Using Single-Antecedent Fuzzy Rules in Fuzzy Knowledge Map

Sebastian W. Khor  
School of IT  
Murdoch University  
Perth, Australia  
s.khor@murdoch.edu.au

M. Shamim Khan  
School of IT  
Murdoch University  
Perth, Australia  
s.khan@murdoch.edu.au

Kok Wai Wong  
School of IT  
Murdoch University  
Perth, Australia  
k.wong@murdoch.edu.au

Abstract

Conventional fuzzy inference methodology relies on the mapping of the input spaces to the output space by partitioning the spaces with membership functions. In cases where there are more than one input variables, an intersection of memberships is adopted by aggregating these regions. This strategy yields an exponential growth in the number of rules as inputs are added to the system, quickly reducing performance to unacceptable levels. We present a methodology that allows the use of single antecedent fuzzy rules to approximate a class of problems in the Fuzzy Knowledge Map - a knowledge representation framework developed by us.

Keywords: Fuzzy logic, single-antecedent fuzzy rules, Fuzzy Knowledge Map, fuzzy modelling.

1. INTRODUCTION

A fuzzy system can always approximate a continuous function if the number of fuzzy sets and fuzzy rules is allowed to increase sufficiently [17, 21]. However, this will exponentially increase the number of rules and thus increase the computation complexity. Fuzzy systems involve the transformation (or mapping) of inputs to output, using linguistic fuzzy rules of the form:

\[
\text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \text{ and } \ldots \text{ and } x_k \text{ is } A_k \text{ then } y \text{ is } B
\]

where \( x_i \) is the \( i^{th} \) input to the fuzzy system, which is defined on the universe of discourse \( X_i \); \( A_i \) is a fuzzy set on \( X_i \); \( y \) is the system output defined on a universe of discourse \( Y \), and \( B \) is a fuzzy set on \( Y \). In this paper, we refer to such conjunctive form of the rules as multi-antecedent fuzzy rules, whereas for fuzzy rules of the form:

\[
\text{If } x_i \text{ is } A_i \text{ then } y \text{ is } B
\]

we refer to them as single-antecedent fuzzy rules.

By allowing partial memberships, it is possible to represent a smooth transition from one rule to another, i.e. interpolate between the rules, the criteria for which we discuss further in the next section. There are many papers in the literature, which show that Mamdani and Takagi–Sugeno (TS) fuzzy systems are good universal approximators. They can uniformly approximate any continuous functions to any degree of accuracy, e.g. [2, 9, 21].

However, the conventional multi-antecedent fuzzy rules lead to the well known combinatorial rule explosion problem [1, 8, 10]. The interpretability of these rules decreases because of the increased complexity of the rules. We will show this evidence in Section 5.

Many techniques have been proposed for rule reduction, such as fuzzy decision trees [5], quad trees [14], singular value decomposition [12], and clustering followed by pruning off the less contributing rules [13]. Since all these techniques are still based on the multi-antecedent rules for inferencing, the potential problems mentioned earlier still remain. We note that in certain problem domain, it is possible to avoid altogether the use the multi-antecedent rules. We propose the use of only single-antecedent fuzzy rules for these cases. These rules are implemented in our Fuzzy Knowledge Map. We show that a class of polynomial functions can be approximated using single-antecedent fuzzy rules. Our experimental simulation indicates that our proposed approach performed better in two aspects: interpretability of rules and output performance.

In the next section (Section 2), this paper introduces some preliminary concepts relating to the characteristics of membership functions relevant in our discussion of the applications of single-antecedent fuzzy rules. In Section 3, we show that the consequents of certain fuzzy rules actually fall on the curve of certain polynomial functions, and explain how this result is implemented in Fuzzy Knowledge Maps in Section 4. We present our simulation in Section 5 and our conclusion in Section 6.

2. PRELIMINARIES

In this section, we provide some definitions to assist in our discussion of the new approach to fuzzy inferencing. These definitions are based on fuzzy theory [18, 19], and can be found from many sources e.g. [11, 16]. We denote \( A_i, i = 1, 2, \ldots, n \) to be the \( i^{th} \) fuzzy set in \( X \subset \mathbb{R} \), and \( A_i(x) \) to be the corresponding fuzzy membership function.
Definition 1. **Pseudotrapezoidal shaped membership function**, or simply PTS function:

Let \([a, d] \subset X \subset \mathbb{R}\). Then a PTS function is defined as a membership function \(A(x) = A(x, a, b, c, d)\) where \(a \leq b \leq c \leq d, x \in X\), and the membership grade \(M = M(A(x)) = [0, 1]\) for \(x \in [a, d]\); strictly monotonically increasing in \([a, b]\); strictly unimodal in \([b, c]\); strictly monotonically decreasing in \([c, d]\); and \(M = 0\) for \(x \in X - [a, d]\).

Figure 1 shows some examples of PTS functions. As can be seen, a triangular shaped function is a special case of trapezoidal shaped function where \(b = c\). We can denote a normal triangular shaped function as \(A_i(x, a, b, d)\). The flank \([b, c]\) forms the core of the function. Where \(b = c\), the core is a point. The flank \([a, d]\) is the support.

**Figure 1**: Examples of PTS membership functions: (a) trapezoidal shaped; (b) triangular shaped; and (c) Gaussian shaped.

For a fuzzy set \(A\) in \(X \subset \mathbb{R}\), by definition 1, the membership grade \(M(A)\) is defined as:

\[
M(A) = \{x \in X \mid A(x) = \sup(A(x))\} \tag{1}
\]

Definition 2. **Completeness of partition**: If for any \(x \in X \subset \mathbb{R}\), there exists \(A_i\) such that \(A_i(x) > 0\), for \(i = 1, 2, \ldots, n\) in the universe of discourse, then the set of fuzzy sets \(A_i\) is said to be a complete partition on \(X\). In short, we say that the set of fuzzy sets \(A_i\) is complete.

Figure 2 shows an example of a set of fuzzy sets with \(A_1 < A_2 < A_3 < A_4 < A_5\); and the set is complete and consistent. Order and consistency are defined in definitions 3 and 4 respectively.

In practical applications, every \(x \in X\) has a possible input and should therefore have a corresponding output [20]. Completeness ensures at least one of the fuzzy IF-then rules will be triggered for every \(x \in X\). This is also known as dense fuzzy rule base.

**Figure 2**: An example of a set of fuzzy sets, illustrating its completeness, consistency, and order.

Definition 3. **Order between fuzzy sets**. Let \(A\) and \(B\) be two fuzzy sets in \(X \subset \mathbb{R}\). \(A < B\) if \(M(A) < M(B)\), i.e. \(\forall x_0 \in M(A)\) and \(\forall x_0 \in M(B)\), \(x_0 < x_0\).

**Figure 2** shows that fuzzy sets \(A_1 < A_2 \prec A_3 < A_4 < A_5\). It follows that in this case, \([b_1, c_1] \leq [b_2, c_2] \leq [b_3, c_3] \leq [b_4, c_4] \leq [b_5, c_5]\).

**Lemma 1**: Let \(A_i\) be \(n\) number of fuzzy sets in \(X \subset \mathbb{R}\) where \(i = 1, 2, \ldots, n\) and PTS membership functions \(A_i(x) = A_i(x, a_i, b_i, c_i, d_i)\). Then \(A_1 < A_2 \prec \ldots < A_n\) if and only if \([b_1, c_1] \prec [b_2, c_2] \prec \ldots \prec [b_n, c_n]\), i.e. \(b_1 \leq c_1 < b_2 \leq c_2 < \ldots < b_n \leq c_n\). Furthermore, as can be seen from Figure 1, \([b_i, c_i] \not\subset \mathbb{R}\), if \([b_i, c_i] \cap [b_j, c_j] = \mathbb{R}\). That is, \(A_i < A_j\) iff \([b_i, c_i] \cap [b_j, c_j] = \mathbb{R}\) for all \(i \neq j\).

**Proof**

Let there be \(n\) number of fuzzy sets in the universe of discourse \(X \subset \mathbb{R}\) with PTS membership functions \(A_i(x) = A_i(x, a_i, b_i, c_i, d_i)\). Then by definition 3, for \(\forall x_0 \in M(A_i)\) and \(\forall x_0 \in M(A_j)\), \(i \neq j, x \in X\), \(x_0 < x_0\) then \(A_i < A_j\). From eq.(1), \(M(A_i) = [b_i, c_i] \) and \(M(A_j) = [b_j, c_j]\). That is, \([b_i, c_i] < [b_j, c_j]\) if \(x_0 < x_0\). Therefore \(A_i < A_j\) if \([b_i, c_i] < [b_j, c_j]\), i.e. \([b_i, c_i] \not\subset [b_j, c_j]\) and \([b_j, c_j] \not\subset [b_i, c_i]\). That is, \([b_i, c_i] \cap [b_j, c_j] = \mathbb{R}\).

Suppose \([b_i, c_i] \cap [b_j, c_j] \not= \mathbb{R}\). Then there exists an \(x\) such that \(x \in [b_i, c_i]\) and \(x \in [b_j, c_j]\). However, definition 3 requires that all PTS functions are ordered such that \(A_i < A_j\) if \(x_0 < x_0\) for all \(x_0 \in [b_i, c_i]\) and \(x_0 \in [b_j, c_j]\). Therefore, for all \(x_0 \in [b_i, c_i]\) and \(x_0 \in [b_j, c_j]\) and \(i \neq j, x_0\) cannot be equal to \(x_0\). That is, \([b_i, c_i] \cap [b_j, c_j] = \mathbb{R}\). Therefore, \(A_i < A_j < \ldots < A_n\) iff \([b_i, c_i] \prec [b_j, c_j] \prec \ldots \prec [b_n, c_n]\) and \([b_i, c_i] \cap [b_j, c_j] = \mathbb{R}\) where \(i \neq j\).

**Corollary**: Lemma 1 implies that the cores of normal fuzzy sets in \(X\) do not overlap.

Definition 4. **Consistency**: Let \(A_i\) be \(n = 1, 2, \ldots\) be \(n\) fuzzy sets in \(X \subset \mathbb{R}\). If \(A_j(x_0) = 1\) for some \(x_0 \in X \subset \mathbb{R}\), then for all \(j \neq i\) (where \(j = 1, 2, \ldots, n\), \(i = 1, 2, \ldots, n\), \(A_j(x_0) \neq 1\) and fuzzy sets \(A_i\) are said to be consistent on \(X\).

Consistency allows arguments based on human
thinking logic [20]. We do not want a case where, for example, a person is definitely old and at the same time definitely middle-aged although we do want to be able to express the case where a person is somewhat old and somewhat middle-aged. This is also to ensure that the linguistic terms are in an ordering manner, e.g. freezing, cold, moderate, warm, and hot.

Definition 5: Convexity:
A fuzzy set $A$ is convex if its membership function is monotonically increasing and then decreasing without any saddle point in the middle. That is,

$$\forall r, s \in X, \lambda \in [0, 1], A(t) \geq \min \{A(r), A(s)\}$$

where $t = \lambda r + (1 - \lambda) s$

Convexity ensures smooth transgression of membership functions. Figure 3 (a) shows an example of a convex fuzzy set and (b) shows a non-convex.

Figure 3: Fuzzy set is (a) convex: $A(t) \geq A(r)$, and (b) non-convex: $A(t) < A(r)$

Figure 4: Common fuzziness. The left-flank of $A_j$ is also the right-flank of $A_{j+1}$, and the right-flank of $A_j$ is also the left-flank of $A_{j+1}$. Left-flank + right-flank = interval between peaks of $A_{j+1}$ and $A_{j+2}$.

Let $A_i, i = 1, 2, \ldots, n$ be $n$ fuzzy sets in $X \subset \mathbb{R}$. Let $A_j$ be one of the fuzzy sets in $A_i$. Then $A_{j+1}$ and $A_{j+1}$ are the adjacent fuzzy sets. Then the fuzziness of the left-flank of the membership function $A_j$ is equal to the fuzziness of the right-flank of membership function $A_{j+1}$, and the fuzziness of the right-flank of $A_j$ is equal to that of the left-flank of $A_{j+1}$; and the sum of both left and right flanks of $A_j$ is equal to the length of the interval between the peak values of the two adjacent membership functions $A_{j+1}, A_{j+1}$.

Definition 6. Common fuzziness:
Let $A_i, i = 1, 2, \ldots, n$ be $n$ fuzzy sets in $X \subset \mathbb{R}$. Let $A_j$ be one of the fuzzy sets in $A_i$. Then $A_{j+1}$ and $A_{j+1}$ are the adjacent fuzzy sets. Then the fuzziness of the left-flank of the membership function $A_j$ is equal to the fuzziness of the right-flank of membership function $A_{j+1}$, and the fuzziness of the right-flank of $A_j$ is equal to that of the left-flank of $A_{j+1}$; and the sum of both left and right flanks of $A_j$ is equal to the length of the interval between the peak values of the two adjacent membership functions $A_{j+1}, A_{j+1}$.

Lemma 2: Let a set of $n$ number of fuzzy sets in $X \subset \mathbb{R}$ where $i = 1, 2, \ldots, n$ and PTS membership functions $A_i(x) = A_i(a_i, b_i, c_i, d_i)$ that are complete, ordered, consistent, convex and conform to the common fuzziness defined in definition 6. Then any crisp input (represented by a vertical line in Figure 4) will ‘cut’ through one to two at most, membership functions.

Definition 7: Simple multivariate polynomial:
It is defined as a polynomial function that contains more than one variable but no summands that are formed by the product of two or more of these variables. In other words, it contains only monomials, i.e. the individual summands with the coefficients only.

Here is an example of a polynomial in two variables $x$ and $z$ with three terms that are products of these variables and it is therefore not a simple multivariate polynomial:

$$y = 3x^4 + x^3z + 4x^2z^3 + 2xz^3 + 6z^4$$

Here are some examples of simple multivariate polynomials:

$$y = 3x^4 + 2x^3 + z + 2x^2 + 2z^2 + x + 6z^4$$

$$y = 3x^4 + 6t^3 + 7$$

$$y = 2x + 3t - 4z + 6$$

Lemma 3: A simple multivariate polynomial can always be decomposed into univariate polynomials. Similarly, two or more univariate polynomials can be composed into a single simple multivariate polynomial.

Proof
The proof is trivial. Since the summands of a simple multivariate polynomial are not formed by the products of the variables, we can always make use of the associative property of polynomial to group the summands of a simple...
multivariate polynomial such that the summands within each parenthesis form a univariate polynomial. Similarly univariate polynomials can be composed into a simple multivariate polynomial by adding the summands of the polynomials together.

Corollary: A simple multivariate polynomial can thus be considered as a composite polynomial comprising two or more univariate polynomials.

Corollary: As there is no summand that is the product of the variables in a simple multivariate polynomial, these variables are said to be independent of each other, in the sense that the net contribution of an input is independent of the value of the input of other variables.

As an example, in the polynomial \( y = 2x^2 + 3x + 2z + 3z^2 \), we can decompose the polynomial to \( y_1 = 2x^2 + 3x \), and \( y_2 = 2z + 3z^2 \). Then the contribution due to \( x \) is \( y_1 \) and that due to \( z \) is \( y_2 \). The value of \( y_1 \) is not affected by whatever the value of the input \( z \) is, and vice versa.

However, if, for example, \( z \) is a function of \( x \), then the polynomial in the above example is no more a simple multivariate polynomial, and \( x \) and \( z \) are no more independent of each other.

Definition 8: A single-antecedent fuzzy rule is defined as a fuzzy rule in the form “if \( x \) is \( A \), then \( y \) is \( C \)”, in the product space \( R = X \times Y \), where \( x \in X \) and \( y \in Y \), and has a relationship membership function \( R(x, y) \).

In a multi-antecedent fuzzy rule, the rule antecedent consists of two or more input variables \( x_i \), \( 1 \leq i \leq n \). A class of operators called the triangular norms or t-norms are applied to aggregate the antecedents to derive the firing strength of the antecedents. Depending on the inference mechanism used, a defuzzification process may be needed to determine the output of the consequent.

3. COMPOSITION OF A FUNCTION

We can view a polynomial as data points of the function of the form of an \( n-1 \) degree polynomial in \( x \), \( P_n(x) = \sum_{i=0}^{n} a_i x^i \) and the solution is reduced to one of approximating the polynomial. We will prove this point as follows.

Consider the case where there exist \( n+1 \) data points \((x_i, y_i)\) where no two \( x_i \) are the same. Then there exists at most an \( n \) degree polynomial of the form \( y = P_n(x) = \sum_{i=0}^{n} a_i x^i \). We can then substitute all the data points in \( P_n(x) \) and solve the \( n+1 \) linear simultaneous equations to obtain the coefficients \( a_i \).

An alternative method is to make use of Lagrange interpolation to obtain the coefficients. In that case, we can express the polynomial as:

\[
P_n(x) = L_0(x) y_0 + L_1(x) y_1 + \ldots + L_n(x) y_n = \sum_{i=0}^{n} L_i(x) y_i \tag{2}
\]

where

\[
L_i(x) = \frac{(x - x_0)(x - x_1)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)(x_i - x_1)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)} \]

where \( i, k = 1, 2, \ldots, n \), and such that

\[
L_i(x) = \begin{cases} 1 & \text{if } k = i \\ 0 & \text{if } k \neq i \end{cases}
\]

For the case of degree = 1 (i.e. first order or linear) with the polynomial passing through two points \((x_1, y_1)\) and \((x_2, y_2)\), the Lagrange polynomial is given by:

\[
P_i(x) = L_i(x) y_1 + L_2(x) y_2 = \frac{(x - x_2)}{(x_i - x_2)} y_1 + \frac{(x - x_1)}{(x_i - x_1)} y_2 \tag{3}
\]

For \( x = x_1 \), \( L_1 = 1 \) and \( L_2 = 0 \), so \( P_1(x_1) = y_1 \), i.e. it contains \((x_1, y_1)\).

For \( x = x_2 \), \( L_1 = 0 \) and \( L_2 = 1 \), so \( P_1(x_2) = y_2 \), i.e. it contains \((x_2, y_2)\).

3.1 Case of a single input

Suppose there exists a set of fuzzy sets that partitions the input space completely, and the membership functions are ordered, consistent, and convex. Suppose that these functions share the common fuzziness as defined in definition 6, and they are PTS functions. Then by Lemma 2, an input singleton \( x \) belongs to one or at most two PTS functions. Suppose the PTS functions are normal. Then \( x \) will have a degree of membership of \([0, 1]\).

A single-antecedent fuzzy rule can be expressed in the form:

\[
\text{If } x \text{ is } A_i, \text{ then } y \text{ is } f_i(x) \tag{4}
\]

where \( A_i \) is a fuzzy set and \( f_i(x) \) is a function of \( x \).

An input will trigger the rules in two situations:

1) In the case where the input \( x \) cuts at one point in the set of PTS functions, the output is determined by the consequent of a single-antecedent fuzzy rule.

2) In the case where the input \( x \) cuts at two points, the output is determined by two rules:

   Rule 1: If \( x \) is \( A_1 \), then \( y = y_1 \cdot f_1(x) \)

   Rule 2: If \( x \) is \( A_2 \), then \( y = y_2 \cdot f_2(x) \)

According to Takagi-Sugeno (TS) fuzzy system, the output is given by:

\[
\text{If } x \text{ is } A_i, \text{ then } y = \sum_{i=1}^{m} \left( \alpha_i \cdot f_i(x) \right)
\]

where \( \alpha_i \) is the membership degree of \( x \) in \( A_i \) and \( f_i(x) \) is a function of \( x \).
\[ y^*(x) = \frac{w_1 y_1 + w_2 y_2}{w_1 + w_2} \]  

(5)

When \( \sum_{j=1}^{2} w_j = 1 \) (i.e. in this case \( w_1 + w_2 = 1 \)) for normalized \( w \), eq.(4) becomes:

\[ y^*(x) = w_1 y_1 + w_2 y_2 \]  

(6)

The functions \( y_1 \) and \( y_2 \) can be constants or linear functions. As would have been noted, this is a case of SISO (single input single output) model. The inference and defuzzification of an input singleton is shown in Figure 5. As can be seen, \( w_1 = (x - x_2)/(x_1 - x_2) \), and \( w_2 = (x - x_1)/(x_2 - x_1) \). Substitute these in eq.(5), we have:

\[ y^*(x) = \frac{(x - x_1)}{(x_1 - x_2)} y_1 + \frac{(x - x_2)}{(x_2 - x_1)} y_2 \]  

(7)

The result is the same as eq.(3). That is, the crisp output of a TS fuzzy system of a univariate input is a point on the polynomial function.

Often we want to think the total output in terms of a normalized output. Here we may weight the contribution of each term such that the total contributions fall within \([0, 1]\) as given by the equation:

\[ y = \beta_1 y_1^* + \beta_2 y_2^* + ... + \beta_n y_n^* \]  

(8)

where \( \sum_{j=1}^{n} \beta_j = 1 \).

\( \beta \) is the weighting of the contribution of each term. In many practical situations, the contributions can be equal, or some terms may have a higher contribution at the expense of others.

An example of the process involving two input variables is shown in Figure 6. As can be seen, the output as in eq. (8) is reduced to \( y = \beta_1 y_1^* + \beta_2 y_2^* \).

3.2 Case of two or more independent input variables

We will follow the same assumptions as in the case of a singleton input, i.e. assuming a set of fuzzy sets that partitions the input space completely, and the membership functions are ordered, consistent, and convex, and that these functions are normal PTS functions that share the common fuzziness as defined in definition 6.

We further assume that there are two or more input variables that are independent of each other, i.e. the outputs are mapped onto a simple multivariate polynomial (definition 7). By Lemma 3, we know that we can always decompose the multivariate polynomial into univariate polynomial such that each of these polynomials is in an input variable. We can then form a set of fuzzy rules for each of these univariate polynomials. The sum of the outputs of the sets of these fuzzy rules for each set of inputs yield a point on the original simple multivariate polynomial.

Often we want to think the total output in terms of a normalized output within an interval say, \([0, 1]\), we may weight the contribution of each term such that the total contributions fall within \([0, 1]\) as given by the equation:

\[ y = \beta_1 y_1^* + \beta_2 y_2^* + ... + \beta_n y_n^* \]  

(8)

where \( \sum_{j=1}^{n} \beta_j = 1 \).

\( \beta \) is the weighting of the contribution of each term. In many practical situations, the contributions can be equal, or some terms may have a higher contribution at the expense of others.

An example of the process involving two input variables is shown in Figure 6. As can be seen, the output as in eq. (8) is reduced to \( y = \beta_1 y_1^* + \beta_2 y_2^* \).

4. FUZZY KNOWLEDGE MAP

Fuzzy Knowledge Maps [6, 7] (or simply FKM) are directed graphs consisting of circles or nodes to represent concepts or variables, and directed arcs or edges linking one node to another to represent the relation between these nodes. The direction of an arc indicates the direction of influence or inference from the antecedent node to the consequent node. Inference is implemented in an arc as a set of single-antecedent fuzzy rules. The rules can be derived from experts’ knowledge of the problem domain or from data.

Figure 7 shows an example of an FKM model. In the example, the output from Node \( X_1 \) has a fuzzy inference on Node \( X_4 \), which in turn has an inference on Node \( X_3 \), and \( X_4 \) has an inference on \( X_5 \). Each of the arcs consists of a set of one or more fuzzy rules. These rules are single-antecedent rules. In the example FKM, the output of the rules between \( X_1 \) and \( X_3 \), and that between \( X_4 \) and \( X_5 \) are aggregated to form the input to the Node \( X_5 \).
for the consequent node is to derive a single output value of these outputs is subjected to a defuzzification output of the relation exists. Let fuzzy output subset figure, the output of the fuzzy rules is the entire with state antecedent node with state rules used to model the relationship between an antecedent nodes, and simulation of the FKMs, by way of an experiment.

Figure 8 shows a set of single-antecedent fuzzy rules used to model the relationship between an antecedent node with state \( a_i \) and a consequent node with state \( d_i \), between which a relationship \( R(x, y) \) exists. Let \( x \in A \) be the input and \( y \in B \) be the output of the relation \( R(x, y) \). Then, as shown in the figure, the output of the fuzzy rules is the entire fuzzy output subset \( b = \{ b_1, \ldots, b_n \} \). The aggregate of these outputs is subjected to a defuzzification process. The purpose of the defuzzification process is to derive a single output value \( d_i \) in the interval \([0, 1]\) for the consequent node \( Y \). The output \( d_i \) represents the state of node \( Y \) in response to the state \( a_i \) of antecedent node \( X \).

In the case where only a single antecedent node is linked to a consequent node, the output \( d_i \) becomes the new state of the consequent node. However, when two or more antecedent nodes are linked to a consequent node, the outputs \( d_i \) from all of the antecedent nodes are aggregated to form the new state of the consequent node, as given in eq. (8).

In the next section, we describe a methodology for identification of the input variables, extraction of the fuzzy rules between antecedent and consequent nodes, and simulation of the FKMs, by way of an experiment.

5. SIMULATION OF HUMAN OPERATION AT A CHEMICAL PLANT

Consider a human operation of a chemical plant as illustrated by [13], where the plant produces a polymer by polymerisation of certain monomers. The control of the start-up of the plant is by a human. There are five sets of inputs – monomer concentration, change of monomer concentration, monomer flow rate, and two temperature readings inside the plant (denoted as \( x_1, x_2, x_3, x_4 \) and \( x_5 \)). We treat these as the five possible input variables, and the control of the set point for monomer flow rate as the single output variable \( y \). There are 70 data points from the actual plant operation as given in [13].

First we constructed FKMs, each consisting of only one of the input variables, \( x_i \), and an output variable \( y \), and an arc linking the input node to the output node. The arc represents a set of fuzzy rules which maps the input to the output spaces. Figure 9 shows an example of the FKM with a single input and a single output.

Table 1 shows the cluster centres of the input and output spaces. The column Step #1 corresponds to the FKMs with single input and single output. Step #2 and Step #3 will be explained later in this section. From these clusters, we constructed the membership functions and the fuzzy rules to map the input to the output spaces.

Table 1: Cluster centres of the input / output spaces of the operation of the Chemical Plant.
Table 2: Regularity criteria of simulation of FKM with single input, two inputs and three inputs. The SY column indicates the RC using the SY method.

<table>
<thead>
<tr>
<th>Input variable</th>
<th>Step #1</th>
<th>Step #2</th>
<th>Step #3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>606,029</td>
<td>600,715</td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td>6,215,378</td>
<td>6,077,539</td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td>41,053</td>
<td>60,756</td>
<td></td>
</tr>
<tr>
<td>$x_4$</td>
<td>11,828,720</td>
<td>6,663,660</td>
<td></td>
</tr>
<tr>
<td>$x_5$</td>
<td>8,329,532</td>
<td>5,570,199</td>
<td></td>
</tr>
<tr>
<td>$x_1 + x_2$</td>
<td>22,431</td>
<td>46,178</td>
<td></td>
</tr>
<tr>
<td>$x_1 + x_3$</td>
<td>39,692</td>
<td>41,418</td>
<td></td>
</tr>
<tr>
<td>$x_1 + x_4$</td>
<td>50,586</td>
<td>60,124</td>
<td></td>
</tr>
<tr>
<td>$x_1 + x_5$</td>
<td>68,698</td>
<td>60,277</td>
<td></td>
</tr>
<tr>
<td>$x_1 + x_2 + x_3$</td>
<td>32,690</td>
<td>36,950</td>
<td></td>
</tr>
<tr>
<td>$x_1 + x_2 + x_4$</td>
<td>52,720</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>$x_1 + x_2 + x_5$</td>
<td>36,199</td>
<td>41,846</td>
<td></td>
</tr>
</tbody>
</table>

From the outputs of the FKM, we then computed the regularity criterion (RC) as given in eq.(9). The results are as shown in Table 2. As can be seen, input variable $x_2$ has the smallest RC and is therefore considered as the winner. The winning input variable is then selected for the next step in the iteration, i.e. Step #2.

In Step #2, one of those variables that were lost to the winning variable $x_1$ in Step #1 was added to the winning input variable to form a set of dual inputs single output (DISO) FKM models. The process of clustering and evaluation is repeated.

Table 1 Step #2 shows the new cluster centres and Table 2 Step #2 shows the RCs of the results of FKM simulations. As can be seen, input variables $x_2$ and $x_1$ form the best combination, having the lowest RC of 22,431 and it is therefore considered as the winning combination. This is repeated for Step #3. However, as can be seen in Table 2, the smallest RC corresponds to $x_3 + x_1 + x_2$ combination, but the RC is larger than that at Step #2. Therefore this winning combination is discarded and the process of identifying parameters is terminated. The FKM model for the chemical plant is a DISO model, as shown in Figure 10.

As can be seen from Table 2, the final RC is 22,431, which is about 40% less than that using SY method. Our FKM model therefore simulates more closely to the actual output.

In FKM, the rules are single-antecedent fuzzy rules. As such, we avoid rule transparency problem. This is apparent from the rules used in the SY method and FKM.

The fuzzy rules of the SY method are as follows:

1) If $x_1$ is more or less BIG, and $x_2$ is not INCREASED, and $x_1$ is SMALL, then $y$ is SMALL or MEDIUM SMALL.

2) If $x_1$ is more or less MEDIUM, and $x_2$ is DECREASED, and $x_1$ is SMALL or MEDIUM SMALL, then $y$ is MEDIUM SMALL.

3) If $x_1$ is MEDIUM, and $x_2$ shows NO CHANGE, and $x_3$ is MEDIUM SMALL or MEDIUM, then $y$ is MEDIUM.

4) If $x_1$ is more or less MEDIUM, and $x_2$ is ANY VALUE, and $x_3$ is MEDIUM, then $y$ is MEDIUM or MEDIUM BIG.

5) If $x_1$ is more or less SMALL, and $x_2$ is very INCREASED, and $x_3$ is MEDIUM BIG, then $y$ is HIGH.

6) If $x_1$ is more or less SMALL, and $x_2$ is some of INCREASED, and $x_3$ is BIG, then $y$ is very BIG.

The rules used in FKM are as follows:

1) If $x_3$ is LOW, then $y_{3i}$ is LOW

2) If $x_3$ is MEDIUM, then $y_{3i}$ is MEDIUM

3) If $x_3$ is HIGH, then $y_{3i}$ is HIGH

4) If $x_3$ is LOW, then $y_{3i}$ is HIGH

5) If $x_3$ is MEDIUM, then $y_{3i}$ is MEDIUM

6) If $x_3$ is HIGH, then $y_{3i}$ is LOW

The first three rules are implemented in the arc between node flow rate and node control set point,
and the next three rules between monomer concentration and control set point. As can be seen in Figure 10, \( y_x \) and \( y_{x1} \) in the above rules are the outputs from \( x_3 \) and \( x_1 \) respectively. These outputs are aggregated to form the final output \( y \). As can be seen, the above rules are easy to understand, especially in conjunction with the FKM in Figure 10.

6. CONCLUSION

The conventional linguistic conjunctive canonical form of fuzzy rules has been known to cause transparency and complexity problems in fuzzy systems.

We have shown that where the input and output variables are terms of a simple multivariate polynomial, single-antecedent fuzzy rules can be implemented instead of multi-antecedent rules. This eliminates the rule explosion problem and significantly simplifies the logic reasoning processes.

The simulation experiment clearly shows our approach to rule construction is simpler and enhances significant performance improvement over the conventional fuzzy rules construction where the multi-antecedent parts are large and the number of rules is increasing, our approach has a clear advantage in terms of computational overheads and maintenance. In a FKM environment, each set of fuzzy rules relates each input variable to the output, thus eliminating any ambiguity.

REFERENCE