Proceedings of the 2004 IEEE
Conference on Cybernetics and Intelligent Systems
Singapore, 1-3 December, 2004

Fuzzy modelling using a simplified rule base

Sebastian W. Khor, M. Shamim Khan, Chun Che Fung
School of Information Technology
Murdoch University
Perth, WA 6150, Australia
{s.khor; s.khan; l.fung}@murdoch.edu.au

Abstract— **Transparency and complexity are two major concerns of fuzzy rule-based systems. To improve accuracy and precision of the outputs, we need to increase the partitioning of the input space. However, this increases the number of rules exponentially, thereby increasing the complexity of the system and decreasing its transparency. The main factor behind these two issues is the conjunctive canonical form of the fuzzy rules. We present a novel method for replacing these rules with their singleton forms, and using aggregation operators to provide the mechanism for combining the crisp outputs.**

Keywords— Fuzzy constraint aggregation operator, fuzzy modelling, singleton fuzzy rule.

I. Introduction

Until recently, it has generally been perceived that systems implementing fuzzy modelling [1] provide better transparency than the black box models such as artificial neural networks. By transparency, we mean interpretability of rules and outputs. The main reason for this perception is that most conventional rule-based fuzzy systems are abstracted from human experts or heuristics, and they are usually easy to comprehend. This provides the transparency enabling one to gain insights into the system and acquire important knowledge. However, as more and more fuzzy rules are automatically generated using training or experimental data, fuzzy modelling becomes less easily understood by humans because of the increase in complexity of these rules. Another issue is that the number of inputs has to be kept low because the dimension of the input space and complexity grows exponentially in terms of the number of input variables [2, 3]. So we have a dilemma: On one hand, the requirement of accuracy calls for the use of dense rule bases with large numbers of antecedent variables and linguistic terms, on the other hand, exponential growth in the size of rule base creates problems with computational time and storage space requirements. Reduction of rules is desirable but limiting the number of rules may destroy the property of the model as a universal approximator [4].

Many techniques have been proposed for rule reduction. We discuss briefly the problems associated with using these in Section II. In Section III we propose a new approach that aims to improve the interpretation of the much simpler fuzzy rules and reduce the complexity of the problem domain, using some examples to illustrate our approach. The paper concludes in Section IV.

II. Traditional Fuzzy Modelling Methods

Conventional fuzzy modelling involves the transformation (or mapping) of the input space to the output space via a linguistic conjunctive canonical form of fuzzy rules like:

R_i: If x_1 is A_1 and x_2 is A_2 and … and x_k is A_k then y is B (1)

where x_i, $i=1,\dots,k$ is the i^{th} input to the fuzzy system, which is defined on the universe of discourse U_i; A_i is a fuzzy set on U_i; y is the system output defined on a universe of discourse V, and B is a fuzzy set on V. For simplicity we are assuming B in this case to be on a one-dimensional output space. We discuss a multi-dimensional output space in the next section.

To derive the fuzzy if-then rules, several approaches have been proposed. Some of the common approaches are outlined in the following subsections:

A. Grid Partitioning

The most common method to construct fuzzy rules is to partition the input space into a specified number of membership functions in the form of a lattice or grid. The fuzzy sets of these functions are convex normal fuzzy sets in the sense that their α-cuts are connected and their core not empty. i.e. $\forall\alpha$: ${}_\alpha A = \{x: \mu(x) \geq \alpha\}$, $A_i \neq \emptyset$, $i = 1, \dots, n$ where n is the number of fuzzy sets in a variable. The rule base is then constructed to cover the antecedent space by using logical combinations of the antecedent terms. For the conjunctive form of the rules' antecedents, the number of rules k needed to cover the antecedent domain is $k=p^n$, where p is the number of partitions in the input space. A clear drawback of this approach is therefore that the number of rules, k, in the model grows exponentially with the partitioned input space p and the number of antecedent variables n. This is an actively researched area, the main aim of which being basically to reduce the number of rules, and to optimize the input space. Some of the approaches include evolutionary computing techniques [5], hierarchical construction of rules [6], and identification of redundant input variables [7]. Other forms of combining rule antecedents have also been attempted, including disjunctive form, but with arguable success [8, 9].

B. Tree Partitioning

Tree partitioning is a method that eliminates the problems associated with grid-partitioning. It can be used to build the decision rules based on a hierarchical structure e.g. fuzzy

0-7803-8643-4/04/$20.00 © 2004 IEEE

decision trees [10] or quad trees [11]. Although these methods curtail the explosion of rules, they tend to have a lot of non-uniform overlapping membership functions to which it is hard to assign an understandable linguistic term. In addition, it is a common practice in fuzzy expert systems to update the fuzzy rules that are abstracted from experts using different learning methods in order to improve their performance. This can also lead to the loss of interpretability of the fuzzy model.

C. *Scatter Partitioning*

The scatter partitioning method [12, 13] allows the IF-parts of the fuzzy rules to be positioned at arbitrary locations in the input space. The Gaussian membership function (often used in scatter partitioning) covers across several partitions and only in the subset of the input space where data exist. Thus, assigning meaningful linguistic labels to these functions is difficult if not impossible.

D. *Compact Fuzzy Rule Bases*

An important objective of rule reduction is to produce a compact fuzzy rule base. Some of these approaches are:

- Irrelevant feature elimination algorithm [14];
- rule selection based on evolutionary algorithms [15, 16]; and
- Fuzzy feature clustering [17].

Generally transparency and complexity still remain an issue in these rule bases. Yubazaki and Hirota [18] report the successful application of a Single Input Rule Modules (SIRMs) dynamically connected model as a controller in inverted pendulum simulations. However, its success depends on the ability of the system to define the importance degree of each input variable and to prioritize these importance degrees.

In conclusion, there has been little progress towards resolving the complexity and transparency issues associated with fuzzy rules. Mendel and Liang [19] are of the opinion that replacing the multi-antecedent rules will be a significant contribution towards eliminating the associated rule explosion. We observe that it will also resolve the complexity and transparency problems.

III. A Simpler Approach to Mapping

A review of the literature shows that all these approaches do not consider the dependence/independence between the input variables, thus side-stepping the possibility of a much simpler approach to addressing the issues raised earlier.

Consider a hypothetical example of tipping of a restaurant dinner. We want to give tips in proportion to the two input variables *service* and *food*. Let's partition each of the two input variables into three regions – poor, good and excellent for *service*, and rancid, good and delicious for *food*. Without reducing the number of rules, either heuristically or otherwise, if we are to apply the conventional grid partitioning method to the input domain, we need $3^2 = 9$ fuzzy rules, e.g.:

```
Rule 1: If service is poor and food is rancid
then tip is cheap
Rule 2: If service is poor and food is good then
tip is better_than_cheap
...
...
Rule 9: If service is excellent and food is
delicious then tip is generous
```

As can be seen from the rules presented above, the interpretability of the rules will be significantly degraded as the number of rules increases. When multiple memberships are ANDed to form a rule, its transparency is reduced as the dimension of the input space is proportional to the number of input variables. Fig. 1 below shows the grid partitions of the relevant input space. Since there are two input variables, we have a 2-dimensional input space. By the same token, if there are *n* input variables, there will be an *n*-dimensional space.

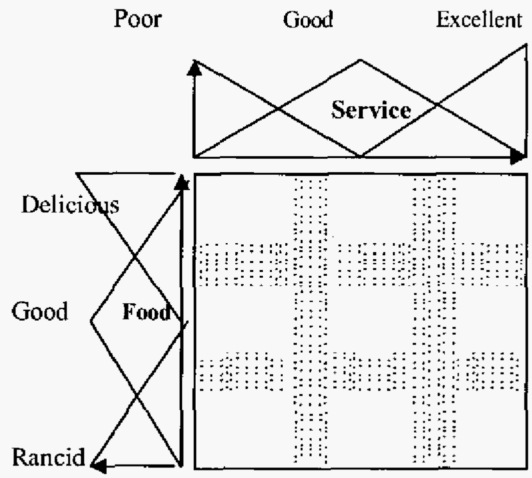

Fig. 1. Partitions of the input space of tipping example

A. *Divide and Conquer*

The conventional conjunctive form of if-then rule assumes that both input variables (in this case, the quality of *service* and *food*) must be present in order to contribute towards the output *tip*. The fact is that *service* and *food* are independent variables and there is no inter-relationship between these two. Fig. 2 shows a graph created by plotting curves of *tips* for *food* with *services* kept constant at 0, 0.6 and 1. As can be seen, a change in *services* causes a "shift" in the curve for *food*. This is because the input variables are independent of each other. In the real world we encounter many such instances. For example, in economics, it is well known that a change in consumers' taste preferences causes a shift in the demand for goods.

Since the contribution of one variable towards *tip* does not affect the contribution of the other, we can treat the two input variables *service* and *food* separately, thus effectively splitting the input space into two one-dimensional spaces. We can then have a set of rules for each of the input space. The two sets of fuzzy rules are as follows:

For *service* input variable:

```
Rule 1: If service is poor then tip is cheap
Rule 2: If service is good then tip is average
Rule 3: If service is excellent then tip is
```

```
generous
```

For *food* input variable:

```
Rule 4: If food is rancid then tip is cheap
Rule 5: If food is good then tip is average
Rule 6: If food is delicious then tip is generous
```

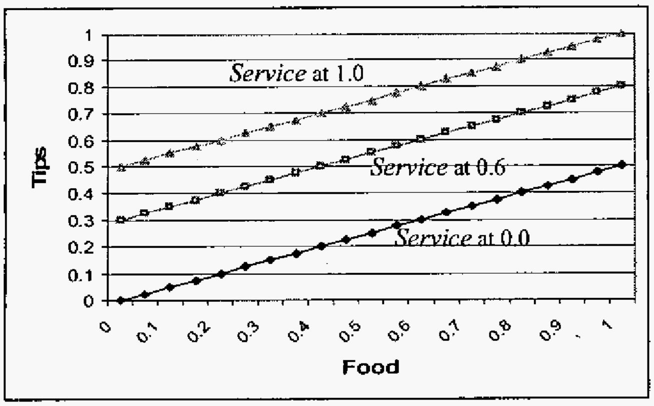

Fig. 2. Graph of *tips* for *food* with *services* kept constant at 0, 0.6, and 1

As can be seen from the sets of fuzzy rules, even without further techniques to reduce the fuzzy rules, the number of rules required is reduced, the linguistic clarity is maintained, and the rules are simple (i.e. singleton) if-then rules. Fig. 3 shows the process of mapping the input variables to the output by the respective set of fuzzy rules. For each input variable, the input is fuzzified. Each of the fuzzy rules is fired to a certain degree depending upon the degree of membership. The output is then defuzzified. As can be seen, the input is crisp and the output is also crisp. This allows us to perform some simple combination of the defuzzified outputs.

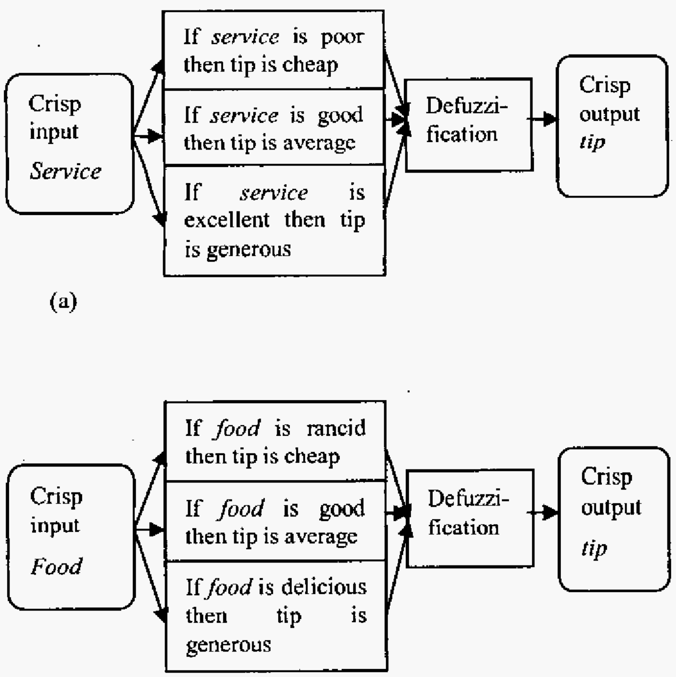

Fig.3. Application of appropriate sets of fuzzy rules to individual input variables (a) *service* and (b) *food*.

B. *Constraint Aggregation Operator*

Although the two input variables are independent, their contribution towards the output may vary. For example, we have the rule "If service is excellent then tip is generous". That does not translate to a maximum *tip* output. We say that the output *tip* has been constrained. *Service* is only part of the contributions towards *tip*. We also need to consider the contribution from *food*. This can be reasoned as follows:

Suppose we have a traditional if-then rule with *n* IF-parts as in (1) in Section II. If an IF-part is true to a certain degree, then the rule is considered to be true to a certain degree. However, the total contributions of truth of all the IF-parts must be within the maximum truth the set of fuzzy rules can have, which is 1. Therefore the defuzzified outputs from the sets of singleton fuzzy rules have to be scaled and combined to give an aggregated output for the whole system. This process is performed by the *constraint aggregation operator*[20]. Fig. 4 shows a diagram of the constraint and combinatorial operation. The defuzzified outputs from X_1 and X_2 are passed through the *constraint aggregation operator* which scales the outputs such that the final aggregated output Y_1 is always in the interval [0, 1]. This is explained shortly.

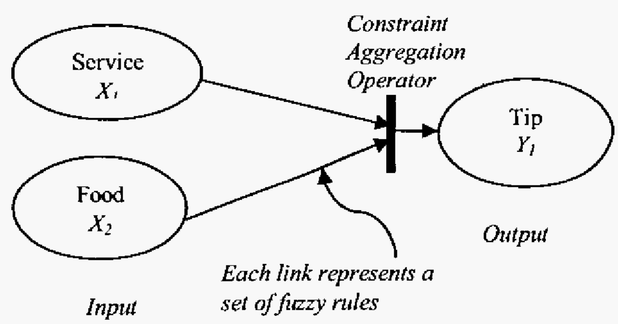

Fig. 4. Mapping of input variables *service* and *food* to output *tip* via the *constraint aggregation operator*

The *constraint aggregation operator* (CAO) serves two purposes:

1. *Constraint weighting.* The idea of weighting the fuzzy rules or input variables is not new. For example, weighting was applied to fuzzy rules to allow for importance weight factors designated to the different rules in a fuzzy expert system in [21]. Weighting was used in uncertainty reasoning to determine the fuzzy truth values of conditions in a fuzzy Petri net [22]. As far as we know, the weighting in all the cases in the literature were applied to the antecedents, whereas we apply the weighting to the consequents. Our *constraint aggregation operator* is

based on the premise that the total output from various input variables should be within certain predefined limit, usually in the interval [0, 1]. This is not unreasonable as in the conventional fuzzy model, if the maximum output is 1, then the final combined value of the output (as in the tipping example) should also lie within the interval [0, 1], where 0 denotes minimum value and 1 denotes the maximum value. Thus, the CAO allows one to design the fuzzy rules in two modes:

- *Local View Mode*: When mapping an input variable in a set of multiple input variables to the output space, we are in the local view mode. We consider only a single input (as if all other input variables remain constant) and the defuzzified output in the interval [0, 1].
- *Global View Mode*: When considering the complete set of defuzzified crisp outputs from all the multiple input variables, we are in the global view mode. Thus the defuzzified outputs from the individual input variables have to be scaled to match their actual contribution to the output. The scaling is performed by CAO.

2. *Importance of Contribution*. The importance of the contributions of inputs towards the output decides the weighting of CAO. In normal circumstances, all inputs are expected to contribute equally towards the output. However, there are occasions when emphasis may be placed on the contribution of certain input variable (factor) such as management policy, market factors, etc. over other input variables.

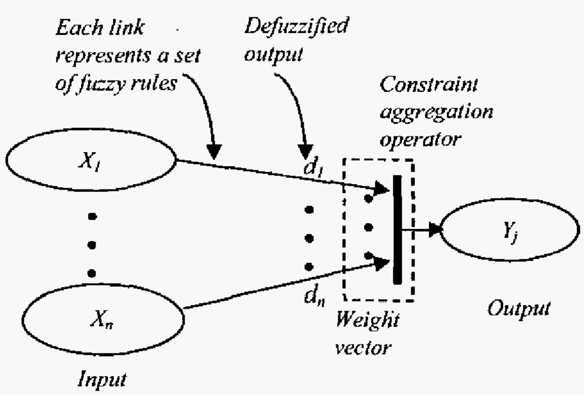

Fig. 5. Multiple input variables X_i mapped to the output variable Y_j via the respective sets of fuzzy rules as indicated by the arrow-headed links. The defuzzified outputs are subjected to the *constraint aggregation operator* for scaling and aggregation

Consider the input variables X_i, $i = 1, \ldots n$, in relation to a dependent output variable Y_j, together with associated weights w_{ij}, as depicted in Fig. 5. The defuzzified outputs d_i, $i = 1, \ldots n$ from the n individual independent variables are scaled and combined using an n-dimensional aggregation operator A, $A: \Re^n \rightarrow \Re$, given by:

$$A(X_i, \ldots, X_n) = \sum_{i=1}^{n} w_i d_i \tag{2}$$

$$\text{where } W = (w_1, w_2, \ldots, w_n)^T, \quad \sum_{i=1}^{n} w_i = 1$$

If there is no specific importance attached to any input variable, a simpler approach is to treat all input variables to be equally important. Thus, the weight for an input variable is given by $w_i = 1/n$, where n is the number of input variables related to the output variable Y_j, $j = 1, \ldots, m$, where m is the number of output variables.

Going back to the tipping example, as we want to treat both the input variables *service* and *food* as equally important, the weight vector for CAO is therefore $(0.5\ 0.5)^T$. Suppose now that the contribution to tips be skewed, say, 75% towards *service* and 25% to *food*. We can easily accomplish this by merely changing the weights parameter in CAO without making any change to the fuzzy rules, or redesigning all the related fuzzy rules as would be the case in a conventional fuzzy rule based system. This is a great advantage in some applications such as decision making, dynamic systems and fuzzy expert systems.

C. Rule Selection Operator

The *constraint aggregation operator* works on the premise that the input variables are independent of each other. However, there are cases where an input variable is dependent of another. By dependence, we mean the degree of the effect of one variable on the output is dependent on the degree of the membership of another variable. We call an input variable that affects another input variable as the auxiliary variable, and the variable being affected as the dependent variable. If we plot a graph of the total output against the dependent variable while holding the auxiliary variable constant at

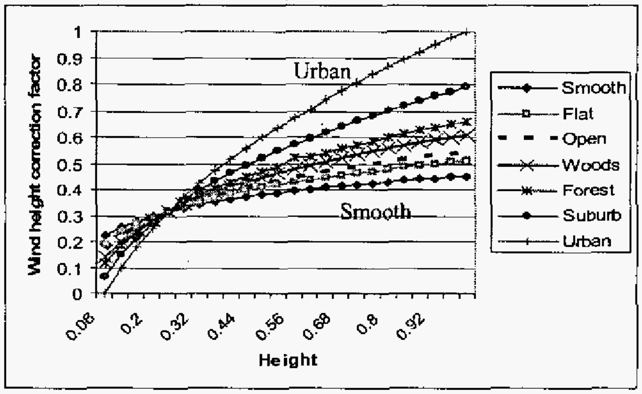

Fig. 6. Wind height correction factors against wind heights for various ground elements

various values in the universe of discourse, we may have a graph similar to Fig. 6. The graph is taken from our experiment described in [23]. The *wind height correction factor*[1] (and therefore wind speed) changes disproportionately for different types of ground elements (i.e. ground surface features) against changes in the heights of wind above ground level. As can be seen, the behaviour of the curves is different from that in Fig. 2.

In this paper, we will only consider the case of a single auxiliary variable, using our wind speed experiment [23] as an example. The case of multiple auxiliary variables will be discussed in a future paper.

To adequately cover the complete 2-dimensional input space ($D \times A$) between the dependent input variable D and auxiliary variable A, we need multiple sets of fuzzy rules to map input space to output space. However, for an input dataset, only one set of the fuzzy rules will be fired with maximum output, corresponding to the auxiliary input variable

G Element / Height	Smooth	Flat	Open	Woods	Forest	Suburb	Urban
0.08	0.22	0.19	0.17	0.14	0.12	0.07	0.00
0.12	0.26	0.24	0.22	0.20	0.19	0.15	0.10
0.16	0.28	0.27	0.26	0.25	0.24	0.21	0.18
0.20	0.30	0.29	0.29	0.28	0.28	0.27	0.25
0.24	0.32	0.32	0.32	0.32	0.32	0.32	0.32
0.28	0.33	0.34	0.34	0.34	0.35	0.36	0.37
0.32	0.34	0.35	0.36	0.37	0.38	0.40	0.42
0.36	0.35	0.37	0.38	0.39	0.41	0.43	0.47
0.40	0.36	0.38	0.39	0.41	0.43	0.46	0.52
0.44	0.37	0.40	0.41	0.43	0.45	0.49	0.56
0.48	0.38	0.41	0.42	0.45	0.47	0.52	0.60
0.52	0.39	0.42	0.43	0.47	0.49	0.55	0.64
0.56	0.40	0.43	0.45	0.48	0.52	0.57	0.68
0.60	0.40	0.44	0.46	0.50	0.52	0.60	0.71
0.64	0.41	0.45	0.47	0.51	0.54	0.62	0.74
0.68	0.41	0.46	0.48	0.52	0.56	0.64	0.78
0.72	0.42	0.46	0.49	0.53	0.57	0.66	0.81
0.76	0.43	0.47	0.50	0.55	0.59	0.68	0.84
0.80	0.43	0.48	0.51	0.56	0.60	0.70	0.87
0.84	0.44	0.49	0.51	0.57	0.61	0.72	0.89
0.88	0.44	0.49	0.52	0.58	0.63	0.74	0.92
0.92	0.44	0.50	0.53	0.59	0.64	0.76	0.95
0.96	0.45	0.51	0.54	0.60	0.65	0.77	0.98
1.00	0.45	0.51	0.55	0.61	0.66	0.79	1.00

Fig. 7. Partitioning of the wind height correction factor in the 2D input space (*wind heights* × *ground elements*) corresponding to the seven *ground elements:* Smooth, Flat, Open, Woods, Forest, Suburb, and Urban

[1] *Wind height correction factor* is the wind shear factor caused by wind turbulences due to ground surfaces. Interested readers are referred to [24 25].

in the universe of discourse. For example, in our previous example in Fig. 6, if we have the auxiliary variable *ground element* corresponding to *Urban,* then the set of fuzzy rules selected for firing will correspond to the curve *Urban*. In another words, membership functions can be constructed with the universe of discourse along dependent variable D to form a fuzzy subset for each partition in the auxiliary variable A. Fig. 7 shows the partitioning of the *wind height correction factor* in the 2-D input space (*height* x *ground elements*) along the seven sets of *ground elements*. Seven sets of fuzzy rules are then constructed corresponding to each of the *ground elements.*

Instead of the *constraint aggregation operator,* we use a *rule selection operator (RSO)* to select the appropriate set of fuzzy rules for the transformation from the input space to the output space. Fig. 8 shows a topology using RSO to map a 2-D input space onto the output space for the example model under consideration. RSO is a function that takes two parameters – one dependent input variable and one auxiliary input variable. It selects a set of the fuzzy rules, based on the maximum membership grades of *ground elements* of the input dataset. The input *height* is transformed to the output by applying the selected set of fuzzy rules, and the output is defuzzified to arrive at a crisp output for the correction factor.

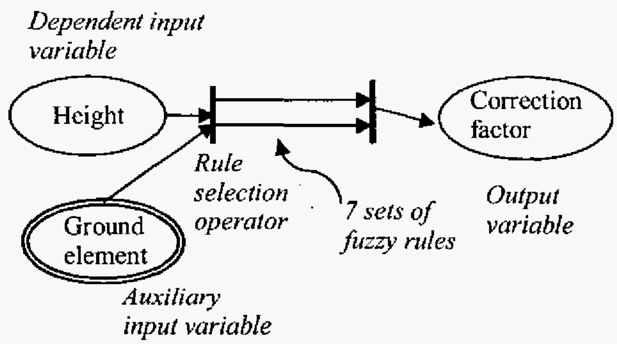

Fig. 8. Input variable *height* is mapped to the output via a set of fuzzy rules which is selected by the *rule selection operator* by selecting the maximum membership grades of the *ground element.*

IV. CONCLUSION

The conventional linguistic conjunctive canonical form of fuzzy rules has been known to cause transparency and complexity problems in fuzzy systems. The literature shows that so far there has been little progress to overcome the concerns despite considerable research interest and numerous attempts.

The method discussed in this paper can replace these multi-IF-part rules by a set of singleton fuzzy rules and an operator. This is a significant proposition because it eliminates the rule explosion problem associated with the conjunctive canonical form of fuzzy rules and significantly simplifies the logic reasoning processes.

In our approach, each IF-part of fuzzy rules is evaluated only once per input variable per input datum, as compared with multiple evaluations of the IF-parts in the conventional rule base. However, defuzzification is required for each input, as compared with only once in the conventional system. When the number of IF-parts is large and the number of rules increases, our approach has a clear advantage in terms of computational overheads, and transparency and complexity of rules. Furthermore, maintenance of rule bases are greatly simplified because of the simplicity associated with the singleton rules.

REFERENCES

1 L. A. Zadeh, "Fuzzy Algorithm," Information and Control, vol. 12, pp. 94-102, 1968.

2 L. T. Kóczy and K. Hirota, "Size reduction by interpolation in fuzzy rule bases," IEEE Transaction on Systems, Man and Cybernetics, vol. 27, pp. 14-25, 1997.

3 P. Barayi, Y. Yam, P. Várlaki, and P. Michelberger, "Singular value decomposition of linguistically defined relations," Int. J. of Fuzzy systems, vol. 2, pp. 108-116, 2000.

4 D. Tikk, P. Baranyi, and R. J. Patton, "Polytopic and TS models are nowhere dense in the approximation model space," presented at IEEE Int. Conf. on Systems, Man and Cybernetics (SMC 2002), Hammamet, Tunisia, USA, pp. 150-153, 2002.

5 C. A. Peña-Reyes and M. Sipper, "Fuzzy CoCo: A Cooperative-Coevolutionary Approach to Fuzzy Modeling," IEEE Trans. on Fuzzy Systems, vol. 9, pp. 727-737, 2001.

6 R. R. Yager, "Alternative structures for knowledge representation in fuzzy logic controllers," in Fuzzy control systems, G. L. Abraham Kandel, Ed. Florida, USA: Boca Raton: CRC Press, 1994, pp. 99–137.

7 T. Slawinski, U. Schwane, J. Praczyk, A. Krone, H. Jessen, H. Kiendl, and D. Lieske, "Application of WINROSA for Controller Adaptation in Robotics and Classification in Quality Control," Reihe Computational Intelligence, Faculty of Electrical Engineering, University of Dortmund, Denmark 1998.

8 W. E. Combs and J. E. Andrews, "Combinatorial Rule Explosion Eliminated by a Fuzzy Rule Configuration," IEEE Trans. on Fuzzy Systems, vol. 6, pp. 1-11, 1998.

9 S. Dick and A. Kandel, "Comments on "Combinatorial Rule Explosion Eliminated by a Fuzzy Rule Configuration"," IEEE Trans. on Fuzzy Systems, vol. 7, pp. 475-477, 1999.

10 C. Z. Janikow and M. Fajfer, "Fuzzy Partitioning with FID3.1," presented at 18th Int Conf of the North American Fuzzy Information Processing Society (NAFIPS), New York, pp. 467-471, 1999.

11 C.-T. Sun, "Rule-base structure identification in an adaptive-network-based inference systems," IEEE Trans. Fuzzy Systems, vol. 2, pp. 64–73, 1994.

12 S.-Y. Ho, H.-M. Chen, S.-J. Ho, and T.-K. Chen, "Design of Accurate Classifiers With a Compact Fuzzy-Rule Base Using an Evolutionary Scatter Partition of Feature Space," IEEE Trans. on Systems, Man, and Cybernetics, vol. 34, pp. 1031-1044, 2004.

13 B. Fritzke, "Incremental neuro-fuzzy systems," presented at Int Symposium on Optical Science, Engineering and Instrumentation: Application of Soft Computing (SPIE), San diego, pp. (Invited paper), 1997.

14 R. Thawonmas and S. Abe, "A Novel Approach to Feature Selection Based on Analysis of Class Regions," IEEE Trans. on Systems, Man, and Cybernetics, B, vol. 27, pp. 196-207, 1997.

15 Y. Jin, W. v. Seelen, and B. Sendhoff, "On Generating FC3 Fuzzy Rule Systems from Data Using Evolution Strategies," IEEE Trans. on Systems, Man, and Cybernetics, B, vol. 29, pp. 829-845, 1999.

16 W. Pedrycz and M. Reformat, "Evolutionary Fuzzy Modeling," IEEE Trans. Fuzzy Systems, vol. 11, pp. 652-665, 2003.

17 H. Roubos and M. Setnes, "Compact and Transparent Fuzzy Models and Classifiers Through Iterative Complexity Reduction," IEEE Trans. Fuzzy Systems, vol. 9, pp. 516-524, 2001.

18 J. Yi, N. Yubazaki, and K. Hirota, "A proposal of SIRMs dynamically connected fuzzy inference model for plural input fuzzy control," Fuzzy Sets and Systems, vol. 125, pp. 79–92, 2002.

19 J. M. Mendel and Q. Liang, "Comments on "Combinatorial Rule Explosion Eliminated by a Fuzzy Rule Configuration"," IEEE Trans. on Fuzzy Systems, vol. 7, pp. 369-371, 1999.

20 M. S. Khan and S. W. Khor, "A Framework for Fuzzy Rule-based Cognitive Maps," presented at 8th Pacific Rim Int. Conf. on Artificial Intelligence (PRICAI 2004), Auckland, New Zealand, pp. (to be published Aug 2004), 2004.

21 D. S. Yeung and E. C. C. Ysang, "A multilevel weighted fuzzy reasoning algorithm for expert systems," IEEE Trans. on Systems, Man, and Cybernetics, A, vol. 28, pp. 149-158, 1998.

22 S.-M. Chen, "A fuzzy reasoning approach for rule-based systems based on fuzzy logic," IEEE Trans. on Systems, Man, and Cybernetics, B, vol. 26, pp. 769-778, 1996.

23 S. W. Khor, J. Payakpate, M. S. Khan, and C. C. Fung, "A new approach to fuzzy knowledge representation," presented at 2004 Postgraduate Electrical Engineering and Computing Symposium (PEECS 2004), Perth, Australia, pp. (to be published), 2004.

24 C. J. Walcek, "A Gaussian dispersion/plume model explicityly accouting for wind shear," presented at 13th Conf. on the Applications of Air Pollution Meteorolgy with the Air and Waste Management Assoc., Vancouver, Canada, pp. 6.2, 2004.

25 M. Sagrillo, "Site Analysis for Wind Generators," in Home Power, vol. 40, 1994, pp. 86-90.