a'_{n-1-i} - a'_i = s_i(a_{n-1-i} - a_i)$, $i = 0, \ldots, \lfloor (n/2) \rfloor - 1$, are imposed to obtain the supports with desired lengths, and $(\lceil (n/2) \rceil - 1)$ equations $((a'_{i+1} - a'_i)/(a'_{n-1-i} - a'_{n-2-i})) = ((a_{i+1} - a_i)/(a_{n-1-i} - a_{n-2-i}))$, $i = 0, \ldots, \lceil (n/2) \rceil - 2$ are imposed to equalize the ratios between the left $(\lceil (n/2) \rceil - 1)$ slopes' lengths and the right $(\lceil (n/2) \rceil - 1)$ slopes' lengths of A' to the ratio counterparts of the original fuzzy set A. The equation $\sum_{i=0}^{n-1} w_i a'_i = \sum_{i=0}^{n-1} w_i a_i$ which ensures the Reps to remain the same before and after the transformation is added to make up of $\lfloor (n/2) \rfloor + (\lceil (n/2) \rceil - 1) + 1 = n$ equations. For clarity, these n equations are collectively written as

$$\begin{cases} a'_{n-1-i} - a'_i = s_i(a_{n-1-i} - a_i) = S_i \\ \left(i = 0, \ldots, \left\lfloor \frac{n}{2} \right\rfloor - 1\right) \\ \frac{a'_{i+1} - a'_i}{a'_{n-1-i} - a'_{n-2-i}} = \frac{a_{i+1} - a_i}{a_{n-1-i} - a_{n-2-i}} = R_i \\ \left(i = 0, \ldots, \left\lceil \frac{n}{2} \right\rceil - 2\right) \\ \sum_{i=0}^{n-1} w_i a'_i = \sum_{i=0}^{n-1} w_i a_i \end{cases} \tag{9}$$

where S_i is the ith support length of the resultant fuzzy set and R_i is the ratio between the ith left slope length and the ith right slope length. Solving these n equations simultaneously results in a unique and convex fuzzy set A'. It can be shown [1] that given a fuzzy set A and the support scale rates s_i, the use of a different Rep will not affect the geometrical shape of the resultant fuzzy set. Instead, it only affects the position of the transformed fuzzy set.

However, arbitrarily choosing the ith support scale rate when the $(i-1)$th scale rate is fixed may lead the ith support to becoming wider than the $(i-1)$th support, i.e., $S_i > S_{i-1}$. To avoid this, the ith *scale ratio* $\mathbb{S}_i$, which represents the actual increase of the ratios between the ith supports and the $(i-1)$th supports, before and after the transformation, normalized over the maximal of such an increase (in the sense that it does not lead to nonconvexity), is introduced to restrict s_i with respect to s_{i-1}

$$\mathbb{S}_i = \begin{cases} \dfrac{\frac{s_i(a_{n-i-1}-a_i)}{s_{i-1}(a_{n-i}-a_{i-1})} - \frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}}}{1-\frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}}}, & \text{if } s_i \geq s_{i-1} \geq 0 \\ \dfrac{\frac{s_i(a_{n-i-1}-a_i)}{s_{i-1}(a_{n-i}-a_{i-1})} - \frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}}}{\frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}}}, & \text{if } s_{i-1} \geq s_i \geq 0. \end{cases} \tag{10}$$

If $\mathbb{S}_i \in [0,1]$ (when $s_i \geq s_{i-1} \geq 0$) or $\mathbb{S}_i \in [-1,0]$ (when $s_{i-1} \geq s_i \geq 0$), then $S_{i-1} \geq S_i$. In summary, if given s_i ($i = 0, \ldots, \lfloor (n/2) \rfloor - 1$) such that $\mathbb{S}_i \in [0,1]$ or $\mathbb{S}_i \in [-1,0]$ (depending on whether $s_i \geq s_{i-1}$ or not), $i = 1, \ldots, \lfloor (n/2) \rfloor - 1$, the scale transformation guarantees to produce a normal and convex fuzzy set.

Conversely, if two convex sets $A = (a_0, \ldots, a_{n-1})$ and $A' = (a'_0, \ldots, a'_{n-1})$ which have the same Rep are given, the scale rate of the bottom support, s_0, and the scale ratio of the ith support, $\mathbb{S}_i$ ($\mathbb{S}_i, i = 1, \ldots, \lfloor (n/2) \rfloor - 1$) can be calculated by (11) and (12), shown at the bottom of the page. Since A and A' are both convex, $\mathbb{S}_i$ must be within the range as given in (12).

C. Move Transformation

Now, consider the move transformation (on the right of Fig. 1) applied to an arbitrary polygonal fuzzy membership function $A = (a_0, \ldots, a_{n-1})$ to generate $A' = (a'_0, \ldots, a'_{n-1})$, such that A and A' have the same Rep and the same lengths of supports, and $a'_i = a_i + l_i, i = 0, \ldots, \lceil (n/2) \rceil - 2$. In order to achieve this, the move transformation is decomposed into ($\lceil (n/2) \rceil - 1$) submoves. The ith submove ($i = 0, \ldots, \lceil (n/2) \rceil - 2$) moves the ith support (indexed from bottom to top beginning with 0) to a desired place. This operator moves all the odd points on and above the ith support, whilst unaltering those points under this support. To measure the degree of the ith submove, the first possible maximal move distance (in the sense that the corresponding submove does not lead to the part of the fuzzy set above the ith support becoming nonconvexity) should be worked out first. To simplify the description of the submove procedure, only the move on the right side (from a_i's point of view) is considered in the discussion hereafter. The left direction simply mirrors this operation.

If the ith point is supposed to move to the right direction, the maximal position $a_i^{(i)*}$ can be calculated as follows when: $\sum_{j=i}^{\lceil (n/2) \rceil - 1} (w_j + w_{n-1-j}) > 0$

$$a_i^{(i)*} = \frac{\sum_{j=i}^{\lceil \frac{n}{2} \rceil - 1} a_j (w_j + w_{n-1-j}) - T}{\sum_{j=i}^{\lceil \frac{n}{2} \rceil - 1} (w_j + w_{n-1-j})} \tag{13}$$

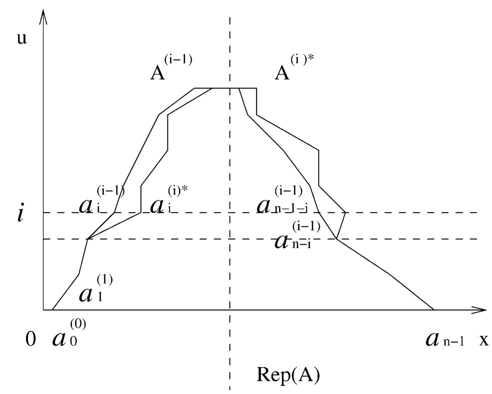


Fig. 2. Extreme move positions in the ith submove.

where $T = \sum_{\substack{w_k + w_{n-1-k} < 0 \\ i<k<\lceil (n/2) \rceil}} [(S_{k-1} - S_k) \sum_{m=k}^{\lceil (n/2) \rceil - 1} (w_m + w_{n-1-m})]$ and S_k is the length of the kth support (either before or after move transformation as they are the same). If, however, $\sum_{j=i}^{\lceil (n/2) \rceil - 1} (w_j + w_{n-1-j}) < 0$, the maximal position $a_i^{(i)*}$ is calculated similarly to (13) except that the condition $w_k + w_{n-1-k} < 0$ on term T is changed to $w_k + w_{n-1-k} > 0$. It can be shown that the other extreme moving points $a_j^{(i)*}$ ($j = i+1, \ldots, \lceil (n/2) \rceil - 1$) which are on the left side of the fuzzy set in the ith submove can be computed by

$$a_j^{(i)*} = \begin{cases} a_{j-1}^{(i)*}, & \text{if } w_j + w_{n-1-j} > 0 \\ a_{j-1}^{(i)*} + S_{j-1} - S_j, & \text{if } w_j + w_{n-1-j} < 0 \end{cases} \tag{14}$$

Also, it can be seen that all the extreme points determine a normal and convex fuzzy set $A^{(i)*}$ (as illustrated in Fig. 2) which must have at least a vertical slope between any two consecutive α-cuts above the ith support. This fuzzy set will have the same Rep as $A^{(i-1)}$ with respect to the move transformation. That is

$$\sum_{j=0}^{\lceil \frac{n}{2} \rceil - 1} a_j^{(i)*} (w_j + w_{n-1-j}) = \sum_{j=0}^{\lceil \frac{n}{2} \rceil - 1} a_j^{(i-1)} (w_j + w_{n-1-j}). \tag{15}$$

From (13), the first maximal move distance can be calculated. However, the ith submove should not only consider nonconvexity above the ith support, but also ensure the avoidance of nonconvexity below. Otherwise it may still lead to nonconvexity as illustrated in Fig. 2. For this, the second maximal move distance is calculated as $(a_{n-i}^{(i-1)} - a_{n-1-i}^{(i-1)})$. It is intuitive to select

$$s_0 = \frac{a'_{n-1} - a'_0}{a_{n-1} - a_0} \tag{11}$$

$$\mathbb{S}_i = \begin{cases} \dfrac{\frac{a'_{n-i-1}-a'_i}{a'_{n-i}-a'_{i-1}} - \frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}}}{1-\frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}}} \in [0,1], & \text{if } \frac{a'_{n-i-1}-a'_i}{a_{n-i-1}-a_i} \geq \frac{a'_{n-i}-a'_{i-1}}{a_{n-i}-a_{i-1}} \geq 0 \\ \dfrac{\frac{a'_{n-i-1}-a'_i}{a'_{n-i}-a'_{i-1}} - \frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}}}{\frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}}} \in [-1,0], & \text{if } \frac{a'_{n-i}-a'_{i-1}}{a_{n-i}-a_{i-1}} \geq \frac{a'_{n-i-1}-a'_i}{a_{n-i-1}-a_i} \geq 0. \end{cases} \tag{12}$$

the minimal of these two maximal move distances to act as the actual maximal move distance, in order to avoid nonconvexity. The move ratio $\mathbb{M}_i$, which is used to measure the degree of such a submove is, thus, calculated by (16), shown at the bottom of the page, where the notation $a_i^{(i-1)}$ represents a_i's new position after the $(i-1)$th submove. Initially, $a_i^{(-1)} = a_i$.

If $\mathbb{M}_i \in [0,1]$ when $l_i \geq (a_i^{(i-1)} - a_i)$, or $\mathbb{M}_i \in [-1,0]$ when $l_i \leq (a_i^{(i-1)} - a_i)$, the submove is carried out as follows. The odd points under the ith support are not changed: $a_j^{(i)} = a_j^{(i-1)} (j = 0,\ldots,i-1, n-i,\ldots,n-1)$ while the other points $a_i^{(i-1)}, a_{i+1}^{(i-1)},\ldots,a_{n-1-i}^{(i-1)}$ are to be moved. At the beginning, when $i = 0$, all odd points are moved of course. If moving to the right side from the viewpoint of $a_i^{(i-1)}$, i.e., $\mathbb{M}_i \in [0,1]$, the moving distances of $a_j^{(i-1)}$ $(j = i, i+1,\ldots,\lceil (n/2)\rceil - 1)$ which are on the left side of the fuzzy set $A^{(i-1)}$ are calculated by multiplying $\mathbb{M}'_i$ with the distances between the extreme positions $a_j^{(i)*}$ and themselves. In doing so, $a_j^{(i-1)}$ will move the same proportion of distance to their respective extreme positions. That is

$$a_j^{(i)} = a_j^{(i-1)} + \mathbb{M}'_i \left(a_j^{(i)*} - a_j^{(i-1)}\right) \tag{17}$$

where

$$\mathbb{M}'_i = \mathbb{M}_i \frac{\min\left\{a_i^{(i)*} - a_i^{(i-1)}, a_{n-i}^{(i-1)} - a_{n-1-i}^{(i-1)}\right\}}{a_i^{(i)*} - a_i^{(i-1)}}. \tag{18}$$

This represents the *applied move ratio* for the ith submove. If $\mathbb{M}_i \in [0,1]$, $\mathbb{M}'_i \in [0,\mathbb{M}_i]$. The adoption of applied move ratio $\mathbb{M}'_i$ avoids the potential nonconvexity below. Such a move strategy leads to a fuzzy set $A^{(i)} = (a_0^{(i)},\ldots,a_{n-1}^{(i)})$ which is convex, has the same Rep as A, and its ith point is located at the new, desired position $a_i^{(i)}$, i.e., $a_{j+1}^{(i)} - a_j^{(i)} \geq 0$ $(j = 0,\ldots,n-2)$, $\mathrm{Rep}(A^{(i)}) = \mathrm{Rep}(A)$, and $a_i^{(i)} = a_i + l_i$.

In summary, if given move ratios $\mathbb{M}_i \in [-1,1]$, $(i = 0,\ldots,\lceil (n/2)\rceil - 2)$, the $(\lceil (n/2)\rceil - 1)$ submoves transform a given normal and convex set $A = (a_0,\ldots,a_{n-1})$ to a new normal and convex set $A' = (a'_0,\ldots,a'_{n-1})$ with the same lengths of supports and the same Rep.

In the converse case, where two convex fuzzy sets $A = (a_0,\ldots,a_{n-1})$ and $A' = (a'_0,\ldots,a'_{n-1})$ of the same representative value are given, the move ratio as $\mathbb{M}_i$, $i = 0,1,\ldots,\lceil (n/2)\rceil - 2$, are computed by

$$\mathbb{M}_i = \begin{cases} \dfrac{a'_i - a_i^{(i-1)}}{\min\left\{a_i^{(i)*} - a_i^{(i-1)}, a_{n-i}^{(i-1)} - a_{n-1-i}^{(i-1)}\right\}}, & \text{if } a'_i \geq a_i^{(i-1)} \\ \dfrac{a'_i - a_i^{(i-1)}}{\min\left\{a_i^{(i-1)} - a_i^{(i)*}, a_i^{(i-1)} - a_{i-1}^{(i-1)}\right\}}, & \text{if } a'_i \leq a_i^{(i-1)} \end{cases} \tag{19}$$

where $a_i^{(i-1)}$ is the a_i's new position after the $(i-1)$th submove. Initially, when $i = 0$, $a_i^{(-1)} = a_i$. This (bottom) submove will not lead to any nonconvexity below as there are no odd points underneath, whilst the other submoves need to consider situations where nonconvexity may arise both above and underneath. When $i = 0$, $a_{n-i}^{(i-1)} - a_{n-1-i}^{(i-1)}$ and $a_i^{(i-1)} - a_{i-1}^{(i-1)}$ are not defined. In order to keep the expression the same for (19), both of them take an infinite value. That is, the denominators in (19) are simplified to $(a_i^{(i)*} - a_i^{(i-1)})$ and $(a_i^{(i-1)} - a_i^{(i)*})$.

Since $A = (a_0,\ldots,a_{n-1})$ and $A' = (a'_0,\ldots,a'_{n-1})$ are both convex, it is obvious that $\mathbb{M}_i \in [0,1]$ when $a'_i \geq a_i^{(i-1)}$ and $\mathbb{M}_i \in [-1,0]$ when $a'_i \leq a_i^{(i-1)}$, $i = 0,1,\ldots,\lceil (n/2)\rceil - 2$.

D. Algorithm Outline

In summary, scale and move transformations transfer a fuzzy set A to another A^* which has the same representative value as A. Scale transformation scales A up or down to A' retaining the ratio between left and right slope, but having a different support length. The closer the scale ratio to 0, the more similar A and A'. Move transformation shifts A' to A^* which has the same support length as A', but has a different location for support. The closer the move ratio to 0, the more similar A' and A^*. Both scale and move transformations guarantee the representative value unchanged, and they both guarantee that the transferred fuzzy sets have the same type of shape as the original one. For example, the transformation of a hexagonal fuzzy set will lead to another hexagonal fuzzy set.

As indicated earlier, it is intuitive to maintain the similarity degree between the consequent parts $B' = (b'_0,\ldots,b'_{n-1})$ and $B^* = (b^*_0,\ldots,b^*_{n-1})$ to be the same as that between the antecedent parts $A' = (a'_0,\ldots,a'_{n-1})$ and $A^* = (a^*_0,\ldots,a^*_{n-1})$, in performing interpolative reasoning. As scale and move transformations allow the similarity degree between two fuzzy sets to be measured by the *scale rate, scale ratios*, and *move ratios*, the desired conclusion B^* can be computed from A' and *scale rate*, *scale ratios*, and *move ratios* calculated from A' to A^* (as illustrated in Fig. 3 for an interpolation involving triangular fuzzy sets). The computation procedure is summarized as follows.

1) Calculate scale rates s_i $(i = 0,1\ldots,\lfloor (n/2)\rfloor - 1)$ of the ith support from A' to A^* by $s_i = ((a^*_{n-1-i} - a^*_i)/(a'_{n-1-i} - a'_i))$.
2) Calculate scale rate s_0 of the bottom support (or just get from the first step) and scale ratios $\mathbb{S}_i$ $(i = 1\ldots,\lfloor (n/2)\rfloor - 1)$ of the ith support from A' to A^* by (11) and (12).
3) Apply scale transformation to A' with scale rates s_i as calculated in the first step to obtain A''.
4) Assign scale rate s'_0 of the bottom support of B' to the value of s_0 (i.e., $s'_0 = s_0$), with the scale ratios $\mathbb{S}'_i$, $(i = 1\ldots,\lfloor (n/2)\rfloor - 1)$ of the ith support of B' calculated as

$$\mathbb{M}_i = \begin{cases} \dfrac{l_i - \left(a_i^{(i-1)} - a_i\right)}{\min\left\{a_i^{(i)*} - a_i^{(i-1)}, a_{n-i}^{(i-1)} - a_{n-1-i}^{(i-1)}\right\}}, & \text{if } l_i \geq \left(a_i^{(i-1)} - a_i\right) \\ \dfrac{l_i - \left(a_i^{(i-1)} - a_i\right)}{\min\left\{a_i^{(i-1)} - a_i^{(i)*}, a_i^{(i-1)} - a_{i-1}^{(i-1)}\right\}}, & \text{if } l_i \leq \left(a_i^{(i-1)} - a_i\right) \end{cases} \tag{16}$$

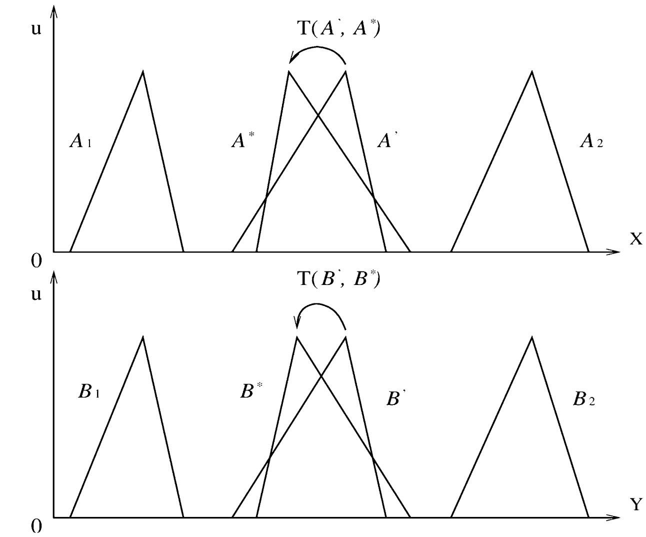


Fig. 3. Interpolative method.

TABLE I
EXAMPLE 1

Attribute Values	$A_1 = (0, 4, 5, 6)$, $A_2 = (11, 12, 13, 14)$ $B_1 = (0, 2, 3, 4)$, $B_2 = (10, 11, 12, 13)$
Observation	$A^* = (6, 6, 9, 10)$
Result	$B^* = (4.73, 4.73, 7.02, 7.70)$

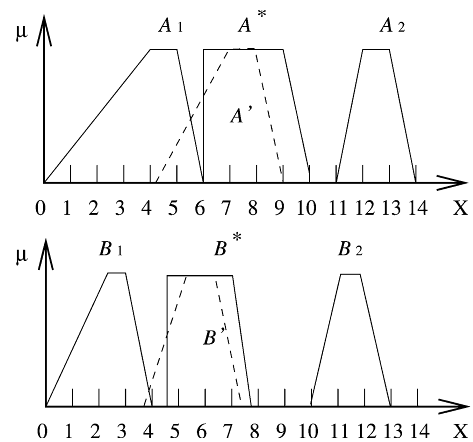


Fig. 4. Example 1: Normal interpolation.

per (12) under the condition that they equal to $\mathbb{S}_i$ ($i = 1 \ldots, \lfloor (n/2) \rfloor - 1$) as calculated in step 2) [see (20), shown at the bottom of the page].

5) Apply scale transformation to B' using s'_i ($i = 0, 1 \ldots, \lfloor (n/2) \rfloor - 1$) as calculated in step 4) to obtain $B'' = (b''_0, \ldots, b''_{n-1})$.
6) Decompose the move transformation to ($\lceil (n/2) \rceil - 1$) submoves. For $i = 0, 1, \ldots, \lceil (n/2) \rceil - 2$.
   a) Calculate the ith submove ratio $\mathbb{M}_i$ from $A^{(i-1)}$ to A^* by (19), where $A^{(i-1)}$ is the fuzzy set obtained after the $(i-1)$th submove with initialization $A^{(-1)} = A''$.
   b) Apply move transformation to $A^{(i-1)}$ using $\mathbb{M}_i$ to obtain $A^{(i)} = (a_0^{(i)}, a_1^{(i)}, \ldots, a_n^{(i)})$.
   c) Apply move transformation to $B^{(i-1)}$ using $\mathbb{M}_i$ to obtain $B^{(i)} = (b_0^{(i)}, b_1^{(i)}, \ldots, b_n^{(i)})$.
7) Return $A^{(\lceil (n/2) \rceil - 2)} = A^*$ and $B^{(\lceil (n/2) \rceil - 2)}$, which is the required resultant fuzzy set B^*, when the *for* loop of step 6) terminates.

Note that, with respect to n (the largest number of odd points for any fuzzy sets involved), the computational complexity of this transformation-based interpolation is $O(n^2)$, mainly owing to step 6) in Section III-D. This is acceptable given that n is not significantly large in most cases.

To explain the computation involved, an example is given as follows.

Example 1: Two rules $A_1 \Rightarrow B_1$, $A_2 \Rightarrow B_2$ and the observations A^* are given in order to determine the result B^*. This example concerns a trapezoidal interpolation. All the attributes and results with observation $A^*(6, 6, 9, 10)$ are shown in Table I and Fig. 4. Consider the use of center of core Rep in this example, $A'(4.12, 7.00, 8.00, 9.00)$ and $B'(3.75, 5.38, 6.38, 7.38)$ are calculated by interpolation of A_1, A_2, and B_1, B_2, respectively, with $\lambda = 0.38$. This is calculated using (6). Then, the interpolation via scale and move transformations is carried out according to the steps of the algorithm. 1) Calculating the bottom support scale rate (0.82) and top support scale rate (3.0) from A' to A^*. 2) Calculating the top support scale ratio (0.69) from A' to A^*. 3) Scaling A' to generate $A''(5.26, 6.00, 9.00, 9.26)$ using the bottom and top scale rates calculated in step 1). Note that A'' is a convex fuzzy set which has the same Rep and has the same bottom and top support lengths as A^*. 4) Computing the bottom and top support scale rates (0.82 and 2.30) over B' according to (20). 5) Scaling B' to generate $B''(4.31, 4.73, 7.02, 7.28)$ using the bottom and top scale rates calculated in step 4). 6) Calculating the move ratio from A'' to A^*. Its value is 1.0 as A^* has vertical left slope. This move ratio is used to move B'' to obtain the resultant fuzzy set $B^*(4.73, 4.73, 7.02, 7.70)$. In this example, the interpolation method resulted in a fuzzy set which still has a vertical left slope.

The above review is given for the simplified cases where both rules involved in the interpolation have one antecedent attribute. This is purely for easy illustration purposes. The work has been developed to cover more general situations where the adjacent rules involve more than one antecedent attribute, as detailed in [1].

$$s'_i = \begin{cases} s_i, & (i = 0) \\ \dfrac{s'_{i-1}(s_i - s_{i-1})\left(\frac{b'_{n-i} - b'_{i-1}}{b'_{n-i-1} - b'_i} - 1\right)}{s_{i-1}\left(\frac{a'_{n-i} - a'_{i-1}}{a'_{n-i-1} - a'_i} - 1\right)} + s'_{i-1}, & (s_i \geq s_{i-1} \geq 0) \\ \dfrac{s'_{i-1} s_i}{s_{i-1}}, & (s_{i-1} \geq s_i \geq 0) \end{cases} \quad (20)$$

IV. EXTENSIONS

All fuzzy interpolation techniques in the literature assume that two closest adjacent rules to the observation are available. Also, most interpolation methods presume that such rules must flank the observation for each attribute (but not necessarily in the same order). In practice, however, there may be a different number of the closest rules to a given observation, and the attribute values of these rules may lie just on one side of the observation. In addition, some interpolation methods cannot handle cases where fuzzy sets with vertical slopes are involved. These limitations inevitably restrict the potential application of the existing techniques. Although fuzzy interpolation has been applied to control problems [20]–[23], little has been done for tasks such as prediction and classification. To resolve this, the work of [1] is first extended herein to allow interpolations that involve multiple multiantecedent rules, without making the strong assumption that antecedent attributes flank the observation. Furthermore, it is shown that exploiting the generality of this extension, extrapolation can be performed over multiple multiantecedent rules in a straightforward manner. This further extension to extrapolation makes the work of [1] much more useful as demonstrated in Section IV-B and the follow-up sections.

A. Interpolation With Multiple Multiantecedent Rules

To allow fuzzy interpolation with more than two rules given a rule base, the first step is to choose n $(n \geq 2)$ closest rules from the rule base. The choice of a larger n will help approaching global consideration of neighboring rules in performing fuzzy interpolation, thereby resulting in smoother decision surfaces. On the contrary, the choice of a relatively smaller n will tend to consider only neighboring rules, whilst taking less computation time. The value of n may be predetermined by trading off between the smoothness of decision surfaces and quick response of decision making. This requires a consistent use for a given application domain, in order to ease interpretation of the interpolated results. In the experiments carried out in this paper, for simplicity of illustration, only two or three closest rules are chosen to perform fuzzy interpolations. Then, selected rules are used to construct the intermediate fuzzy rule. Once the intermediate rule is worked out, the rest of the process remains the same as described in Section III. The following shows these two important steps.

1) Choose the Closest n Rules: Without losing generality, suppose that a rule R_i and an observation are represented by

$$\text{Rule } R_i\text{: if } X_1 \text{ is } A_{1i} \text{ and} \ldots \text{and } X_m \text{ is } A_{mi} \text{ then } Y \text{ is } B_i \tag{21}$$

$$\text{Observation: } X_1 \text{ is } A_1^* \text{ and} \ldots \text{and } X_m \text{ is } A_m^*. \tag{22}$$

According to the distance definition (5) between two fuzzy terms, the distances d_k, $k = 1, \ldots, m$, between the pairs of A_{ki} and A_k^* can be calculated as

$$d_k = d(A_{ki}, A_k^*) = d(\text{Rep}(A_{ki}), \text{Rep}(A_k^*)). \tag{23}$$

As attributes may have different domains, the absolute distances may not be compatible with each other. To make these comparable, each distance measure is normalized into the range of 0 to 1

$$d'_k = \frac{d(A_{ki}, A_k^*)}{\max_k - \min_k} = \frac{d(\text{Rep}(A_{ki}), \text{Rep}(A_k^*))}{\max_k - \min_k} \tag{24}$$

where $\max_k$ and $\min_k$ are the maximal and minimal values of attribute k given. The distance d between a rule and an observation can be calculated as the average of all attributes' distances. The Euclidean version of the distance, which is to be used in the later implementation, is, therefore

$$d = \sqrt{{d'_1}^2 + {d'_2}^2 + \cdots + {d'_m}^2}. \tag{25}$$

If, however, the importance of attributes are not equal, weights may be used. Note that if a conditional part of a rule is missing, the distance of this attribute is treated as 0 to reflect that any data value is very close to the *null* attribute value. This allows for measuring the distances between a given observation and rules which may not have fuzzy sets associated with certain attributes.

Once the distance definition of (25) is given, the distances between a given observation and all rules in the rule base can be calculated. The n rules which have minimal distances are chosen as the closest n rules from the observation. It is worth noting that the n closest rules do not necessarily flank the observation. In the extreme case, all the chosen rules may lie on one side, resulting in extrapolation rather than interpolation (see Section IV-B).

2) Construct the Intermediate Rule: The main issue that remains is how to construct the intermediate rule after n $(n \geq 2)$ closest rules have been chosen. Let w_{ki}, $i = 1, \ldots, n$, $k = 1, \ldots, m$, denote the weight to which the kth term of the ith fuzzy rule contributes to constructing the kth intermediate fuzzy term A'_k. Intuitively, the larger the distance from A_{ki} to A_k^* is, the less value w_{ki} should take. In particular, the inversion of the distance can be used to act as the weight

$$w_{ki} = \frac{1}{d(A_{ki}, A_k^*)} \tag{26}$$

where $d(A_{ki}, A_k^*)$ is defined as per (23). Of course, if desired, alternative nonincreasing functions such as $w_{ki} = \exp^{-d(A_{ki}, A_k^*)}$ may be adopted to assign different weights.

For each attribute k, the weights w_{ki}, $i = 1, \ldots, n$, are used to compute the intermediate fuzzy term A_k^*. Prior to that, they are normalized as follows:

$$w'_{ki} = \frac{w_{ki}}{\sum_{t=1,\ldots,n} w_{kt}} \tag{27}$$

so that their sum over attribute k equals to 1. Therefore, the intermediate fuzzy term A''_k, $k = 1, \ldots, m$, are computed as

$$A''_k = \sum_{i=1,\ldots,n} w'_{ki} A_{ki} \tag{28}$$

which correctly degenerates to (7) when only two rules $(n = 2)$ are considered for interpolation. That is, the two-rule interpolation case is a special case of the generalized multirule interpolation.

In the two-rule interpolation case, the A''_k calculated via (28) has the same Rep as the input A_k^*. However, this is generally

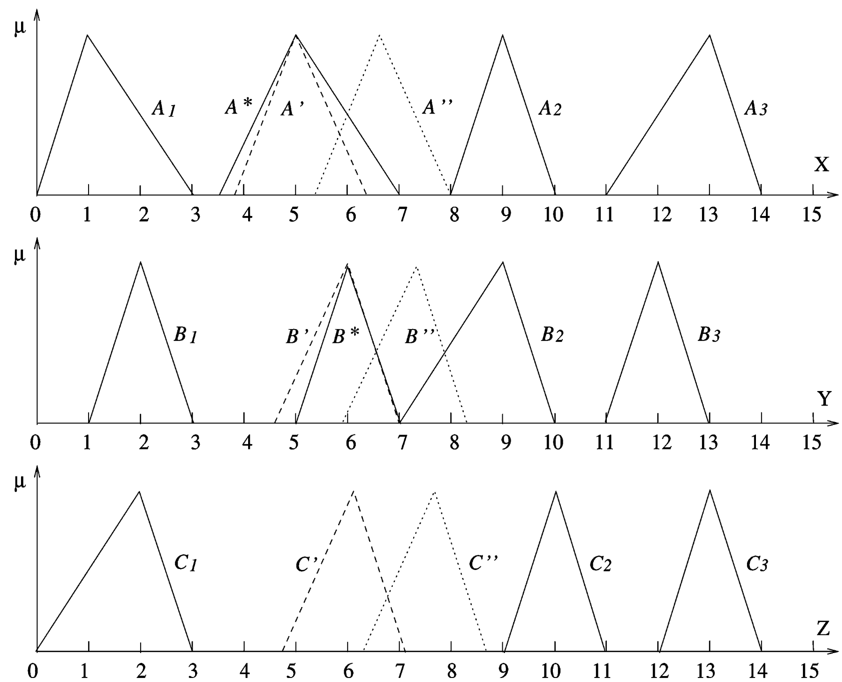


Fig. 5. Example 2: Interpolation with multiple rules.

not true when more than two rules are involved (that is why the symbol A''_k, rather than A'_k, is used here). Thus, it does not satisfy the requirement, of having the same Rep value before and after transformation, as imposed by the scale and move transformation-based interpolation. In order to solve this problem, the process of *shift* is suggested to modify A''_k so that it becomes a new intermediate fuzzy term A'_k which has the same Rep as A^*_k. In particular, the shift of A''_k, $k = 1, \ldots, m$, is performed as follows:

$$A'_k = A''_k + \delta_k(\max_k - \min_k) \quad (29)$$

where δ_k is a constant defined by

$$\delta_k = \frac{\text{Rep}(A^*_k) - \text{Rep}(A''_k)}{\max_k - \min_k}. \quad (30)$$

In doing so, the following holds:

$$\text{Rep}(A'_k) = \text{Rep}(A^*_k). \quad (31)$$

Regarding the consequent, by analogy to (28), the intermediate fuzzy output B'' can be computed by

$$B'' = \sum_{i=1,\ldots,n} w'_{ai} B_i \quad (32)$$

where w'_{ai} is the mean of w'_{ki}

$$w'_{ai} = \frac{1}{m} \sum_{k=1}^{m} w'_{ki}. \quad (33)$$

B'' is then shifted to B' as follows:

$$B' = B'' + \delta_a(\max - \min) \quad (34)$$

where $\max$ and $\min$ are maximal and minimal values of output attribute, and δ_a is the mean of the shift parameters δ_k, $k = 1, \ldots, m$

$$\delta_a = \frac{1}{m} \sum_{k=1}^{m} \delta_k. \quad (35)$$

From this, the intermediate fuzzy rule

$$\text{if } X_1 \text{ is } A'_1 \text{ and} \ldots \text{and } X_m \text{ is } A'_m \text{ then } Y \text{ is } B'_m$$

can be constructed via (29) and (34). The rest of the interpolation reasoning is, hence, applied to this intermediate rule and the observation, in the same way as presented in Section III. An example follows to explain how this works.

Example 2: Three rules $A_i \wedge B_i \Rightarrow C_i$, $i = 1, 2, 3$, and the observation, i.e., the observed values A^* and B^* for the two conditional attributes are given in Table II. For the first attribute A, the distances between A_i, $i = 1, 2, 3$ and the observed A^* are calculated as 4, 4, and 8 respectively (assuming the center of core Rep is adopted). According to (26), the weights are calculated as 0.25, 0.25, and 0.13 respectively. They are normalized using (27), resulting in the new weights of 0.4, 0.4, and 0.2. According to (28), a fuzzy term $A''(5.4, 6.6, 8.0)$ is obtained using these normalized weights. As A'' does not have the same Rep as the input A^*, it is shifted so that it has the same Rep as A^*. According to (30), $\delta_A = -0.11$ is computed. The fuzzy term A'' and δ_A are then used to generate the required intermediate fuzzy set $A'(3.8, 5, 6.4)$. Similarly, $B''(5.89, 7.33, 8.33)$ is constructed with the normalized weights of B_1, B_2 and B_3 (0.33, 0.44, and 0.22, respectively). $B'(4.56, 6, 7)$ is then computed based on B'' and $\delta_B = -0.11$. For the consequent, fuzzy set $C''(6.33, 7.7, 8.7)$ is computed using the average weights of attributes A and B (0.37, 0.42, 0.21) according to (32). The intermediate output $C'(4.76, 6.13, 7.13)$ is then calculated using the average of δ_A and δ_B, that is -0.11, with respect to (34). The overall computational process is summarized in Fig. 5.

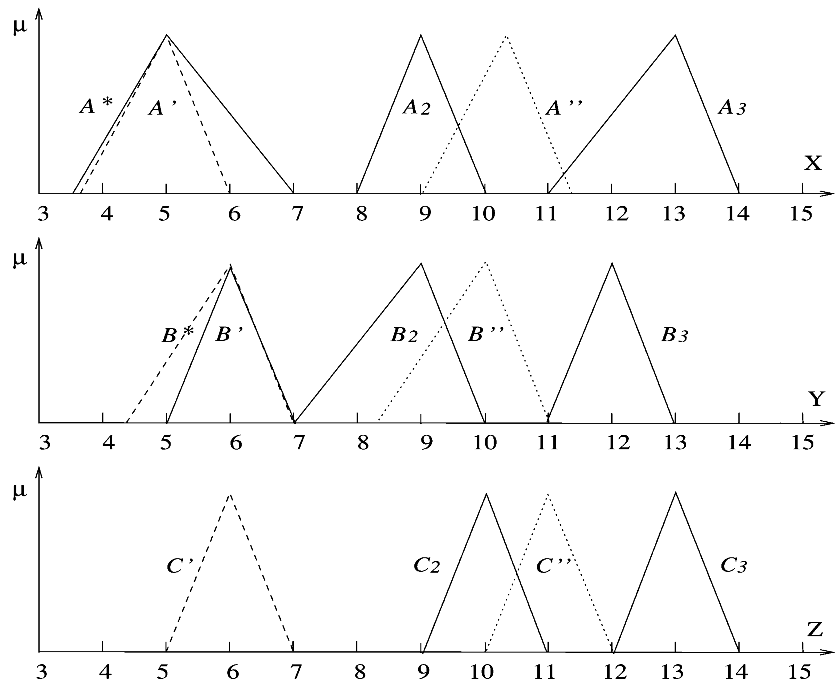


Fig. 6. Example 3: Extrapolation as a specific case of interpolation.

TABLE II
EXAMPLE 2

Attribute Values	$A_1 = (0,1,3)$, $B_1 = (1,2,3)$, $C_1 = (0,2,3)$ $A_2 = (8,9,10)$, $B_2 = (7,9,10)$, $C_2 = (9,10,11)$ $A_3 = (11,13,14)$, $B_3 = (11,12,13)$, $C_3 = (12,13,14)$
Observation	$A^* = (3.5,5,7)$, $B^* = (5,6,7)$

B. Extrapolation

The extension of the above to perform extrapolation is readily attainable. It is a special case of interpolation as indicated in Section IV-A. In particular, when all of the n closest rules chosen (see Section IV-A1) lie on one side of the given observation, the interpolation problem becomes extrapolation. Both choosing the closest rules and constructing the intermediate rule are carried out in the exactly same way as those procedures for interpolation as described in Section IV-A.

An example follows to explain the computation. Suppose that only the second and third rules in Example 2 are considered, the interpolation becomes an extrapolation of two rules.

Example 3: Two rules $A_i \wedge B_i \Rightarrow C_i$, $i = 2,3$ and the observation A^*, B^* as given in Table II are used to carry out fuzzy extrapolation in this example. Again, assume the center of core Rep is used. For the first attribute A, the normalized weights of A_i, $i = 2,3$ are computed to be 0.67 and 0.33. According to (28), a fuzzy term $A''(9, 10.33, 11.33)$ is obtained. Since A'' does not have the same Rep as A^*, it has to be shifted. According to (30), $\delta_A = -0.38$ is obtained. Fuzzy term A'' and δ_A are used to generate the required intermediate fuzzy set $A'(3.67, 5, 6)$. Similarly, B_2 and B_3 have normalized weights 0.67 and 0.33 in constructing the intermediate fuzzy set $B''(8.33, 10, 11)$. With $\delta_B = -0.33$, B'' is shifted to $B'(4.33, 6, 7)$. The fuzzy set $C''(10, 11, 12)$ can then be computed using the average weights of A and B (0.67, 0.33) according to (32). The intermediate output $C' = (5, 6, 7)$ is then computed using the average of δ_A and δ_B, that is -0.36, with respect to (34). This is shown in Fig. 6.

Note that the rules which are used for extrapolation may be twisted. That is, their associated fuzzy sets may not have the same order (as in Example 3) for each attribute. The following illustrates this case.

Example 4: Two new rules $A_2 \wedge B_3 \Rightarrow C_2$ and $A_3 \wedge B_2 \Rightarrow C_3$, and the observation A^*, B^* of Table II are given for fuzzy extrapolation. For the first attribute A, $A''(9, 10.33, 11.33)$ is obtained with the normalized weights of A_i, $i = 2,3$, being 0.67 and 0.33. Fuzzy term A'' is shifted (with $\delta_A = -0.38$) to $A'(3.67, 5, 6)$. Similarly, B_2 and B_3 have normalized weights 0.33 and 0.67 in constructing $B''(8.33, 10, 11)$. With $\delta_B = -0.33$, B'' is shifted to $B'(4.33, 6, 7)$. Fuzzy set $C''(10.5, 11.5, 12.5)$ is then computed using the average weights 0.5 and 0.5. The intermediate output $C'(5.5, 6.5, 7.5)$ can, thus, be computed using the average of δ_A and δ_B, that is -0.36, with respect to (34). This is shown in Fig. 7.

V. EXPERIMENTAL RESULTS

Fuzzy interpolation methods not only help reduce rule bases by removing fuzzy rules which can be approximated by their neighboring rules, but also support reasoning in sparse fuzzy rule bases. In this section, the truck backer-upper control problem shows how the extension of work [1] (denoted as HS hereafter) helps simplify a fuzzy rule base. Further, the problem

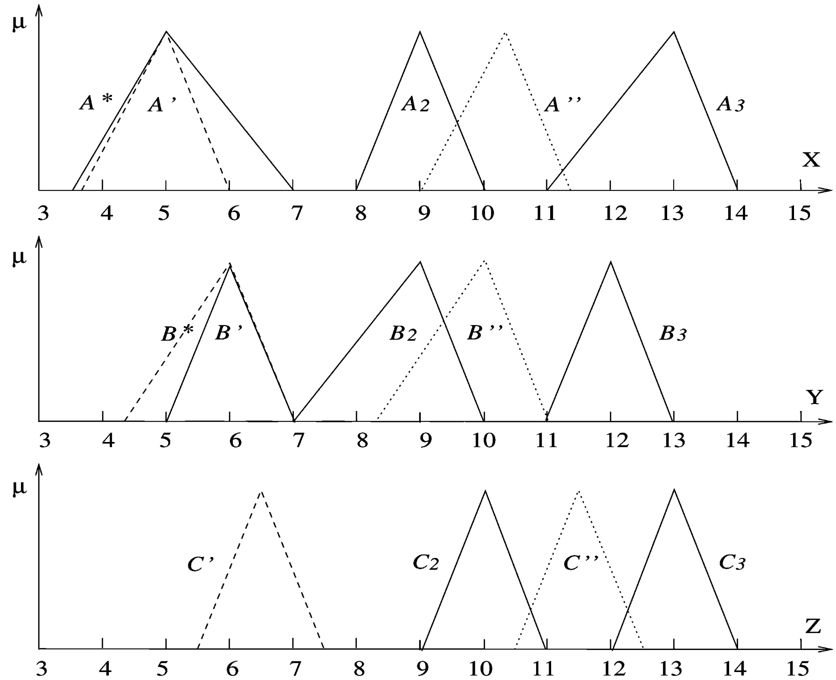


Fig. 7. Example 4: Extrapolation with twisted rules.

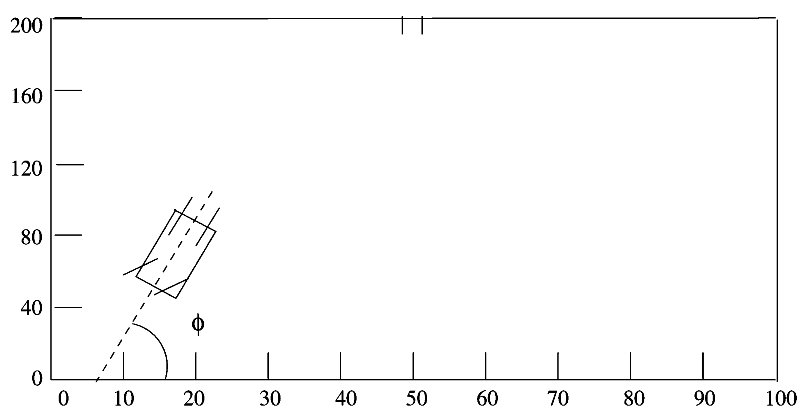


Fig. 8. Truck backer-upper system.

of computer activity prediction shows how the extension serves as a fuzzy inference mechanism for a sparse rule base. In addition, results are compared to those obtained by the most widely used fuzzy inference method, Mamdani inference, and by other interpolation-based inferences, including the general [9], QMY [7], and linear HS method [24].

A. Truck Backer-Upper Control

To demonstrate the usage of the extended interpolation method, the truck backer-upper problem [25]–[28] is considered in this section. This problem is a well-known benchmark in nonlinear control and, thus, has attracted much interest in the literature. It can be illustrated in Fig. 8: The small cab is the truck whose behavior can be determined by three state variables $x \in [0, 100]$, $y \in [0, 200]$ and $\phi \in [-90, 270]$, where x and y are the coordinate values for horizontal and vertical axes respectively, and ϕ is the azimuth angle between the horizontal axis and the truck's onward direction. The truck begins from a certain initial position (x_0, y_0, ϕ_0) and should reverse to the desired end point (50, 200) with desired azimuth angle 90. To control the truck, the steering angle $\theta \in [-30, 30]$ should be provided after every small move made by the truck. The control problem can, thus, be formulated as $\theta = f(x, y, \phi)$. Typically, it is assumed that enough clearance between the truck and the loading dock exists so that the truck y-position can be ignored, simplifying the controller function to $\theta = f(x, \phi)$.

This example involves the case of FISMAT [29], which has nine fuzzy rules as shown in Fig. 9, with each row interpreted as a fuzzy rule

$$\text{IF } x \text{ is } A \text{ AND } \phi \text{ is } B \text{ THEN } \theta \text{ is } C$$

where A, B, and C are the linguistic labels of the system variables. Controlled by these nine fuzzy rules, the truck backing trajectories for four initial points are shown in Fig. 10. All four trajectories roughly converge to the destination point (50, 200). The reaching positions of these four trajectories are given in the second row of Table III.

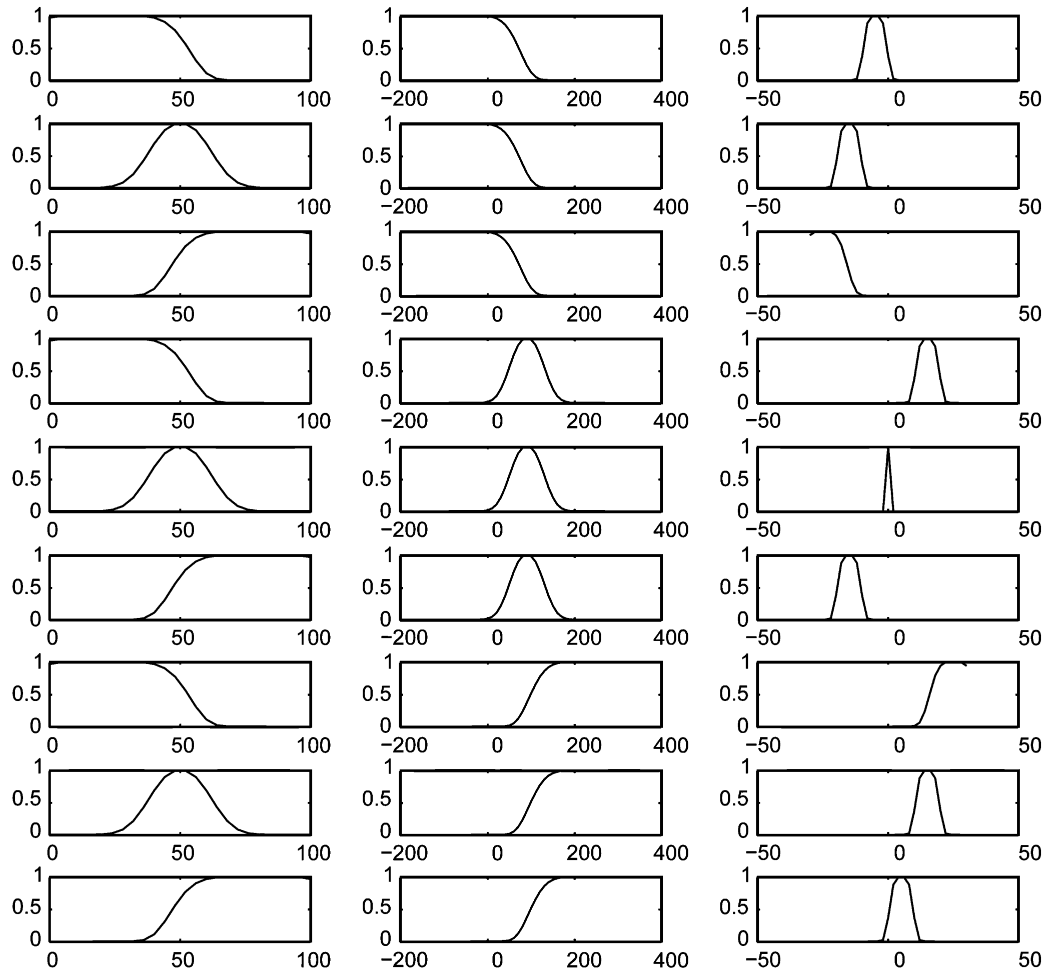


Fig. 9. Membership functions for nine rules.

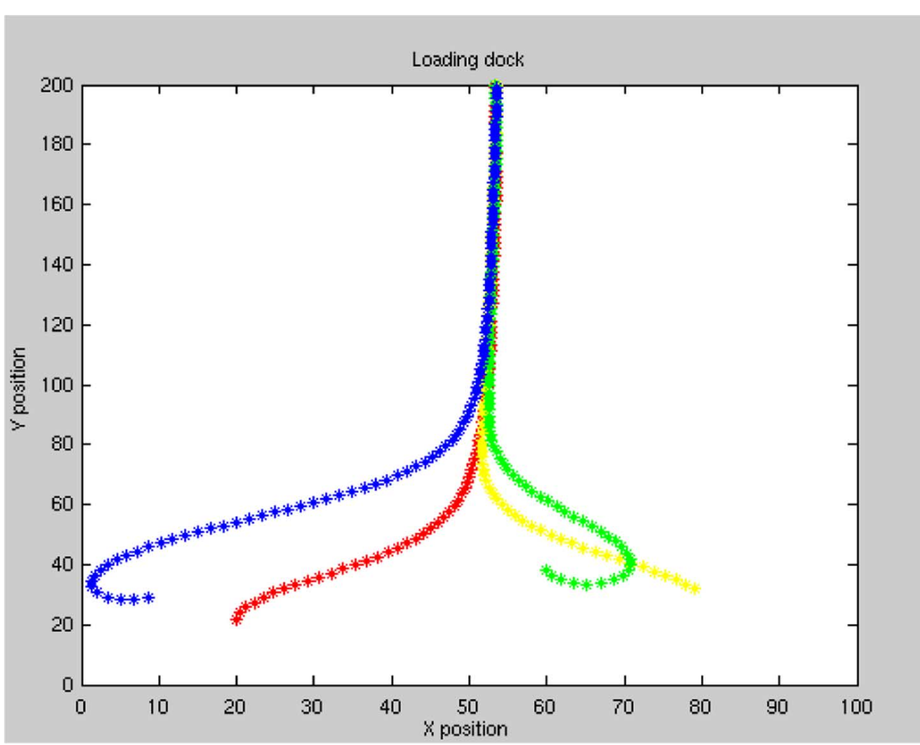


Fig. 10. Trajectories from the use of nine fuzzy rules.

TABLE III
REACHING POSITIONS

Initial states	(20,20,90)	(80,30,120)	(60,40,-90)	(10,30,220)
9 rules without interpolation	(53.35, 89.69)	(53.45, 90.52)	(53.37, 90.35)	(53.37, 90.58)
6 rules without interpolation	(53.44, 89.51)	(53.40, 90.45)	(53.43, 90.84)	(53.48, 90.84)
6 rules with interpolation	(49.68, 84.65)	(49.49, 84.83)	(49.84, 97.97)	(49.71, 97.98)

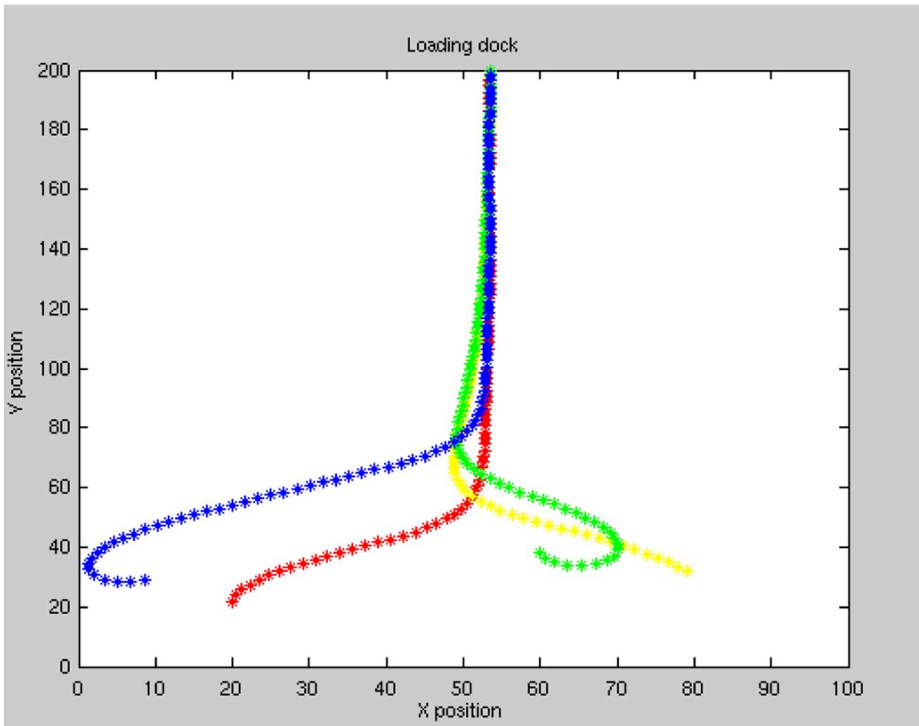


Fig. 11. Trajectories from the use of six fuzzy rules.

For this illustrative example, such an expert fuzzy controller may not suffer from the curse of dimensionality, thanks to the current computational potential. Yet, in general, the number of rules increases exponentially as the input variables and the fuzzy linguistic labels associated with each variable increase. This is because each domain partition has to be covered by at least one fuzzy rule, without the use of interpolation, in order to ensure completeness of the inference.

It is, however, interesting to notice that, for the given rule base, domain partitions appear to be symmetrical in some sense. For example, rule 4 and rule 6 are symmetrical if they are mirrored by rule 5: both rules 4 and 6 have the same ϕ, and they are symmetrical for attribute x and θ from rule 5's point of view. This indicates that rule 5 can be interpolated by rules 4 and 6, and, therefore, it may be removed from this fuzzy controller. Similarly, rules 2 and 8 may be removed as they can be interpolated by rules 1 and 3, and rules 7 and 9, respectively. In doing so, a much more compact fuzzy controller which consists of only six fuzzy rules is obtained. The trajectories and reaching positions of the truck controlled by the remaining six fuzzy rules, without interpolation, are respectively shown in Fig. 11 and the third row of Table III. It can be seen that the reaching positions still roughly converge to the destination point.

However, as the rule base becomes more sparse (due to the removal of rules 2, 5, and 8), it is possible that no fuzzy rules may fire for a given observation (truck state here), although this does not happen in this particular experiment. Yet, if the firing strength threshold is set to be 0.7 (that is, any rule fires only if the firing strength is greater than 0.7), then no rule fires given the observation whose x is around 50 and ϕ is around 90. This leads to the sudden breaks of the trajectories as shown in Fig. 12.

Fuzzy interpolation technique can be employed to support the application for this problem. A possible solution is to predetermine a threshold to decide which inference mechanism (Mamdani or fuzzy interpolation based) should be applied. That is, for a given observation under certain firing strength, the rule base should be treated as sparse. Therefore, the interpolation-based inference becomes a natural choice where otherwise no rules may fire. In this experiment, the threshold is set to be 0.72 after several trials. With the interpolation using two closest rules, Fig. 13 and the fourth row of Table III show the results of performing inferences by following this approach (with the threshold set to 0.72). It is interesting to note that all four trajectories better converge to the destination, although with a slightly more azimuth error.

This experiment clearly demonstrates that the interpolation method can help to simplify a given rule base and to support inferences in a sparse rule base. First, it removes the fuzzy rules which can be approximated (interpolated) by their neighboring rules, resulting in a more compact rule model. This alleviates

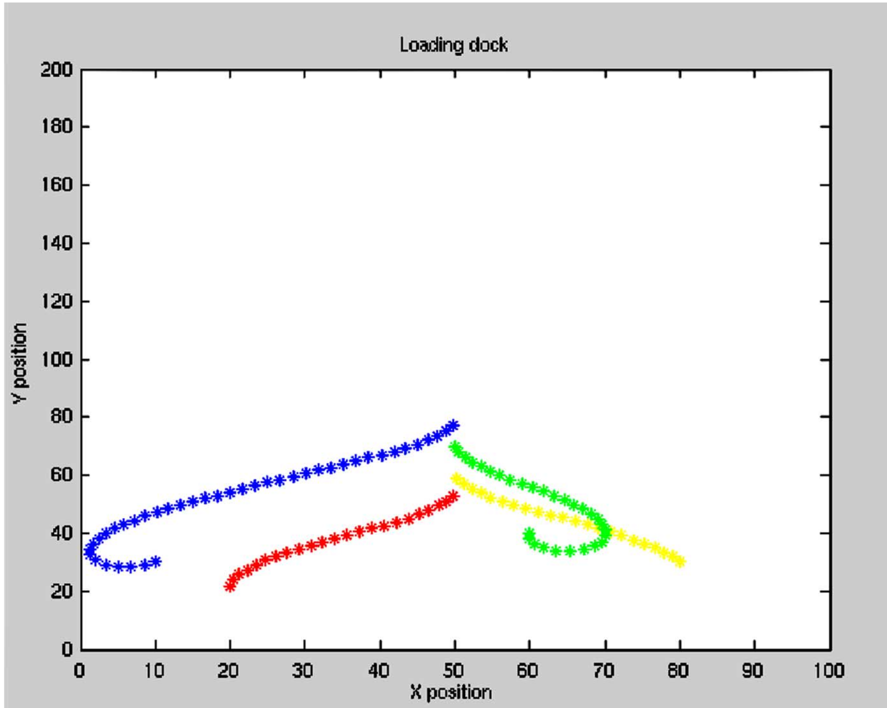


Fig. 12. Sudden breaks of trajectories for six fuzzy rules with rule-firing threshold being 0.7.

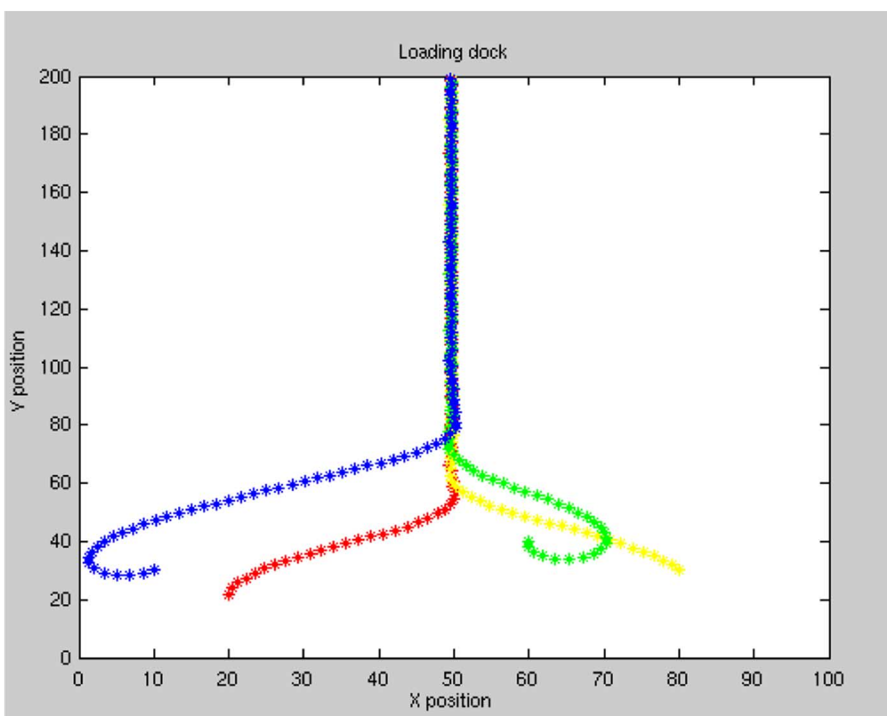


Fig. 13. Trajectories from interpolation with six fuzzy rules.

the curse of dimensionality by keeping important rules only, rather than using all possible rules. Second, as an alternative for performing traditional fuzzy inferences (such as Mamdani and Sugeno), it helps generate the results even when no fuzzy rules may fire with certain firing strengths for the traditional approaches.

B. Computer Activity Prediction

The computer activity database [30] contains a collection of a computer system activity measures. It includes 8192 cases, with each involving 22 continuous attributes. The task is to predict the numeric value of attribute *usr* based on all other attributes.

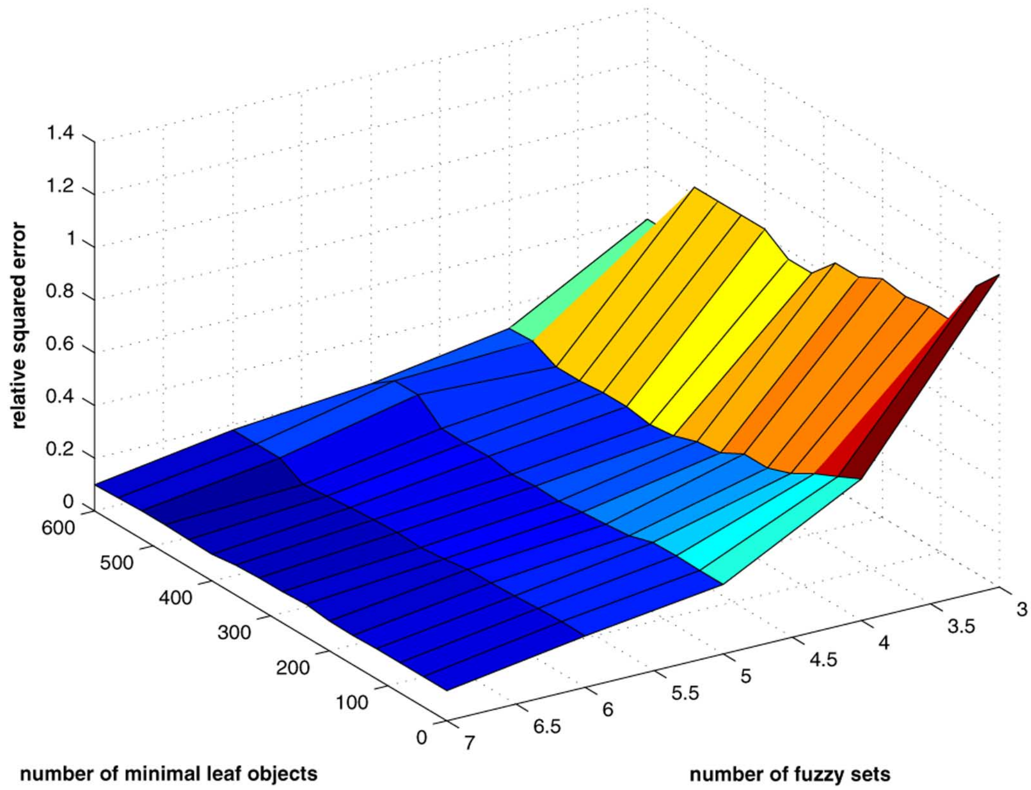


Fig. 14. Relative squared error of fuzzy ID3s.

In this experiment, the database is divided into a training and a test dataset. The training dataset covers approximately $2/3$ of the whole database (5462 cases, to be precise) and the test dataset containing the rest. As there exists redundant or less relevant information in the initial 22 attributes, feature selection is carried out to select the most informative ones. For simplicity, the correlation-based feature subset selection [31], [32] is used and it selects 11 (*read, small, sread, swrite, exec, rchar, pflt, vflt, runqsz, freeswap, and usr*) attributes.

The well-known fuzzy ID3 training scheme [33] is adopted to form fuzzy rule bases on which different fuzzy inference methods including Mamdani and interpolation-based ones, can be applied and then compared with each other. For simplicity, the triangular fuzzy sets are used and they are assumed to be evenly distributed over each attribute domain. Fuzzy ID3 with different configurations (in terms of the number of fuzzy sets and the number of minimal leaf objects) are carried out and the relative squared errors (relative to the simple average predictor) are shown in Fig. 14. This reveals a general trend in that the more fuzzy sets used in the training, the better performance the resulting rules have. However, the number of rules may become very large at the same time. For instance, with the number of minimal leaf objects being 1, the resulting rule base size quickly increases from 55 to 477 if the number of fuzzy sets increases from 3 to 7. For comprehensibility a fuzzy decision tree which has 47 rules and an error rate of 13.29% is chosen (where the number of fuzzy sets is six per attribute and the number of minimal leaf objects is 480) to be used in comparing different fuzzy inference methods. Note that in this rule base, four among the 2730 test data are not fired by any of the 47 rules using Mamdani inference. That is, the obtained rule base is not complete, involving sparse regions.

The errors shown in Fig. 14 are those produced by using Mamdani fuzzy inference. Now, the interpolation-based inference techniques are compared against it using the same rule base and same test data. Note that the methods used here, including general [9], [10], QMY [7], linear HS [24], and HS methods all make use of intermediate rules in performing interpolation. In particular, the solid cutting method [10] and the revision principle-based technique [9] are deployed amongst the family of general interpolations. Other methods such as KH [2], [34], and modified KH [12], [35], which perform without making use of intermediate rules, are not considered here.

In this experiment, the test dataset may be preprocessed before use. In particular, the data are fuzzified to isosceles triangular fuzzy sets, by assigning appropriate support lengths with the center of a fuzzified observation to be the same as its original crisp value. For example, with respect to an attribute, $1/8$ fuzzification assigns $1/8$ of the support length of the fuzzy terms that are used in the rule base to that of the test data object. This fuzzification is purely implementational. The reason for applying fuzzification is that the test data may not be precise in practice due to various factors in data collection. Of course, fuzzification 0 is allowed in which no fuzzification is performed.

It is possible that some attribute values of the intermediate rule exceed the domain limit of that attribute. This is because during the construction of the intermediate rule, extrapolation may be involved and it may lead to the intermediate fuzzy terms being out of range. Also, it is possible that the fuzzified test data exceed the domain limit. If either of these two cases occurs, gen-

TABLE IV
RELATIVE SQUARED ERRORS

Fuzification	0	1/8	1/4
General	8.45%	60.01%	56.53%
QMY	8.05%	7.62%	7.60%
Linear HS (center of core)	8.05%	7.81%	7.80%
Linear HS (average)	6.92%	6.92%	6.92%
Linear HS (average weighted)	6.22%	9.53%	18.86%
HS (center of core)	8.05%	7.58%	7.20%
HS (average)	6.92%	6.92%	6.92%
HS (average weighted)	6.22%	6.25%	6.28%

eral interpolation ignores this attribute in performing interpolation as it simply cannot handle these two cases. For QMY and the linear HS methods, they suffer from another problem as they cannot handle the case where the intermediate rule has a vertical slope (on either the left or right side) for a certain attribute. Such an attribute is ignored if it occurs. Fortunately, all these issues are not a problem for the HS method. Thus, no attributes would be dropped in performing HS interpolations or extrapolations.

The results of using different methods with respect to various fuzzifications of the test data are shown in Table IV. The words within parenthesis indicate which type of Rep is employed for linear HS or HS methods. Note that all the errors shown are the average of the errors in interpolating two and three closest rules. These results clearly show that all interpolation-based inferences (except some cases of the general interpolation) outperform Mamdani inference. The reason of the poor performance for the general is that it drops too many attributes during the interpolation process (if either intermediate fuzzy terms or the fuzzified test data exceed the domain limit), leading to substantial information loss. On the contrary, as the HS method does not need to drop any attributes, it generally results in a stable and high performance. QMY and the linear HS method are between those two extremes. Nevertheless, they still generate better performance than that produced by Mamdani. It is worth noting that the attribute dropping is not part of the general interpolation method, it is simply made so to facilitate comparisons. There may exist other approaches, on which the general interpolation can hopefully perform better.

The best performance is 6.22% when the HS or linear HS interpolation is used and no fuzzification is made for the test data. This error is even less than half of the original error rate of 13.29%, which was produced by Mamdani inference. In addition to the high performance, the interpolation-based inferences are capable of firing all test data including those that are not fired by Mamdani. It is worth noting that the fuzzification of the test data with different support lengths does not significantly affect the prediction error of the HS method. If the average Rep is used for linear HS and HS, the results are exactly the same across different fuzzifications. This is because the value of the average Rep over a fuzzy set is exactly the same as the fuzzified crisp value created from the defuzzification method used (namely, center of gravity) over the same fuzzy set. These results clearly demonstrate the robustness of the HS method.

VI. CONCLUSION

Fuzzy interpolation does not only help reduce the complexity of fuzzy models, but also makes inference in sparse rule-based systems possible. Although fuzzy interpolation techniques have been applied to control problems, they have not yet been reported in prediction or classification applications in machine learning sense as 1) almost all fuzzy interpolation techniques in literature unrealistically assume that there are two closest adjacent rules available to the observation; 2) such rules must flank the observation for each attribute; and 3) some interpolation methods cannot handle the case when fuzzy sets with vertical slopes are involved.

To make fuzzy interpolation ready to be used in practice, this paper has further extended the work of [1] to deal with interpolations that involve multiple multiantecedent rules, without making the strong assumption that antecedent attributes flank the observation. Furthermore, exploiting the generality of this newly developed method, extrapolation can be performed over multiple multiantecedent rules in a straightforward manner. This extension helps bridge the gap between theory and application.

Two realistic applications, namely truck backer-upper control and computer activity prediction, are given in this paper to illustrate the potential of fuzzy interpolation (and indeed extrapolation as well) in both rule base simplification and inference with sparse rule bases. Comparative studies to Mamdani inference and other existing fuzzy interpolation methods have been provided to show the success of the proposed method.

Further development on the proposed method may be desirable in maximal accordance with Jenei's axioms [18]. Also, work remains to investigate the possible implications of using optimized partition of a problem domain. However, this should not affect the theoretical approach put forward both here and in the original paper [1] which the present work is based on, since the techniques developed do not rely on this implementational issue. In the experimental studies presented in this paper, only evenly partitioned datasets are used and these partitions are not optimized in any sense. It is, therefore, expected that with the use of an optimized fuzzification mechanism, the performance of the proposed work may be further improved. It is also very interesting to apply the work to more complex domains, perhaps in conjunction with the utilization of more powerful feature selection tools (e.g., [36]) than those used in this paper, in an effort to fully realize the aims of developing this practical approach.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their helpful comments on this paper.

REFERENCES

[1] Z. H. Huang and Q. Shen, "Fuzzy interpolation reasoning via scale and move transformations," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 2, pp. 340–359, Feb. 2006.

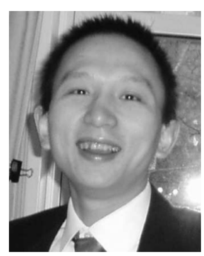
[2] L. T. Kóczy and K. Hirota, "Approximate reasoning by linear rule interpolation and general approximation," *Int. J. Approx. Reason.*, vol. 9, no. 3, pp. 197–225, 1993.

[3] B. Moser and M. Navara, "Fuzzy controllers with conditionally firing rules," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 3, pp. 340–348, Mar. 2002.

[4] Y. Shi and M. Mizumoto, "Some considerations on Kóczy's interpolative reasoning method," in *Proc. IEEE Int. Conf. Fuzzy Systems*, 1995, vol. 4, pp. 2117–2122.

[5] S. Yan, M. Mizumoto, and W. Z. Qiao, "Reasoning conditions on Kóczy's interpolative reasoning method in sparse fuzzy rule bases," *Fuzzy Sets Syst.*, vol. 75, no. 1, pp. 63–71, 1995.

[6] G. Vass, L. Kalmar, and L. T. Kóczy, "Extension of the fuzzy rule interpolation method," in *Proc. Int. Conf. Fuzzy Sets Theory Applications*, 1992, pp. 1–6.
[7] W. Z. Qiao, M. Mizumoto, and S. Y. Yan, "An improvement to Kóczy and Hirota's interpolative reasoning in sparse fuzzy rule bases," *Int. J. Approx. Reason.*, vol. 15, no. 3, pp. 185–201, 1996.
[8] W. H. Hsiao, S. M. Chen, and C. H. Lee, "A new interpolative reasoning method in sparse rule-based systems," *Fuzzy Set Syst.*, vol. 93, no. 1, pp. 17–22, 1998.
[9] P. Baranyi, T. D. Gedeon, and L. T. Kóczy, "A general interpolation technique in fuzzy rule bases with arbitrary membership functions," in *Proc. IEEE Int. Conf. Systems, Man, Cybernetics*, 1996, pp. 510–515.
[10] P. Baranyi, L. T. Kóczy, and T. D. Gedeon, "A generalized concept for fuzzy rule interpolation," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 6, pp. 820–837, Jun. 2004.
[11] P. Baranyi, D. Tikk, Y. Yam, and L. T. Kóczy, "A new method for avoiding abnormal conclusion for α-cut based rule interpolation," in *Proc. IEEE Int. Conf. Fuzzy Systems*, 1999, pp. 383–388.
[12] D. Tikk and P. Baranyi, "Comprehensive analysis of a new fuzzy rule interpolation method," *IEEE Tran. Fuzzy Syst.*, vol. 8, no. 3, pp. 281–296, Mar. 2000.
[13] D. Tikk, P. Baranyi, T. D. Gedeon, and L. Muresan, "Generalization of the rule interpolation method resulting always in acceptable conclusion," *Tatra Mountains Math. Pub.*, vol. 21, pp. 73–91, 2001.
[14] D. Dubois, H. Prade, and M. Grabisch, "Gradual rules and the approximation of control laws," in *Theoretical Aspects of Fuzzy Control*, H. T. Nguyen, M. Sugeno, R. Tong, and R. R. Yager, Eds. New York: Wiley, 1995, pp. 147–181.
[15] B. Bouchon-Meunier, C. Marsala, and M. Rifqi, "Interpolative reasoning based on graduality," in *Proc. IEEE Int. Conf. Fuzzy Systems*, 2000, pp. 483–487.
[16] Y. Yam and L. T. Kóczy, "Cartesian representation for fuzzy interpolation," in *Proc. IEEE Conf. Decision and Control*, 1998, vol. 3, pp. 2936–2937.
[17] Y. Yam and L. T. Kóczy, "Representing membership functions as points in high-dimensional spaces for fuzzy interpolation and extrapolation," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 6, pp. 761–772, Jun. 2000.
[18] S. Jenei, "Interpolation and extrapolation of fuzzy quantities revised—An axiomatic approach," *Soft Comput.*, vol. 5, pp. 179–193, 2001.
[19] S. Jenei, E. P. Klement, and R. Konzel, "Interpolation and extrapolation of fuzzy quantities—The multiple-dimensional case," *Soft Comput.*, vol. 6, pp. 258–270, 2002.
[20] S. Kovács and L. T. Kóczy, *Application of the Approximate Fuzzy Reasoning Based on Interpolation in the Vague Environment of the Fuzzy Rulebase in the Fuzzy Logic Controlled Path Tracking Strategy of Differential Steered AGVs*. Berlin, Germany: Springer, 1997, vol. 1226, pp. 456–467, Lecture Notes in Computer Science.
[21] K. W. Wong, D. Tikk, T. D. Gedeon, and L. T. Kóczy, "Fuzzy rule interpolation for multidimensional input spaces with applications: A case study," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 6, pp. 809–819, Jun. 2005.
[22] K. W. Wong, C. C. Fung, and T. D. Gedeon, *Fuzzy Hydrocyclone Modelling for Particle Separation Using Fuzzy Rule Interpolation*. Berlin, Germany: Springer, 2000, vol. 1983, pp. 103–108, Lecture Notes in Computer Science.
[23] Y. Bai and D. L. Wang, "Improve the position measurement accuracy using a dynamic on-line fuzzy interpolation technique," in *Proc. IEEE Int. Symp. Computational Intelligence for Measurement Systems and Applications*, 2003, pp. 227–232.
[24] Z. H. Huang, "Rule Model Simplification," Ph.D. dissertation, School of Informatics, Univ. Edinburgh, Edinburgh, U.K., 2006.
[25] D. Nguyen and B. Widrow, "The truck backer-upper: An example of self-learning in neural network," *IEEE Control Syst. Mag.*, vol. 10, no. 2, pp. 18–23, Feb. 1990.
[26] S. G. Kong and B. Kosko, "Adaptive fuzzy systems for backing up a truck-and-trailer," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 211–223, May 1992.
[27] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cyben.*, vol. 22, no. 6, pp. 1414–1427, Dec. 1992.
[28] A. Riid and E. Rustern, "Fuzzy logic control: Truck backer-upper problem revised," in *Proc. IEEE Int. Conf. Fuzzy Systems*, 2001, vol. 1, pp. 513–516.
[29] A. Lotfi, "Fuzzy inference system toolbox for matlab (fismat)," 2000 [Online]. Available: http://www.lotfi.net/research/fismat/index.html
[30] C. E. Rasmussen *et al.*, "Data for evaluating learning in valid experiments (delve)," [Online]. Available: http://www.cs.toronto.edu/~delve/ 1995–1996
[31] M. A. Hall, "Correlation-based feature subset selection for machine learning," Ph.D. dissertation, Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, 1999.
[32] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Mateo, CA: Morgan Kaufmann, 1999.
[33] C. Z. Janikow, "Fuzzy decision trees: Issues and methods," *IEEE Trans. Syst., Man., Cybern.*, vol. 28, no. 1, pp. 1–14, Jan. 1998.
[34] L. T. Kóczy and K. Hirota, "Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases," *Inf. Sci.*, vol. 71, no. 1, pp. 169–201, 1993.
[35] D. Tikk, P. Baranyi, Y. Yam, and L. T. Kóczy, "Stability of a new interpolation method," in *Proc. Int. Conf. Systems, Man, Cybernetics*, 1999, vol. 3, pp. 7–9.
[36] Q. Shen and R. Jensen, "Selecting informative features with fuzzy-rough sets and its application for complex systems monitoring," *Pattern Recognit.*, vol. 37, no. 7, pp. 1351–1363, 2004.



Zhiheng Huang received the B.Sc. degree in industrial equipment and control engineering from the South China University of Technology, China, and the M.Sc. and Ph.D. degrees in artificial intelligence from the University of Edinburgh, Edinburgh, U.K.

He is a Research Fellow at Berkeley Initiative Soft Computing with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA. His research interests includes machine learning, data mining, intelligent data analysis, fuzzy set theory, information retrieval, and information extraction.

Qiang Shen received the B.Sc. and M.Sc. degrees in communications and electronic engineering from the National University of Defence Technology, China, and the Ph.D. degree in knowledge-based systems from Heriot-Watt University, Edinburgh, U.K.

He is a Professor with the Department of Computer Science, University of Wales, Aberystwyth, U.K., and an Honorary Fellow at the University of Edinburgh. His research interests include fuzzy and imprecise modeling, model-based inference, pattern recognition, and knowledge refinement and reuse.

Dr. Shen is an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS , and an editorial board member of the *Fuzzy Sets and Systems Journal*, among others. He has published over 180 peer-refereed papers in academic journals and conferences on topics within artificial intelligence and related areas.