PERGAMON

Computers and Structures 80 (2002) 629–641

Computers & Structures

www.elsevier.com/locate/compstruc

# A fuzzy system for concrete bridge damage diagnosis

Zhiye Zhao *, Chuanyu Chen

*School of Civil and Environmental Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore*

Received 28 November 2000; accepted 9 January 2002

**Abstract**

The widely used bridge management systems require appropriate preliminary deterioration diagnosis and modeling. This paper presents a fuzzy rule-based inference system for bridge damage diagnosis and prediction which aims to provide bridge designers with valuable information about the impacts of design factors on bridge deterioration. The validity of these influence parameters is verified by an input variable identification method. Fuzzy logic is utilized to handle uncertainties and imprecision involved. A modified mountain clustering method is employed to create the training data set. A fuzzy partitioning algorithm is implemented to construct the membership functions of the input variables and to induce the fuzzy rules from the numerical data. The generated rule base is checked and optimized based on the similarity measures among the input fuzzy sets. Illustrative examples show that the system has a high classification accuracy rate with a small number of rules. 

*Keywords:* Concrete bridge; Damage diagnosis; Fuzzy inference

## 1. Introduction

Bridges are crucial components of highway networks and require both corrective and preventative maintenance activities. In recent years, continuing aging and heavy utilization of many bridges have come into conflict with limited funds available in many countries. Consequently, research interests in this field have been focused on the efficient management of existing structures and the development of bridge management systems (BMSs) to assist the decision makers in establishing efficient repair and maintenance programs. A key to success in BMSs relies heavily upon the reliability of the technique adopted for deterioration diagnosis, which evaluates the defect causes by isolating a set of parameters related with structural and environmental conditions.

The objective of the bridge deterioration diagnosis is to find out the causes of the observed defects, and to evaluate the impacts of these defects on bridge safety, and to propose rehabilitation recommendations. This is a typical ill-defined problem due to the complex nature of structural deterioration and the lack of the accurate information. The many different types of defects and the corresponding possible causes make the deterioration diagnosis a complex task, and require the intuition, judgment and heuristic knowledge of experienced engineers [1]. Therefore, it is very meaningful to build an expert system to help engineers to make appropriate decisions by performing preliminary diagnosis and modeling [2]. Moreover, since heuristic knowledge plays an important role in the process of deterioration diagnosis and modeling, exploiting expert systems to capture the expertise and to simulate the reasoning patterns of experts is a promising direction.

On the other hand, the representations of heuristic knowledge from bridge engineers and the descriptions on the observed defects by bridge inspectors are usually in the form of natural language that contains intrinsic imprecision and uncertainty. Various possibly occurring exceptions may influence bridge engineers' confidence on their decision, and thus result in imprecise statement [3]. Therefore, a reasoning mechanism that can deal with uncertain and imprecise information is expected in the proposed expert systems. This has suggested the

* Corresponding author. Fax: +65-791-0676.
*E-mail address:* czzhao@ntu.edu.sg (Z. Zhao).


PII: S0045-7949(02)00031-7

application of fuzzy sets theory, which is specially defined to analyze the linguistic data within a formal mathematical framework.

Many studies have been conducted in the area of deterioration diagnosis using the fuzzy models. The fuzzy set theory has been employed to link the bridge deck damage with its appearance and the work was further improved by the introduction of generic algorithm in the knowledge acquisition [4]. The fuzzy mapping formalism has been used to estimate the remaining life and soundness of concrete bridges [5]. A fuzzy approach which combined the probability theory and the fuzzy reasoning was employed to assess bridge damage levels and their causes [3].

This paper aims to construct the framework of a fuzzy expert system for diagnosing bridge damages so as to provide bridge designers with valuable information about the impact of design parameters on bridge deterioration. Efficient methods for input variable identification, knowledge generation, and rules optimization are used in developing the system. The generated rules from the system are generally in agreement with the experts' opinion. Illustrative examples show that the system has a high classification accuracy rate (CAR) with a small number of rules.

## 2. Development of the DIASYN system

In this section, the development of the framework of a fuzzy inference system DIAgnosis SYNthesis (DIASYN) is presented. DIASYN is supposed to be a concept demonstration system for providing the bridge maintenance engineers and the bridge design engineers with assistance to obtain preliminary but important knowledge on individual bridge defects.

The DIASYN system incorporates a fuzzy reasoning process containing a rule base with its acquisition and update facility and a fuzzy inference engine with an explanation facility, and a user interface with option selecting capacity. The rules are **if-then** statements that describe associations between fuzzy parameters. Given the required input data, the inference engine evaluates the rules and generates an appropriate conclusion. Users can choose to make diagnoses of new cases or to update the rule base with new training data through the user interface.

A block diagram of the proposed system is shown in Fig. 1. The fuzzy rules provide associations between observed bridge conditions and damage causes. They are created by a rule generation algorithm that can convert crisp training data into fuzzy statements. The training data are collected from bridge inspection records and formalized into standard vectors. In the operational mode, the system reads a state vector of observed bridge condition and the inference engine performs damage cause implication through evaluation of the rules. The output of this implication procedure is a linguistic variable that describes the possible damage cause with a confident degree. This linguistic variable can be defuzzified by the explanation facility if a crisp output is desired. In the updating mode, new training vectors are input to generate new rules together with the existing training data. New rules, if any, will be installed in the rule base before the system gives a prompting of 'updating finished' as output.

This system is implemented through the developments of several Matlab modules, which are discussed in details in the following sections.

### 2.1. Input variable identification module

Concrete bridges deteriorate with many influence factors. In general, the following list of explanatory parameters are considered:

- design factors, i.e. structural type, span length, deck width, number of spans, wearing surface type, skew angle, etc.,
- environmental factors, i.e. humidity and precipitation, climate region, traffic volume, temperature variations, etc.,
- other factors, such as structure age, function class and location of damages.

But it is likely that those statistically less important components of transformed variables may arise from noise, or extraneous flourishes not relevant to the intrinsic nature of bridge deterioration. So their exclusion is necessary. The process of feature extraction prior to object classification (rule generation) can be regarded as parameter recognition, which is supposed to benefit both system behavior modeling and computation complexity reduction [6]. To effectively identify the influencing variables, we first select a certain number of possible candidates by taking a heuristic method based on expertise and common sense knowledge. Seven candidates are selected in our system, namely, defect location, defect condition, structural type, overall length, span number, overall width, and bridge age. Note that our selection may be restricted by the availability of the data source, we shall build the system flexibly so that more factors can be adapted if necessary and possible. Our identification process begins with principal component analysis as a preliminary analysis, and is followed by a neural network method.

### 2.2. Probabilistic neural network and forward selection method

The idea in identifying input variables by neural network is that the significant input variables of the

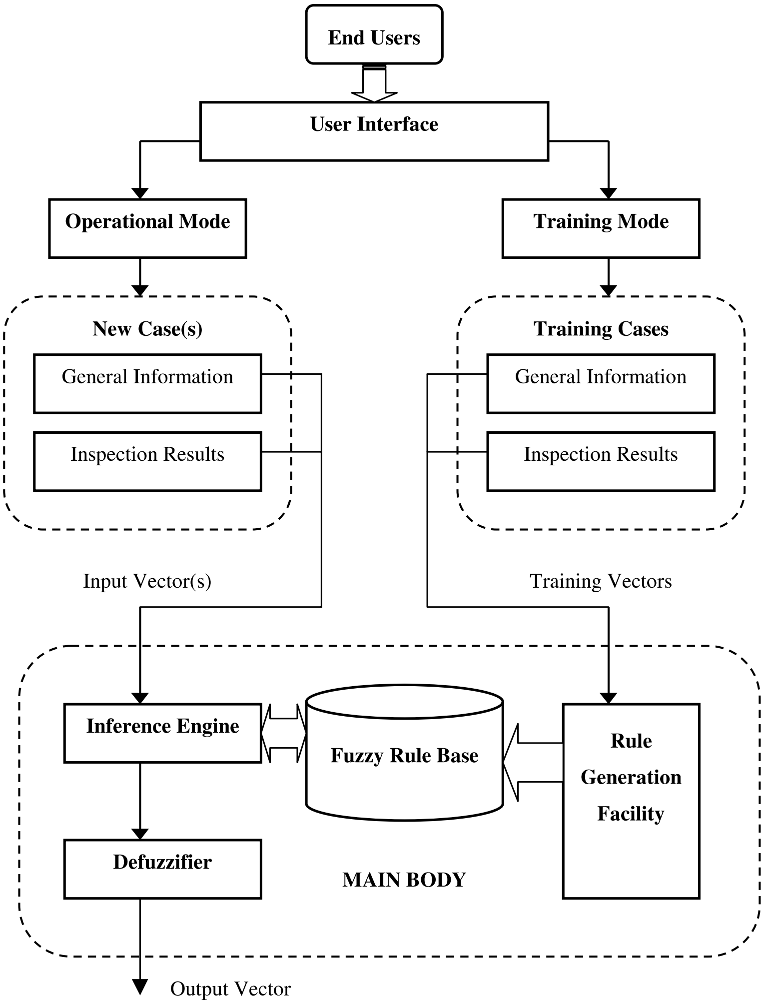


Fig. 1. Basic configuration of the proposed system.

network can definitely influence its performance. We can thus identify the input variables by evaluating their significance to the system. *Probabilistic neural network* (PNN) is a feed-forward neural network that implements the Bayes decision strategy for classifying input patterns, in which the sigmoid activation function commonly used in neural networks is replaced with an exponential function [7]. The PNN is chosen because of the nature of our case: we have seven input candidates, $x_1$–$x_7$, and have to confirm their effects to the output, $y$, which is classified into four categories (see Table 1). This is a typical pattern classification problem.

The basic architecture of a PNN is illustrated in Fig. 2 [7]. The input units distribute their values unchanged to the units of the second layer, which are called pattern units. Each of these pattern units forms a dot product of the input pattern vector $X$ with a weight vector $W_i$, so $Z_i = XW_i$, and then performs a nonlinear operation on $Z_i$ before outputting its activation level to the summation units. The summation units in the third layer sum the outputs of the pattern units belonging to the class from which the training pattern was selected, and in effect generate a posteriori probability distribution function for that class, evaluated at the input point. The output, or decision, units are $c$-input neurons, $c$ is the number of target classes. Generally they compute the decision risks for each class through a risk function that assigns cost to a decision for class $A$ in the case of the input vector $X$ actually belonging to class $B$. The class with minimum risk or with maximum probability would be chosen as output of the net.

In implementation, we should generally count $2^7 - 1 = 127$ cases in our case: seven cases if the system has only one input, 21 cases if it has two inputs, 35 cases if it

Table 1
Representative damage causes

| Damage cause groups | Details of damage groups |
|---|---|
| *Damage cause A* | |
| Loads and the likes | Extreme wheel loads |
| | Traffic crash |
| | Inadequate arrangement of supporting girders |
| *Damage cause B* | |
| Design and structural factors | Short rigidity due to short deck depth |
| | Short quality of main steel bars and distribution bars |
| | Inadequacy of distributed cross beams |
| | Additional bending moment due to differential settlement |
| | Tensile stress due to shrinkage of main girders |
| *Damage cause C* | |
| Construction work | Poor quality of concrete |
| | Inappropriate curing |
| | Inappropriate treatment of construction joint |
| | Short covering |
| | Poor workmanship |
| *Damage cause D* | |
| Other factors | Salt (anti-freezing mixture) |
| | Poor drainage |
| | Action of alkali material |
| | Aging |
| | Chemical attack |

has three inputs, and so on. It is not practical to build and train so many PNN nets. To solve this problem, we construct the PNN nets with a straightforward design in which the network's performance does not depend on training but convergence to a Bayesian classifier is still guaranteed. So that the training of the networks can be omitted and the identification procedure would not be too time consuming. The architecture is shown in Fig. 3 (Matlab manual), the key points here include:

- $IW^{1,1}$ are set to the transpose of the matrix formed from the $Q$ training pairs.
- ||dist|| box produces a vector whose elements indicate how close the input is to the vectors of the training set. These elements are multiplied, element by element, by the bias and then sent to the radbas transfer function.
- An input vector close to a training vector will be represented by a number close to 1 in the output vector $a^1$. If an input is close to several training vectors of a single class, it will be represented by several elements of $a^1$ that are close to 1.
- $LW^{2,1}$ are set to the matrix of the target vectors. Each vector has a 1 only in the row associated with that particular class of input and zeros elsewhere.
- *Compete* transfer function produces a 1 corresponding to the largest element of $n^2$, and zeros elsewhere.

Moreover, we apply a forward selection method proposed by Sugeno and Yasukawa [8]: increase the

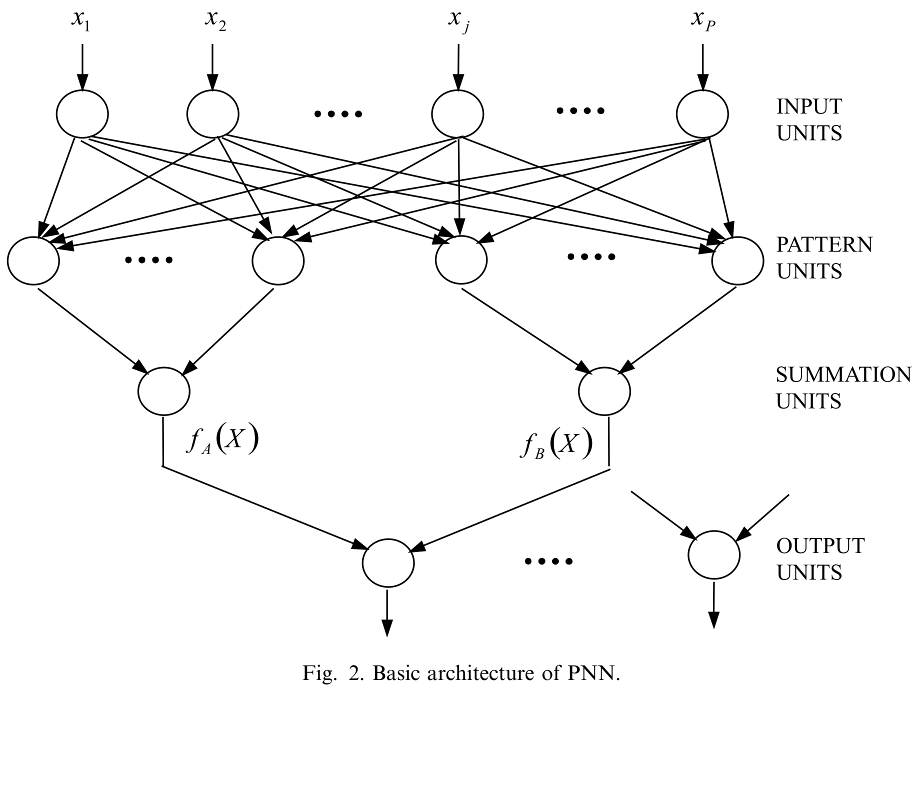


Fig. 2. Basic architecture of PNN.

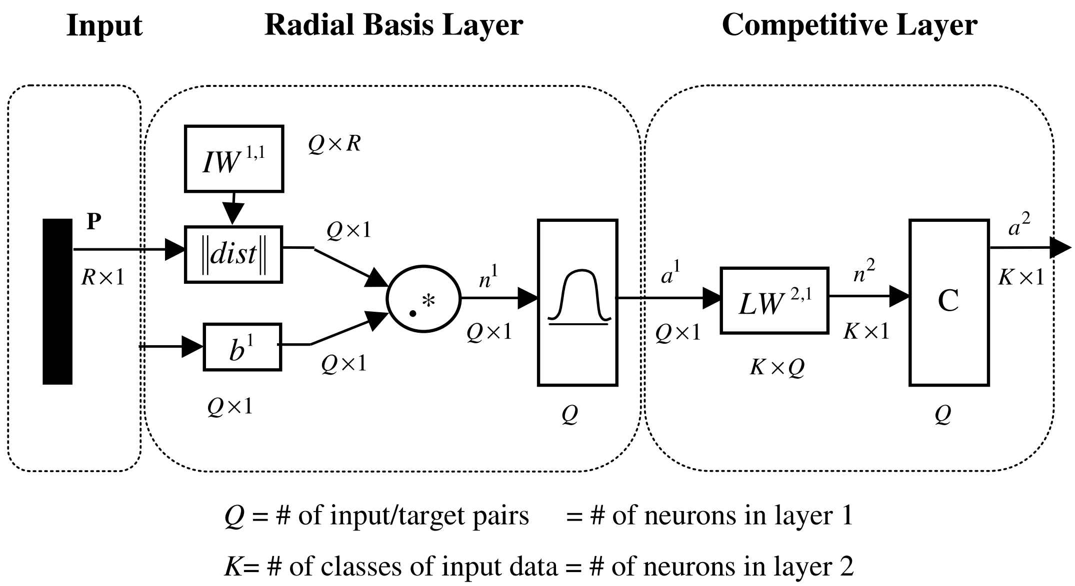


Fig. 3. Architecture of the proposed PNN.

number of inputs one by one and watch the CAR of the PNN nets as a judging criterion. The outline of this algorithm is as follows:

1. Build single input PNN nets. In total, there are seven models, one model for one input candidate.
2. Calculate CAR of each network and select one network to minimize CAR from among the models. The respective input in the selected network is thus identified as a significant input.
3. Fix the confirmed input and add another input to the networks from among the remaining six candidates. A series of two-input PNN nets are built at this stage.
4. Select the second input according to the value of CAR.
5. Continue the above process until CAR does not increase.

This approach was implemented through a Matlab module. The results are shown in Table 2. Considering that our system should be applicable for both crack diagnosis and spalling diagnosis, we combine the confirmed input variables in the two cases, and conclude that all the seven input candidates are all influential variables. Thus, these seven inputs become the antecedent part of our system.

Table 2
Input variable identification result

| Iteration | Variable(s) confirmed | Value of CAR |
|---|---|---|
| *Crack* | | |
| 1 | Damage condition | 0.70642 |
| 2 | Overall width, bridge age | 0.75229 |
| 3 | Damage location, structural type | 0.82569 |
| 4 | Span number | 0.91743 |
| *Spalling* | | |
| 1 | Overall width | 0.65823 |
| 2 | Damage condition | 0.75949 |
| 3 | Bridge age | 0.82278 |
| 4 | Overall length, span number | 0.89873 |
| 5 | Damage location | 0.91139 |

### 2.3. Data filtering module

The original data bank consists of inspection records containing information that embeds subjective judgments from individual inspection engineers and different design codes. It is possible that there are some 'odd' cases from the majority cases, which are called 'special cases' and should be eliminated from the training data. This requires a data filtering module for the proposed system. However, it should be noted that an odd record may not necessarily be attributed to a bad inspection, and that it is separated for the purpose of a guarantee of convergence.

The basic concept of filtering is to classify the scattered data and thus eliminate the special cases, which are graphically far away from the cluster centers. Consider the revised data as patterns, we can represent them by unity vectors that capture their characteristic features. Patterns can be viewed as points in a multidimensional feature space. For classification purpose, we expect that patterns that are similar in some respects, on the basis of class membership or other attribute values, would be close to each other in that pattern space. Thus, patterns belonging to one class would cluster more closely to one another than these patterns belonging to other classes. And the points representing the special cases would scatter from any cluster centers.

Taking consideration of the computation speed and robustness with respect to noisy data as criteria, we chose *modified mountain method* (MMM) [9] as the filter of the system. The filtering procedure undergoes the following steps:

1. Calculate the potential value for each data point:

$$P_j = \sum_{j=1}^{n} e^{-\alpha \|x_i - x_j\|^2} \tag{1}$$

   where $\alpha = 4/r_a^2$ and $r_a$ is a positive constant, $r_a = 0.5$ is adopted in the DIASYN system.
2. Choose the first cluster center with the highest value, then revise the potential of each data point by

$$P_i \Leftarrow P_i - P_1^* e^{-\beta \|x_i - x_1^*\|^2} \tag{2}$$

   where $\beta = 4/r_b^2$ and $r_b$ is a positive constant, $r_b = 1.5 r_a$ in the DIASYN system.
3. Select the next cluster center with the highest remaining potential value, and then further reduce the potential of each data point by

$$P_i \Leftarrow P_i - P_k^* e^{-\beta \|x_i - x_k^*\|^2} \tag{3}$$

   where $x_k^*$ is the location of the $k$th cluster center and $P_k^*$ is its potential value.
4. Repeat the process until either one of the following criteria is satisfied:

$$P_k^* < \underline{\varepsilon} P_1^* \tag{4}$$

$$\underline{\varepsilon} P_1^* \leqslant P_k^* \leqslant \bar{\varepsilon} P_1^* \quad \text{and} \quad \frac{d_{\min}}{r_a} + \frac{P_k^*}{P_1^*} < 1 \tag{5}$$

   where $\bar{\varepsilon}$ and $\underline{\varepsilon}$ specify the thresholds which control the convergence.

Consequently, each cluster center becomes essentially a prototypical data point that exemplified a characteristic behavior of the system. Other data points would be regarded as the representations of special cases and filtered. Following the above procedure, we obtained the training data set as shown in Table 3.

Table 3
Data summary after filtering

| Cause group | Crack | Spalling |
|---|---|---|
| A: Loads and the likes | 10 | 5 |
| B: Design and structural factors | 14 | 4 |
| C: Construction work | 6 | 9 |
| D: Other factors such as aging | 12 | 13 |

### 2.4. Rule generation and optimization module

The α-cuts of fuzzy equivalence relations and fuzzy sets can effectively group numerical data as the basis of the constructions of membership functions for the involved fuzzy variables [10]. Fuzzy rules can be subsequently induced based on the constructed membership functions. Consequently, for a fuzzy system, the rule generation process will thus become a problem of finding the binary equivalence relations between the input or output values. Instead of finding the fuzzy equivalence relations directly, we can determine the fuzzy compatibility relations, which are reflexive and symmetric, in terms of an appropriate function applied to the given data. Then, the fuzzy equivalence relations can be obtained by the max–min transitive closure of the fuzzy compatibility relations.

In our system, the output is categorized into four groups and does not require the procedure of division of the output space. For the input linguistic variables corresponding to each output group, the fuzzy equivalence relations between the input values can be derived after their fuzzy compatibility relations have been constructed. Thus, based on the α-cuts of the generated fuzzy equivalence relations, each input value set is divided into several subsets, and based on the α-cuts of these input subsets, the input membership functions are constructed. With these membership functions, fuzzy rules can be generated based on the hierarchical relationship between the output groups and their corresponding input fuzzy sets. Such procedure is straightforward and is summarized as follows [11]:

1. Divide the training data set into four groups according to the types of outputs.
2. Sort the input values of every input variable in each data group in ascending order. And for every group, execute the following step.
3. Define the fuzzy compatibility relations $R$ between the input values of individual input variables:

$$R = 1 - |x_{i,m} - x_{i,n}|/\delta \tag{6}$$

   where $x_{i,m}$, $x_{i,n}$ are two input values of an input variable $X_i$, and $\delta$ is calculated by

$$\delta = \frac{\sum_{p=1}^{N-1} |x_{i,p} - x_{i,\max}|}{N-1} \tag{7}$$

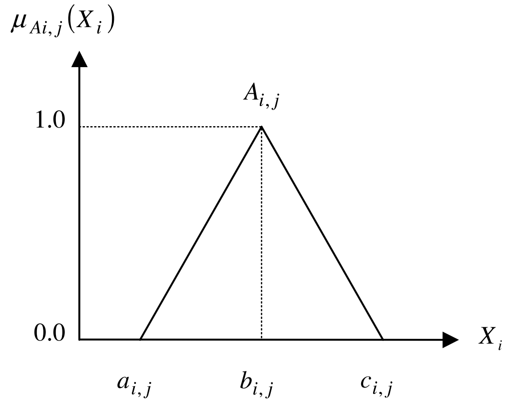


Fig. 4. The triangular fuzzy set with its representative triplet.

Here $N$ is the number of input values of $X_i$, whereas $x_{i,\max}$ is the maximum value.

4. Construct the equivalence relations $R^T$ based on the following steps:
   (a) $R' = R \cup (R \circ R)$.
   (b) If $R' \neq R$, make $R' = R$ and go to step 1.
   (c) Stop: $R' = R^T$.
5. Divide the input values of individual input variables according to the α-cuts of the calculated equivalence relations $R^T$'s.
6. Construct the triangular membership functions for individual input variables, which are expressed by triplets $(a_{i,j}, b_{i,j}, c_{i,j})$ as in Fig. 4. Here $i$ is the index of input variable, $j$ is the number of input groups for $X_i$:

$$\begin{aligned} b_{i,j} &= \frac{x_{i,j,\min} + x_{i,j,\max}}{2} \\ a_{i,j,} &= b_{i,j} - \frac{b_{i,j} - x_{i,j,\min}}{1-\alpha} \\ c_{i,j} &= b_{i,j} + \frac{x_{i,j,\max} - b_{i,j}}{1-\alpha} \end{aligned} \tag{8}$$

7. Generate fuzzy rules based on the hierarchical relationship between the output group and the constructed fuzzy sets of individual input variables. In this phase, all cases in all groups are considered, and the number of created fuzzy rules is

$$\text{Total Num} = \sum_{j=1}^{4} \prod_{i=1}^{7} \text{Num}_j(X_i) \tag{9}$$

   where $\text{Num}_j(X_i)$ is the number of fuzzy sets constructed for input variable $X_i$ in output group $j$.
8. Optimize the generated fuzzy rules based on similarity measures among the fuzzy sets of individual input variables:

$$\text{SM}(A_{1,1,1}, A_{1,2,1}) = \frac{|A_{1,1,1} \cap A_{1,2,1}|}{|A_{1,1,1} \cup A_{1,2,1}|} \tag{10}$$

   where $|A_{1,1,1} \cap A_{1,2,1}|$ is the area of intersection of the two fuzzy sets and $|A_{1,1,1} \cup A_{1,2,1}|$ is the area of their union. Compare this SM with the pre-specified threshold $\alpha_{\text{SM}} (\alpha_{\text{SM}} \in [0,1])$, if $\text{SM} \geqslant \alpha_{\text{SM}}$, we combined the two fuzzy sets $A_{1,1,1}$ and $A_{1,2,1}$ to generate a new fuzzy set $A_{1,\text{new}}$. Accordingly the fuzzy rules are optimized.

The above steps are summarized in Fig. 5.

### 2.5. Inference engine module

The inference engine in DIASYN basically executes *Mamdani's original reasoning* procedure [12]. The overall firing strength of the individual rule whose antecedents are connected with an **AND** operator, the intersection, is typically determined by taking the minimum value of the individual firing strengths of the antecedents. But in the output determination phase, we made some slight changes that can be summarized as follows:

1. When the input is a fuzzy set, the degrees of firing of the individual antecedents are calculated based on a similarity measure, instead of a minimum closure.
2. The fuzzy set inferred by each fired rule is a truth-value set (Fig. 6), not a linguistic fuzzy set, because our output, cause group, is not a linguistic variable.
3. Not every rule is fired regardless to its firing strength. Only the rules with a firing strength higher than a specified threshold will be fired.
4. Only the fired rules with the same consequent parts are aggregated. The output result may include different cause groups with different truth-values.
5. If a crisp result is desired, we give a percentage that represents the inference confidence for each fired cause group.

Conceptually there are five main steps in the inference schema, which are illustrated in Fig. 7.

## 3. Illustrative examples

The applicability of the present system DIASYN is demonstrated by using actual data in damage cause diagnosis of concrete vehicular bridges (VBCs). The results at various stages of DIASYN execution are summarized as follows:

1. Input data points: 82 VBC bridges, 301 damage cases including 184 for crack diagnosis and 117 for spalling diagnosis.
2. Identify the influencing factors using the PNN: seven influencing input variables are identified.
3. Filter training data using MMM: 73 cluster centers are obtained, representing 73 training data points, 42 for crack diagnosis and 31 for spalling diagnosis.

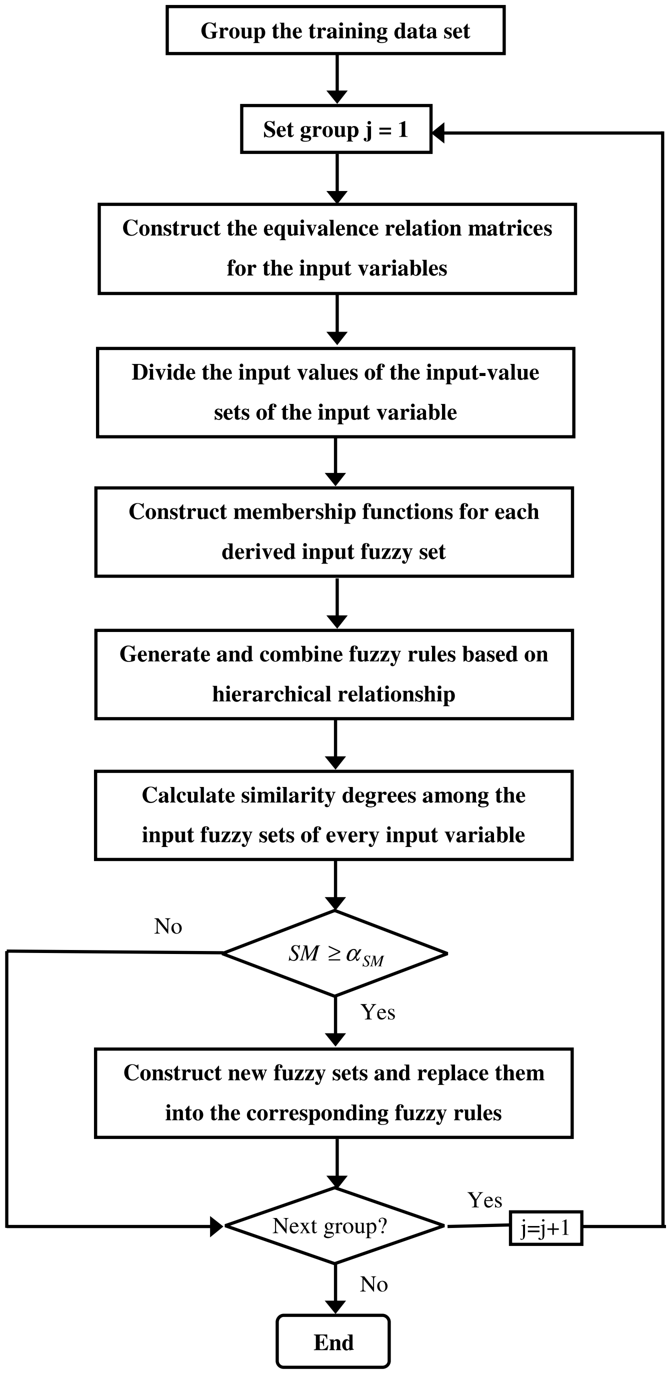


Fig. 5. Flowchart for rule generation and optimization module.

4. Update training data set: choose option 1 when prompted and update the data following the instructions.
5. Generate and modify fuzzy rules using a fuzzy partition method: choose option 2 when prompted. The parameters that identify the membership functions

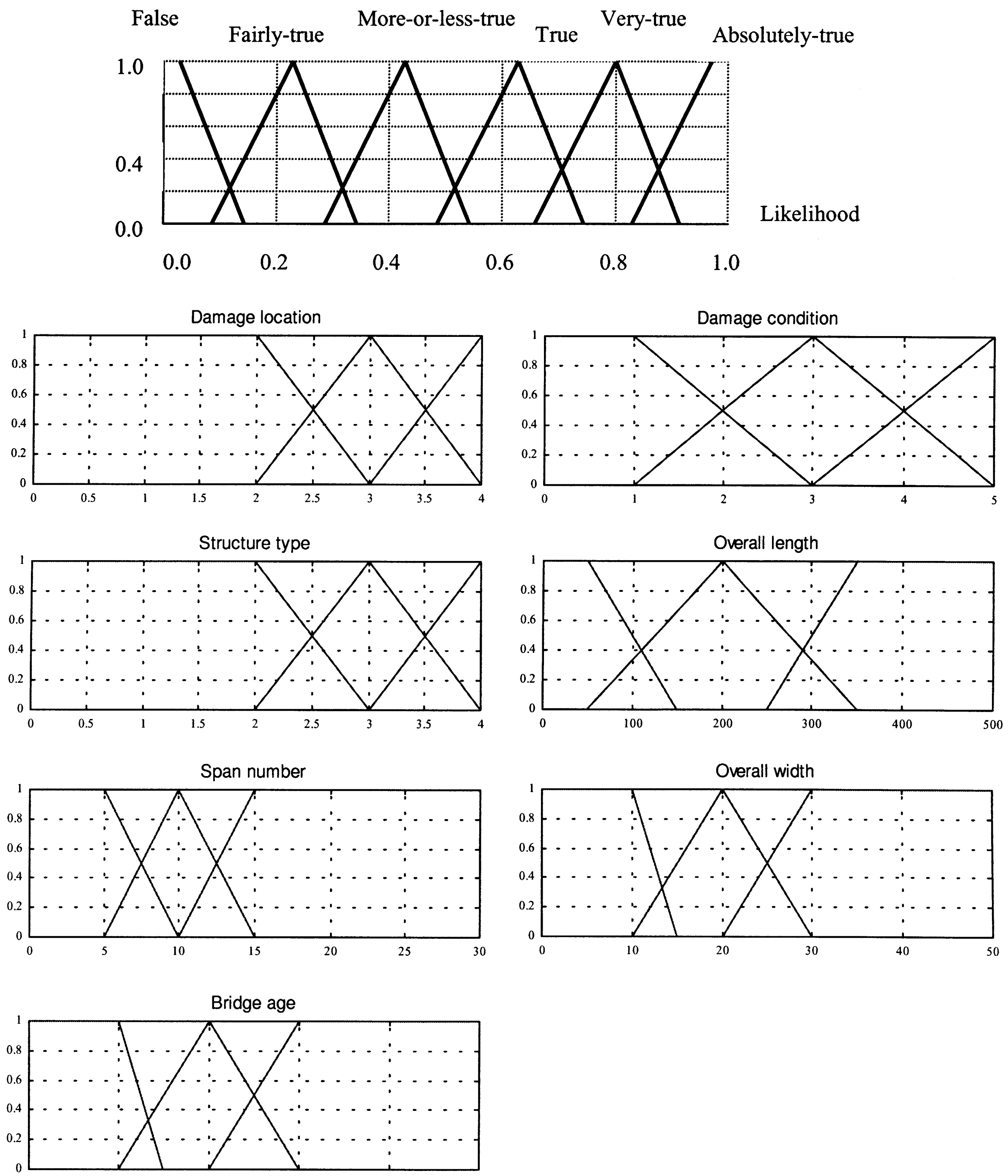


Fig. 6. Membership function of truth-value and initial definition of fuzzy variables for involved variables.

of the antecedent parts of the rules are optimized, rule base is established.

6. Input testing data and compare the results with expert opinions.

### 3.1. Knowledge base of the DIASYN system

The 73 representing training data points for estimating damage causes are

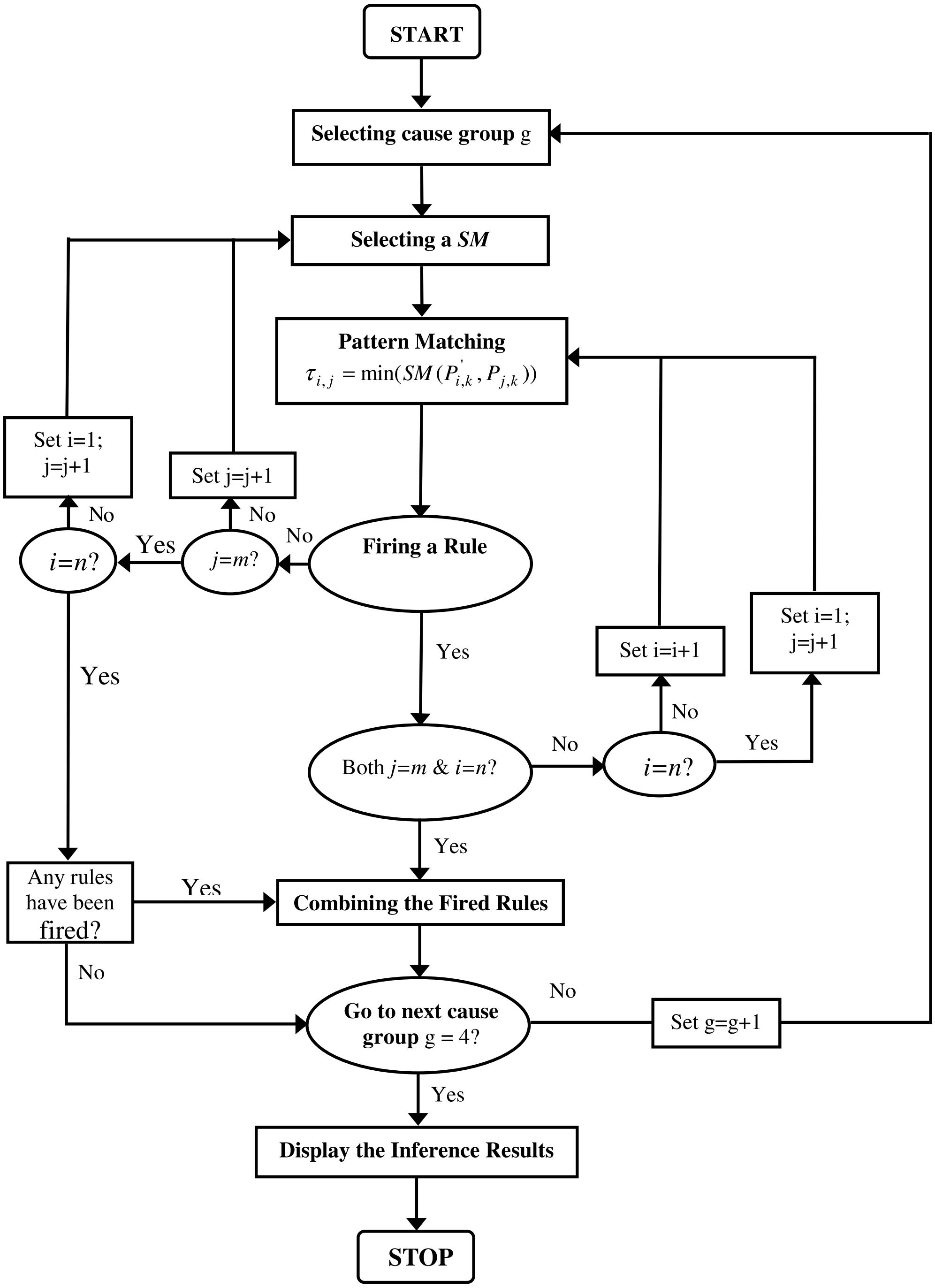


Fig. 7. Flowchart for inference engine module.

- for crack cause group A 10
- for crack cause group B 14
- for crack cause group C 6
- for crack cause group D 12
- for spalling cause group A 5
- for spalling cause group B 4
- for spalling cause group C 9
- for spalling cause group D 13

Some samples of the fuzzy rules generated based on the above data points are illustrated in Fig. 8 and Table 4.

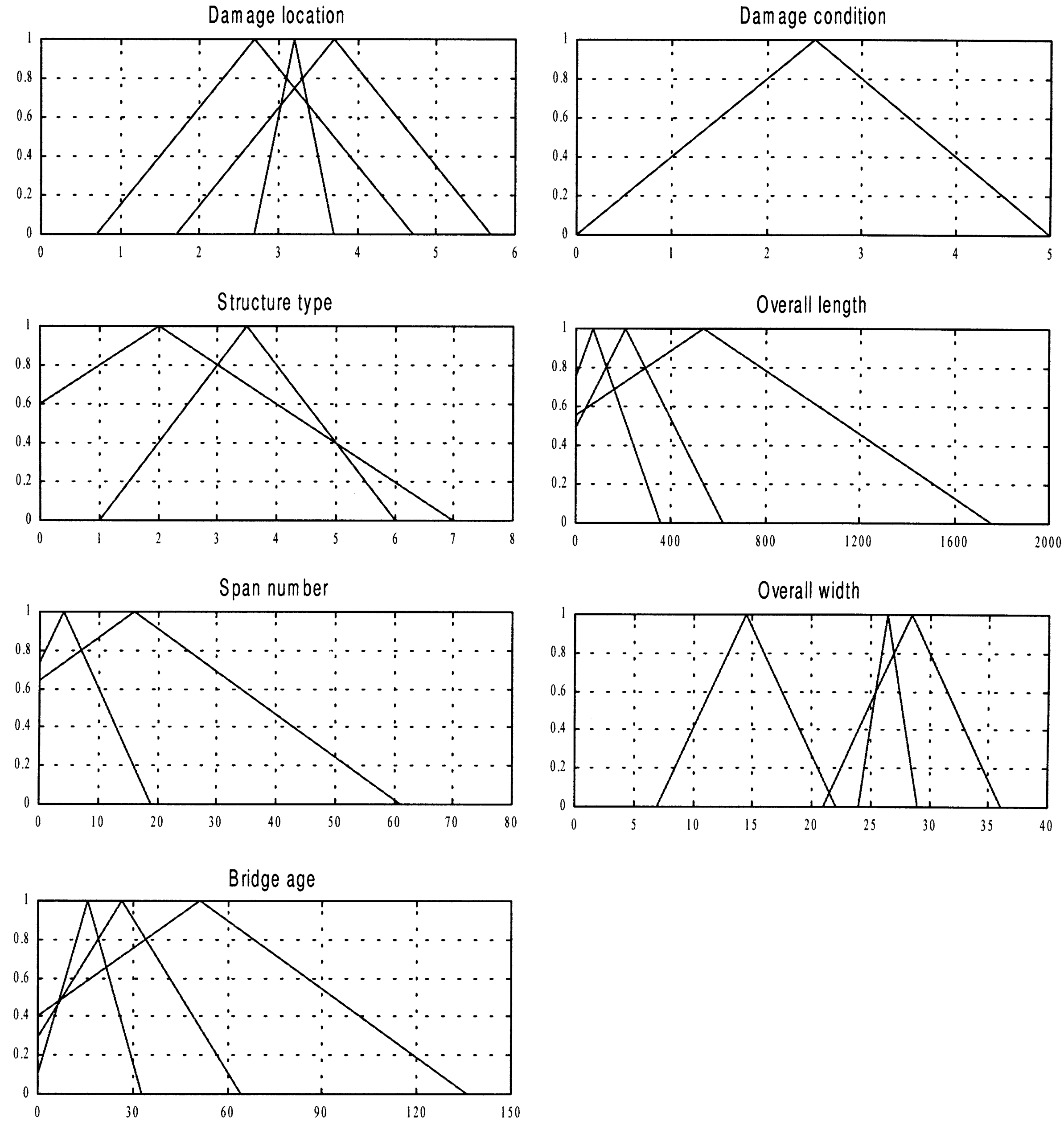


Fig. 8. Sample fuzzy rules.

### 3.2. Test examples

Once the system training is completed, it is ready to be used to diagnose new bridge deterioration case. Two test examples are presented in the follows, one for crack diagnosis and one for spalling diagnosis. The input data of the bridge including survey and inspection information is presented in Table 5, which shows that a crack occurs in superstructure with a condition mark of 1, and a spalling in support-structure with a condition mark of 2. The inference results, along with expert opinions, are listed in Table 6. The results indicate that the particular crack was caused by 'loads and its likes' with a confidence degree of 'very true', and that the spalling was caused by 'others' with a confidence degree of very true. Both of the results are in accordance with the expert opinion, which suggests 'overloaded' and 'aging' are the causes of the crack and spalling, respectively.

Table 4
Sample fuzzy rules

| Parameters and causes | Crack | | Spalling | |
|---|---|---|---|---|
| | Rule 1 | Rule 2 | Rule 3 | Rule 4 |
| Damage condition | About 2 | About 1.5 | About 0.5 | About 0.5 |
| Damage location | Carrying-superstruc. | Carrying-substruc. | Carrying-substruc. | Auxiliary-superstruc. |
| Structural type | PSC beam | RC culvert | PSC beam | RC beam |
| Overall length | About 200 | About 25 | About 600 | About 10 |
| Span number | About 5 | About 2 | About 20 | About 1 |
| Deck width | About 15 | About 10 | About 10 | About 35 |
| Bridge age | About 30 | About 15 | About 50 | About 25 |
| Cause group | A | B | D | C |

Table 5
Inventory and inspection data of example bridge

| Factors | Numeric | Non-numeric |
|---|---|---|
| Bridge age | 16 years | Moderate |
| Overall length | 781 m | Long |
| Span number | 25 | Many |
| Deck width | 16 m | Moderate |
| Structural type | PSC beam | – |
| Condition (crack) | 1 | Good |
| Location (crack) | Superstructure | Very-true |
| Condition (spalling) | 2 | Moderate |
| Location (spalling) | Supportstructure | Very-true |

Table 6
Inference results vs. expert opinion

| Damage | Crack | Spalling |
|---|---|---|
| *Inference results* | | |
| A: Loads & its likes | Very true | Fairly true |
| B: Design & structural factors | False | False |
| C: Construction work | False | False |
| D: Other factors, e.g. aging | More or less true | Very true |
| Expert opinion | Over loaded | Aging |

## 4. Conclusions

The objective of this study is to develop a practical assistant tool for diagnosing deterioration of concrete bridges, which is an essential part of the efficient BMS. The developed DIASYN system makes use of the regular inspection records to derive expert rules for the deterioration diagnosis of concrete bridges. The fuzzy based DIASYN system, though still at the initial development stage, offers several benefits as described in the following:

- By introducing the PNN concept, the input variables adopted in the system can be more reliable and more meaningful. And additional information on the impact of bridge design factors on bridge defects is available to professionals involved.
- Based on the modified mountain clustering method, it is possible to filter the redundant or inaccurate training examples, so that the training data set will be much more efficient.
- Using the proposed fuzzy partition method, knowledge was extracted from numeric training examples in the form of fuzzy if-then rules. Such trial of extraction is useful for building a practical system. Moreover, the system developed facilitates the modification of the knowledge base, based on the similarity measure of the membership functions, which were employed to modify the input variables.
- The efficiency of rule-based reasoning is improved by comparing different inference methods.

The inference results of the system are generally in agreement with the expert's opinion, and can provide substantial assistance to authorities in their planing. The generated rules from the DIASYN system confirms that the use of fuzzy set theory for describing uncertainty and imprecision inherent in knowledge representation could substantially reduce the size of the knowledge base without losing accuracy. The preliminary investigation carried out shows a promising result for the integration of expert systems with the fuzzy theory in structure diagnosis.

There are still a few issues which need to be addressed and improved for the current DIASYN system: (1) other influence factors should be introduced, such as the traffic volume and the history of the previous defects; (2) reliability of the system should be further investigated with respect to the number of influence parameters, the size of the data points and the number of cluster centers, through techniques such as the sensitivity analysis; (3)

the bridge data used are quite small. More bridge data should be added to better present the general bridge conditions in Singapore.

## Acknowledgements

The authors gratefully acknowledge the use of computer facilities and financial support provided by NTU. They also would like to express their appreciation to the Land Transportation Authority, Singapore for their assistance during the data collection.

## References

[1] Hathout I. Damage assessment of existing transmission towers using fuzzy weighted averages. Proceedings of Second International Symposium on Uncertainty Modeling and Analysis, 1993. p. 573–80.
[2] Furuta H, He JH, Watanabe E. A fuzzy expert system for damage assessment using genetic algorithms and neural networks. Microcomput Civil Eng 1996;11:37–54.
[3] Lee J, Liu KFR, Chiang W. A fuzzy petri net-based expert system and its application to damage assessment of bridges. IEEE Trans Syst Man Cybernetics-Part B: Cybernetics 1999;29(3):350–69.
[4] Furuta H, He JH, Watanabe E. A fuzzy expert system for damage assessment using generic algorithm and neural networks. Microcomput Civil Eng 1996;11:37–54.
[5] Kushida M, Miyamoto A, Kinoshita K. Development of concrete bridge rating prototype expert system with machine learning. J Comput Civil Eng, ASCE 1997;11(4): 238–47.
[6] Pandya AS, Macy RB. Pattern recognition with neural networks in C++. Boca Raton (FL): CRC Press, Inc; 1996.
[7] Spetch DF. Probabilistic neural network. J Neural Networks 1990;3:109–18.
[8] Sugeno M, Yasukawa T. A fuzzy-logic-approach to qualitative modeling. IEEE Trans Fuzzy Syst 1993;1(1): 7–31.
[9] Chiu SL. Fuzzy model identification based on cluster estimation. J Intell Fuzzy Syst 1994;2:267–78.
[10] Zadeh L. Fuzzy sets and applications: selected papers. New York: Wiley; 1987.
[11] Wu TP, Chen SM. A new method for constructing membership functions and fuzzy rules from training examples. IEEE Trans Syst Man Cybernetics 1999;29(1): 25–40.
[12] Yager RR, Filev DP. Essential of fuzzy modeling and control. New York: John Wiley & Sons, Inc; 1994.