PERGAMON

Transportation Research Part C 11 (2003) 91–119

TRANSPORTATION RESEARCH PART C

www.elsevier.com/locate/trc

Calibrating the membership functions of the fuzzy inference system: instantiated by car-following data

Partha Chakroborty [a], Shinya Kikuchi [b,*]

[a] Department of Civil Engineering, Indian Institute of Technology-Kanpur, Kanpur 208 016, India

[b] Department of Civil and Environmental Engineering, University of Delaware, Newark, DE 19716, USA

Received 12 October 2000; accepted 18 July 2001

Abstract

The fuzzy rule based inference is known to be a useful tool to capture the behavior of an approximate system in transportation. One of the obstacles of implementing the fuzzy rule based inference, however, has been to calibrate the membership functions of the fuzzy sets used in the rules. This paper proposes a way to calibrate the membership function when a set of input and output data is given for the system. First, the mathematical operations of the fuzzy rule based inference system are represented by a neural network construction. The operations of each node of this neural network are designed so that they correspond to specific logical operations of the fuzzy rule based inference system. The values of the weights of this neural network are set to correspond to the parameters that control the shape and location of each membership function. Second, given a set of input–output data, the weights are corrected sequentially using the principle of the generalized delta rule based back-propagation mechanism. After correction, the values of the weights are used to specify the exact shape of the membership functions of the fuzzy sets in the rules. The procedure implements a set of logical rules that can be applied when calibrating the shapes of the membership functions of a fuzzy inference system. An example, in which the membership functions of a fuzzy inference model for car-following behavior are calibrated using the real world data, is shown.

Keywords: Fuzzy set theory; Calibration of membership function; Neural network; Fuzzy inference system; Car-following model

* Corresponding author. Tel.: +1-302-831-2657; fax: +1-302-831-3640.
E-mail address: kikuchi@ce.udel.edu (S. Kikuchi).

0968-090X/02/$ - see front matter
doi:10.1016/S0968-090X(02)00022-0

1. Introduction

The fuzzy rule based inference system has been recognized as a useful approach to model many complex phenomena in the field of transportation engineering. During the past two decades a number of papers were published, and also many products, both in software and hardware, were introduced. Most applications are found in the areas of inference and control in complex behavioral systems. The first application to transportation was introduced by Pappis and Mamdani (1977) on fuzzy controlled traffic signal, and they set the stage for not only the practical mathematical operations of fuzzy inference but also opened the door to various transportation applications. Perhaps the most significant milestone was the successful real world application to the control of subway vehicles in Sendai, Japan (Sugeno, 1989). Many applications are found in problems that deals with approximate data and approximate reasoning for causal relationships or the stimulus–response process. The areas of application are expanding rapidly. We believe that the fuzzy inference based modeling will play an important role in prediction, diagnosis and control problems in transportation in the future. Good summaries of the past works are found in Hoogendoorn et al. (1998), Dijker and Hoogendoorn (1998) and Teodorovic and Vukadinovic (1998).

A fuzzy rule based inference system consists of a series of "if x is $\tilde{A}$ then y is $\tilde{B}$" rules, where $\tilde{A}$ and $\tilde{B}$ are expressed in linguistic terms and represented by fuzzy sets. This type of inference is suited for modeling systems in which the cause-effect relationships or the logical reasoning processes are inherently fuzzy or approximate. This inference mechanism, however, has suffered from (1) the lack of a procedure that calibrates the membership functions of the fuzzy sets in the rules logically and systematically; and (2) the lack of procedure that develops the optimum set of rules. We feel that the latter is not so much of a problem since the expert's real world experience often helps in the formulation of a reasonable set of rules, and further, the fuzzy inference system (FIS) and its mathematical operations themselves are designed to accommodate the uncertainty of the rule making. On the other hand, the former, determination of the parameter values of the membership functions have always been problematic because the knowledge of the behavior of the system cannot be easily translated into the proper shapes of the membership functions in each rule. This paper develops a procedure that calibrates the membership functions of the fuzzy sets in the rules systematically and according to some logical guidelines.

The key to the proposed procedure is to represent the FIS by an artificial neural network (ANN), which is structured such that the parameters of the membership functions of the fuzzy sets and the logical operations of the inference system are exactly replicated in the weights and the operations of the neural network. The neural network is so constructed that modifying the weights of the neural network effect changes in the shape of each membership function. The paper applies the proposed calibration procedure to a FIS that models a driver's car-following behavior. The results, which utilize real world data on car-following behavior, show that the proposed approach can successfully calibrate the membership functions used in the rules.

The paper is divided into 10 sections, each building foundation for the next. Section 2 explains the nature of the problem. Section 3 describes the FIS for which the proposed calibration approach is applicable. Section 4 outlines the principles of the proposed calibration procedure and presents a set of bases for calibration of FISs. Section 5 introduces the relevant aspects of ANNs. Section 6 describes in detail how to represent the FIS as an ANN. Section 7 develops the calibration process for the membership functions using the generalized delta rule based back-propagation learning algorithm and also illustrates how the process implements the bases for modification discussed in the fourth section. Section 8 describes how to implement the proposed procedure on a computer for practical applications. This section is followed by presentation of the results and conclusions.

2. The nature of the problem

A fuzzy rule based inference system (FIS) consists of a series of rules of the following type: "If x is "large" then y is "very fast", where the "if ..." part is called the antecedent and the "then ..." part is called the consequent of the rule. Expressions like, "large" or "very fast", are fuzzy expressions and are represented using fuzzy sets. The problem is how to specify (or calibrate) the shapes of the membership functions of the fuzzy sets systematically such that the behavior of a particular FIS matches (to a reasonable extent) that of the real world system for which the FIS attempts to model. It is assumed that the behavior of the real world system is specified through a set of input–output data.

The traditional approach to calibration has been the intuitive trial and error process, in which the analyst modifies the shapes of the membership functions little by little until the predicted output approximately fits the output data obtained from the real world. This approach has worked well in many cases. However, this process is time consuming, cannot be used when dealing with a large number of fuzzy sets, and lacks any tenable reasoning.

Wang and Mendel (1992a,b,c) have developed a procedure in which the trial and error process is made somewhat systematic. This method initially assumes a number of membership functions for each fuzzy set and selects the one that matches the input data the most. The procedure is repeated many times to select one of the candidate shapes. This method allows a systematic computation procedure, however, it still remains in the domain of the trial and error process and the results depend on the number of membership functions initially assumed.

Other attempts have also been made. For example, Homaifar and McCormick (1995) developed a genetic algorithm based procedure for FIS calibration. This procedure is an optimization procedure aimed at improving the performance of the FIS through modification of the membership functions and rules of the FIS. However, the procedure does not have any logical bases behind the modification of the membership functions of the FIS; hence, it can be considered as an efficient trial and error method. Others have developed procedures that are applicable only in particular situations. For example, Abe and Lan (1995) have developed a procedure that is applicable for FISs for pattern recognition.

3. The fuzzy rule based inference system being considered

The FIS for which calibration is proposed is presented here. Details of such FISs and the set operations involved in it are found in a number of references on fuzzy set theory. Among them are, Baldwin and Pillsworth (1980), Klir and Folger (1988) and Zimmermann (1996). However, in the following a commonly used form of such a fuzzy rule based inference system is explained briefly.

3.1. Basic structure

A FIS consists of three elements: input, a set of rules, and conclusion. It has the following descriptive form (Note that the number of variables in the antecedent need not be three as shown.):

Input: x is a_k, y is b_k, and z is c_k,
Rules: a set of rules expressed as follows:

$$\begin{array}{llllllll} \text{Rule 1}: & \text{If } x \text{ is } \tilde{A}_1 & \text{AND} & y \text{ is } \tilde{B}_1 & \text{AND} & z \text{ is } \tilde{C}_1 & \text{THEN} & d \text{ is } \tilde{P}_1 \\ \text{Rule 2}: & \text{If } x \text{ is } \tilde{A}_1 & \text{AND} & y \text{ is } \tilde{B}_1 & \text{AND} & z \text{ is } \tilde{C}_2 & \text{THEN} & d \text{ is } \tilde{P}_2 \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \text{Rule } r: & \text{If } x \text{ is } \tilde{A}_i & \text{AND} & y \text{ is } \tilde{B}_j & \text{AND} & z \text{ is } \tilde{C}_q & \text{THEN} & d \text{ is } \tilde{P}_r \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \text{Rule } N: & \text{If } x \text{ is } \tilde{A}_l & \text{AND} & y \text{ is } \tilde{B}_m & \text{AND} & z \text{ is } \tilde{C}_n & \text{THEN} & d \text{ is } \tilde{P}_N \end{array} \tag{3.1}$$

Conclusion: d is $\tilde{D}(k)$,

where a_k, b_k, and c_k are values for x, y and z, respectively for the kth input vector; $\tilde{A}_i$, $\tilde{B}_j$, and $\tilde{C}_j$ in the rules are the fuzzy sets used to describe the different conditions of x, y, and z in natural language, respectively; ℓ, m, and n are the total numbers of fuzzy sets describing the conditions of the variables, x, y, and z, respectively; $\tilde{P}_r$ in the rules is the fuzzy set describing the consequent of Rule r; $\tilde{D}(k)$ is the conclusion given the kth input vector; and N is the total number of rules.

The process of obtaining the conclusion given the input conditions is explained in the following section.

3.2. Computational details

In the following, the mathematical operations of the fuzzy rule based inference system are explained step-by-step. These steps are illustrated in Fig. 1.

Step 1: Determine the match between the input and each of the propositions used in the antecedents of the rules. The degree of match between an input condition, say $x = a_k$, and a proposition, say "x is $\tilde{A}_i$," is measured by the membership value of a_k in the fuzzy set $\tilde{A}_i$; that is, the degree of match is equal to $\mu_{\tilde{A}_i}(a_k)$. In this step, therefore, $\mu_{\tilde{A}_i}(a_k)$ for all i, for all i, $\mu_{\tilde{B}_i}(b_k)$ and for all j, and $\mu_{\tilde{C}_q}(c_k)$ for all q are determined.

Step 2: In this step, the truth value (or degree of applicability) for each rule (or more specifically the antecedent of each rule) for a given input condition is determined. Since the AND operator is used to connect different propositions in the antecedents, the following relation gives the truth value, $w_r(k)$, for Rule r:

$$w_r(k) = \min\left\{\mu_{\tilde{A}_i}(a_k),\ \mu_{\tilde{B}_j}(b_k),\ \mu_{\tilde{C}_q}(c_k)\right\} \tag{3.2}$$

where i, j, and q are such that the fuzzy sets $\tilde{A}_i$, $\tilde{B}_j$, and $\tilde{C}_q$ appear in the antecedent of Rule r.

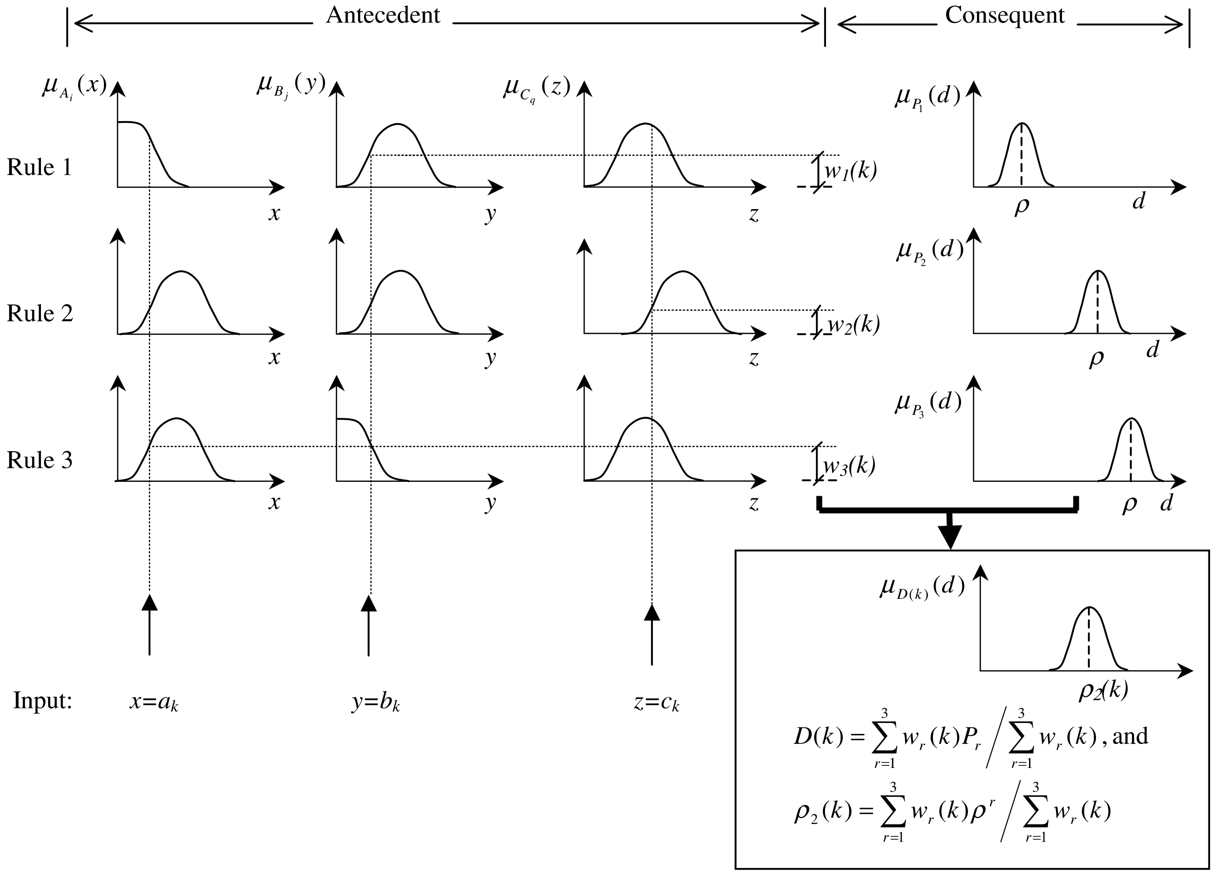


Fig. 1. Illustration of fuzzy rule based inference system.

Step 3: Amongst the many ways of obtaining the conclusion, in this work the conclusion for the kth input condition, $\tilde{D}(k)$, is obtained, in principle, through a weighted average of the consequents, $\tilde{P}_r$'s, of the rules that are "fired" for the kth input vector (i.e., rules with $w_r(k) > 0$). The truth value associated of the rule is taken as the weight of the consequent. Since, the rules which are not relevant for the kth input condition have a $w_r(k) = 0$, in practice, the following relation can be used to obtain $\tilde{D}(k)$:

$$\tilde{D}(k) = \sum_{r=1}^{N} \left(\frac{w_r(k)}{\sum_{r=1}^{N} w_r(k)} \right) \tilde{P}_r \tag{3.3}$$

Before proceeding further it may be mentioned that more often than not the fuzzy set obtained in the conclusion is defuzzified to get a crisp output. This is especially necessary when the FIS represents an engineering system in which the binary decision is necessary.

A possible way of defuzzification is to obtain a defuzzified value, say $\rho_2(k)$, such that

$$\mu_{\tilde{D}_k}(\rho_2(k)) = \alpha \quad \text{where } 0 < \alpha \leqslant 1 \tag{3.4}$$

If such a defuzzification is done then from the properties of fuzzy arithmetic it can be shown that one could directly write the following:

$$\rho_2(k) = \sum_{r=1}^{N} \left(\frac{w_r(k)}{\sum_{r=1}^{N} w_r(k)} \right) \rho^r \tag{3.5}$$

where ρ^r is such that $\mu_{\tilde{P}_r}(\rho^r) = \alpha$ and the value of α is the same as that used in Eq. (3.4).

4. Principles of the proposed calibration procedure

The purpose of calibration is to modify the membership functions of the FIS so that the outcome predicted by the model is equal (or nearly equal) to the outcome obtained in the real world. The modification should be done in a logical manner. In the following, first the parameters of an FIS that affects its conclusion are presented. Next, a set of logical bases for modification of the FIS is presented. Finally, a representation framework that provides a platform for modification of the FIS parameters in consonance with the bases is introduced.

4.1. The parameters of the FIS that affect the conclusion

The factors that dictate the output of the FIS (for a given input vector) are the following:

(i) The parameters which define the membership functions of the fuzzy sets appearing in the antecedents of the rules.
(ii) The parameters which define the membership function of the fuzzy sets appearing in the consequents of the rules.
(iii) The algebraic operators used for (a) the logical connectives and (b) the determination of the final inferred value (or the conclusion).

In a wide variety of FIS applications generally three kinds of fuzzy sets are used in representing the propositions of the antecedents and the consequents of the rules. These are shown in Fig. 2 and are referred to here as Type I unbounded fuzzy sets, Type II unbounded fuzzy sets, and bounded fuzzy sets.

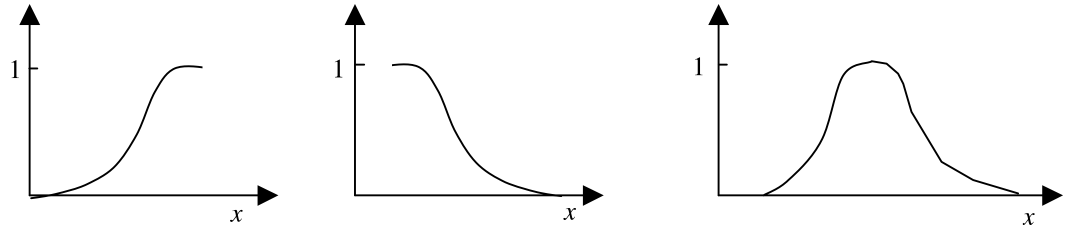
The membership functions of the sets introduced through Figs. 2(a)–(c) can be respectively written using the following functions:

$$\text{Type I unbounded fuzzy set}: \quad f(x) = \frac{1}{1 + e^{-w_2(x-w_1)}} \tag{4.1}$$

$$\text{Type II unbounded fuzzy set}: \quad f(x) = \frac{1}{1 + e^{w_2(x-w_1)}} \tag{4.2}$$

$$\text{Bounded fuzzy set}: \quad f(x) = \frac{1}{1 + e^{-w_{21}(x-w_{11})}} - \frac{1}{1 + e^{-w_{22}(x-w_{12})}} \tag{4.3}$$

The values of w_2, w_{21}, and w_{22} above control the gradient of the membership functions as shown in Fig. 2(d) and are referred to as the gradient parameters. The values of w_1, w_{11}, and w_{12} control the



(a) Type I Unbounded set **(b)** Type II Unbounded set **(c)** Bounded Set

(d) Gradient of the curve controlled by w_2, w_{21}, w_{22} **(e)** Placement of the curve controlled by w_1, w_{11}, w_{12}

Fig. 2. Shapes of the membership functions used and their parameters.

placement of the membership function along the x-axis as shown in Fig. 2(e) and are referred to as the placement parameters. Thus, by manipulating the values of these parameters the shape and the location of the membership functions can be controlled.

It must be mentioned that the fuzzy sets of the antecedents, $\tilde{A}_i$ (for all i), $\tilde{B}_j$ (for all j), and $\tilde{C}_k$ (for all k), are represented by the type shown above in our proposed calibration procedure. The consequents are, however, represented as ρ^r's because one could use Eq. (3.5) to obtain the output $\rho_2(k)$ from the FIS.

4.2. Bases for modifying the shape of a membership function

In this section, the bases for modifying the shape of the membership function are described using the following notation:

- $T(k)$ The kth target value obtained from the real world; it is the output value from the real world for the kth input vector.
- $\rho_2(k)$ The output predicted by the FIS model for the kth input vector; also see Eq. (3.5).
- $w_r(k)$ The truth value (or weight) of Rule r, for the kth input vector obtained using Eq. (3.2).
- ρ^r The consequent of Rule r.

Consider Fig. 3(a) and (b), each showing the target $T(k)$, consequent values ρ^r's of only the rules that fired for the kth input vector, the associated truth values $w_r(k)$'s (written as w_r in the figure), and the corresponding predicted value $\rho_2(k)$ obtained using Eq. (3.5). In Fig. 3(a), the predicted value, $\rho_2(k)$ (represented by the bold, solid arrow), is smaller than the target value,

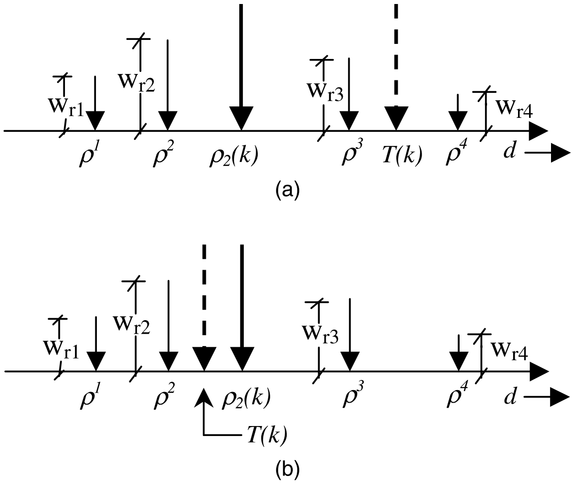


Fig. 3. Two scenarios in the calculation of $\rho_2(k)$ and its comparison with target value, $T(k)$.

$T(k)$ (represented by the bold, broken arrow) and hence is found to the left of $T(k)$. In Fig. 3(b), the predicted value is greater than the target value and hence is found to the right of the target value. These figures are provided here so that the reader may readily see the effects of the various bases for modifying the parameters on $\rho_2(k)$.

The objective, here, is to make $T(k)$ and $\rho_2(k)$ as close as possible by: (a) modifying the gradient and placement parameters of the membership functions, so that the $w^r(k)$ values are adjusted; and, (b) modifying the values of ρ^r.

Specifically, the bases for modification of the parameters are as follows:

Basis 1: Do not modify any parameter of the fuzzy sets of (a) the antecedent or (b) the consequent of Rule r, if $w_r(k) = 0$.
Reasons: $w_r(k) = 0$ means that Rule r did not fire, and hence, it did not play any role in the determination of the conclusion (or output). Therefore, the mismatch between the observed value $T(k)$ and the predicted conclusion value $\rho_2(k)$ cannot be attributed to the parameters of the membership functions of Rule r.

Basis 2: Change the relevant parameter values so as to increase the ρ^r values, if $T(k) > \rho_2(k)$, and decrease the values of ρ^r, if $T(k) < \rho_2(k)$. In other words, in case of Fig. 2(a), the values of ρ^r should be increased; while in case of Fig. 2(b) the values should be decreased.
Reasons: A possible reason for $T(k) > \rho_2(k)$ is that the consequents of the rules that determine $\rho_2(k)$ are less than what they should be; and thus, the value of the consequents should be increased. The opposite is true for the case when $T(k) < \rho_2(k)$.

Basis 3: Change the relevant parameters so as to increase the value of $w_r(k)$, if

$$(\rho^r - \rho_2(k)) \times (T(k) - \rho_2(k)) > 0$$

For example, in case of Fig. 2(a), $w_{r3}(k)$ and $w_{r4}(k)$ should be increased. Similarly, in case of Fig. 2(b), $w_{r1}(k)$ and $w_{r2}(k)$ should be increased (also see Basis 5).

Reasons: Suppose $T(k) > \rho_2(k)$; one of the reasons for $T(k)$ being greater than $\rho_2(k)$ could be that the rules which provide consequents which are *larger than* $\rho_2(k)$ are actually more applicable than their current level of applicability. Therefore, the $w_r(k)$ values (which indicate applicability of Rule r) of such rules should be increased (by modifying the membership functions in the antecedents of such rules). On the other hand, suppose $T(k) < \rho_2(k)$; one of the reasons for such an occurrence could be that the rules which provide consequents which are *smaller than* $\rho_2(k)$ are actually more applicable than their current level of applicability. Therefore, the $w_r(k)$ values of such rules should be increased.

Basis 4: Change the relevant parameters so as to decrease the value of $w_r(k)$ if

$$(\rho^r - \rho_2(k)) \times (T(k) - \rho_2(k)) < 0$$

For example, in case of Fig. 2(a), $w_{r1}(k)$ and $w_{r2}(k)$ should be decreased. Similarly, in case of Fig. 2(b), $w_{r3}(k)$ and $w_{r4}(k)$ should be decreased (also see Basis 5).

Reasons: Suppose $T(k) > \rho_2(k)$; one of the reasons for $T(k)$ being greater than $\rho_2(k)$ could be that the rules which provide consequents which are *smaller than* $\rho_2(k)$ are actually less applicable than their current level of applicability. Therefore, the $w_r(k)$ values of such rules should be decreased. On the other hand, suppose $T(k) < \rho_2(k)$; one of the reasons for such an occurrence could be that the rules which provide consequents which are *larger than* $\rho_2(k)$ are actually less applicable than their current level of applicability. Therefore, the $w_r(k)$ values of such rules should be decreased.

Basis 5: Modify the parameters of the membership function of only that fuzzy set of the antecedent of Rule r which controls the value of $w_r(k)$. If two or more fuzzy sets of the antecedent had yielded values close to $w_r(k)$ then modify the membership functions of each of these fuzzy sets.

Reasons: The applicability, $w_r(k)$, of Rule r is determined by the fuzzy set of the antecedent which yields the least value of the membership grade for the given input condition. Hence, membership function of only that fuzzy set should be modified. If, however, more than one fuzzy set yield values close to the minimum value (which is, $w_r(k)$), then all the fuzzy sets which yielded values close to the minimum should be modified.

These are the bases for modifying the parameters of the fuzzy rule based inference system in this paper. The primary task that remains is the development of a procedure which implements the bases presented above.

4.3. Representation of the FIS for its modification

Most commonly, the FIS's are represented in the "*If–then–else*" format. In order to implement the modification bases described above, a different representation of an FIS is developed. It is a multi-layer ANN. This representation has been motivated by the facts that (i) the sigmoid activation functions (or any such bounded activation functions) of an ANN can aptly represent membership functions of fuzzy sets, and (ii) the framework of ANNs offer systematic calibration (learning) mechanisms.

It will be shown that a one-to-one correspondence between the operations of an FIS and its ANN counterpart can be established. More importantly, it will be proven here that the generalized delta rule based back-propagation learning algorithm of ANNs implements the bases for FIS modification presented in the previous section. The next three sections discuss: how an ANN can be used to represent the FIS, and how the generalized delta rule based back-propagation learning mechanism implements the bases for FIS modification, respectively.

5. Relevant aspects of artificial neural networks

The key to the proposed calibration procedure is to represent the fuzzy rule based inference system as an ANN. This section introduces the features of an ANN that are particularly relevant to the proposed procedure and the notation used here in describing an ANN. The discussion on ANNs is kept brief; good discussions can be found in Rumelhart et al. (1986a), Wasserman (1989) and Hecht-Nielsen (1990) among others.

5.1. Notation and overview of an artificial neural network

An ANN is a parallel distributed processing system as shown in Fig. 4. In the figure, the circles (nodes) represent the processing units. They are typically grouped into "layers." The directional arcs in the figure represent the weighted connections between processing units of adjacent layers. These arcs transmit the output from one node of a layer to a node of the subsequent layer. A detailed description of the processing units and the connections is provided later. However, before that, the notation used here and in later sections is introduced.

Li The ith layer of the neural network; it is conglomeration of nodes generally performing similar operations.

$\mathrm{Net}(i,J,k)$ A value residing in node J of the ith layer for the kth input vector. It is obtained by operating on the inputs to the node.

$\mathrm{Var}(i,J,k)$ A value residing in node J of the ith layer for the kth input vector. It is obtained by operating on the $\mathrm{net}(i,J,k)$.

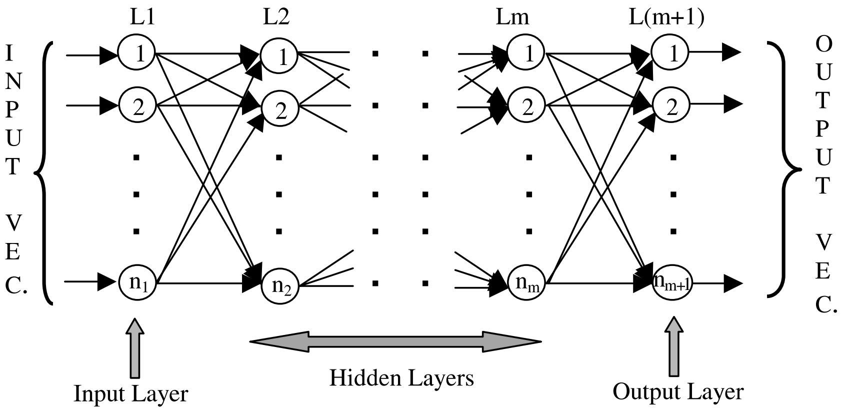


Fig. 4. The structure of an ANN model.

$w(I,J,K)$ The weight of the link (directional arc) connecting Node J of the ith layer to Node K of the $(I+1)$th layer. Further, if $w(i,J,K)$ is referred to as $+w(i,J,K)$ then it means that the weight can take only positive values; if the weight is referred to with a negative sign then it means that the weight can take only negative values. If no sign is provided, then the weight can take any value.

$E(k)$ The error (or mismatch) between the target value (observed value) and the predicted value for the kth input vector.

Typically, an ANN model attempts to capture the relationship between the input and output of a real world system by:

(i) Receiving inputs through a layer of nodes (the input layer), processing them in nodes of various layers (the input layer, the hidden layers and the output layer) connected through the weighted directional arcs and producing an output; and
(ii) Comparing the output of the ANN with the target (i.e., the desired output); based on the comparison, the ANN modifies the weights associated with each directional arc, so that the predicted output becomes closer to the target. This is called learning.

The topics relevant for understanding the ANN will be discussed. They are: (i) the processing units (nodes) and the weights of connections, (ii) the pattern of connectivity (i.e., the type of connections between the nodes), and (iii) the learning mechanism.

5.2. The processing units and the weights of connection

All the computations of an ANN are performed at the nodes. Hence, defining the computations in each node is crucial to the development of a meaningful ANN. Some of the nodes are used to represent the concepts that constitute the antecedents of rules, others are used to represent the truth value of a rule for a particular input vector, and yet others are used as abstract elements in which meaningful operations are performed.

In Fig. 5, the circle represents a node. Each chamber in the node denotes a process. In the far left chamber, variables coming in as inputs from the nodes of the previous layer are stored. These variables then crossover to the middle chamber after being aggregated into a single value, $\mathrm{net}(i,J,k)$. Finally, this value crosses over to the right chamber by transforming to $\mathrm{var}(i,J,k)$. The $\mathrm{var}(i,J,k)$ is then transmitted through connections emanating from the node to nodes of the next layer.

The processing units receive inputs from other processing units through some connections. Associated with each connection is a weight. The output of a node is multiplied with a weight before it is passed on as input to the node on the next layer. For example, the input to the processing unit in Fig. 3 are, $w(i-1,J_1,J)\times \mathrm{var}(i-1,J_1,k)$, $w(i-1,J_2,J)\times \mathrm{var}(i-1,J_2,k)$, and $w(i-1,J_3,J)\times \mathrm{var}(i-1,J_3,k)$.

The inputs are transformed into $\mathrm{net}(i,J,k)$ through various algebraic operators. In our representation of fuzzy inference, two such operators are used. In certain cases, the following summation operator is used:

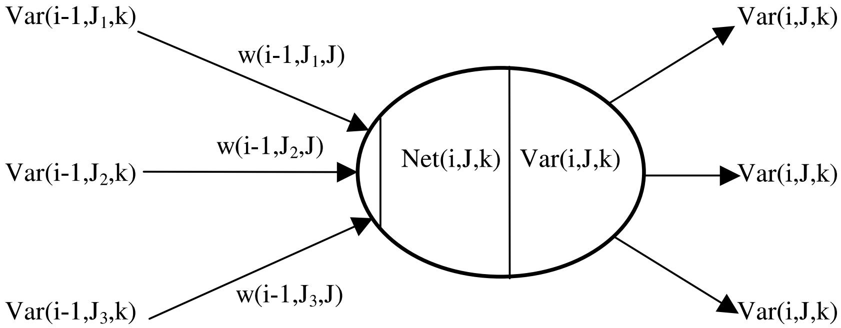


Fig. 5. Structure of a processing unit showing the notation used.

$$\text{net}(i,J,k) = \sum_{\forall j} w(i-1,J_j,J) \times \text{var}(i-1,J_j,k) \tag{5.1}$$

While in other cases, a differentiable approximation of the minimum operator, $m(\cdot)$ is used

$$\text{net}(i,J,k) = \underset{\forall j}{m}(w(i-1,J_j,J) \times \text{var}(i-1,J_j,k)) \tag{5.2}$$

The approximation to the minimum operator stated simply is as follows:

$$m(x,y) = \begin{cases} x & \text{if } x \leqslant y-\delta \\ y & \text{if } x \geqslant y+\delta \\ \phi(x,y) & \text{if } y-\delta < x < y+\delta \end{cases}$$

where $\phi(x,y)$ is a continuous increasing function which has a value of x and a unit slope with respect to x at $x = y-\delta$, and a value of y and a zero slope with respect to x at $x = y+\delta$.

The transformation of $\text{net}(i,J,k)$ to $\text{var}(i,J,k)$ takes place through a function called the *activation function*. Among many types of activation functions, two are of particular interest here; namely, a linear identity function, and a non-linear bounded function. This function is of the type $f(\text{net}(i,J,k)) = \text{net}(i,J,k)$; it simply passes the information without altering it, i.e., $\text{var}(i,J,k) = \text{net}(i,J,k)$.

The non-linear bounded function, used here, belongs to a class of functions which map the $\text{net}(i,J,k)$ to the closed interval [0,1]. The output from such functions can be interpreted as membership values in fuzzy sets (Williams, 1986). The particular type of function used here is called the sigmoid function and is given by:

$$\text{var}(i,J,k) = f(\text{net}(i,J,k)) = \frac{1}{1+\mathrm{e}^{-\text{net}(i,J,k)}} \tag{5.3}$$

5.3. Pattern of connectivity

The particular way in which processing units of different layers are connected to one another is referred to as the pattern of connectivity. This pattern defines the computation process. The issues related to the pattern of connectivity are: (i) whether a particular processing unit should connect to another unit, and (ii) if it should, whether the connection should be excitatory (positive weight) or inhibitory (negative weight).

5.4. Learning

The process of learning in the ANN amounts to modifying the weight of the connections in light of the input–output data obtained in real world. The value of the weight may be modified based on the mismatch (often referred to as the *error*) between the predicted output and the target. The way in which the mismatch is utilized to correct the weights is referred to as the learning rule or correction algorithm. A particular learning rule that is of interest here is the *generalized delta rule based back-propagation learning mechanism*. An excellent discussion on this learning mechanism is found in Rumelhart et al. (1986b).

This method corrects the weight values proportional to the degree to which the weights affect the error. Given the existing weight, $w(i,J,K)$, the modified weight, $w'(i,J,K)$, is obtained as follows:

$$w'(i,J,K) = w(i,J,K) + \Delta w(i,J,K) \tag{5.4}$$

where

$$\Delta w(i,J,K) = \eta\delta(i+1,K,k)\frac{\partial \mathrm{net}(i+1,K,k)}{\partial w(i,J,K)} \tag{5.5}$$

In this equation, η is the proportionality constant (which defines the step size of a correction to a weight) and

$$\delta(i+1,K,k) = -\frac{\partial E(k)}{\partial \mathrm{net}(i+1,K,k)} \tag{5.6}$$

In general, $\delta(i,J,k)$, can be written as

$$\delta(i,J,k) = \frac{\partial \mathrm{var}(i,J,k)}{\partial \mathrm{net}(i,J,k)}\sum_{K\in\Psi}\delta(i+1,K,k)\frac{\partial \mathrm{net}(i+1,K,k)}{\partial \mathrm{var}(i,J,k)} \tag{5.7}$$

where Ψ is the set of nodes in Layer $i+1$ to which Node J of Layer i is connected.

Note, Eq. (5.7) expresses $\delta(i,J,k)$ values for nodes of Layer i in terms of the $\delta(\cdot)$ values of the subsequent layer, Layer $i+1$.

6. Neural network representation of fuzzy rule based inference system

This section develops the ANN that represents a FIS. An important task is to represent the membership functions of the fuzzy sets of the FIS in the language of ANNs. Hence, this section first describes how a membership function can be represented, and later, presents how the entire FIS can be represented.

6.1. Artificial neural network representation of the membership functions

The three types of membership functions described in Section 4.1 is represented by an ANN in this section. The procedure used here is derived from the work of Horikawa et al. (1991). A little thought and the description of the membership functions help in realizing that such membership functions can be represented by the nodes and arc constructions shown in Fig. 6 (a more detailed exposition of this development can be found in Chakroborty (1993)).

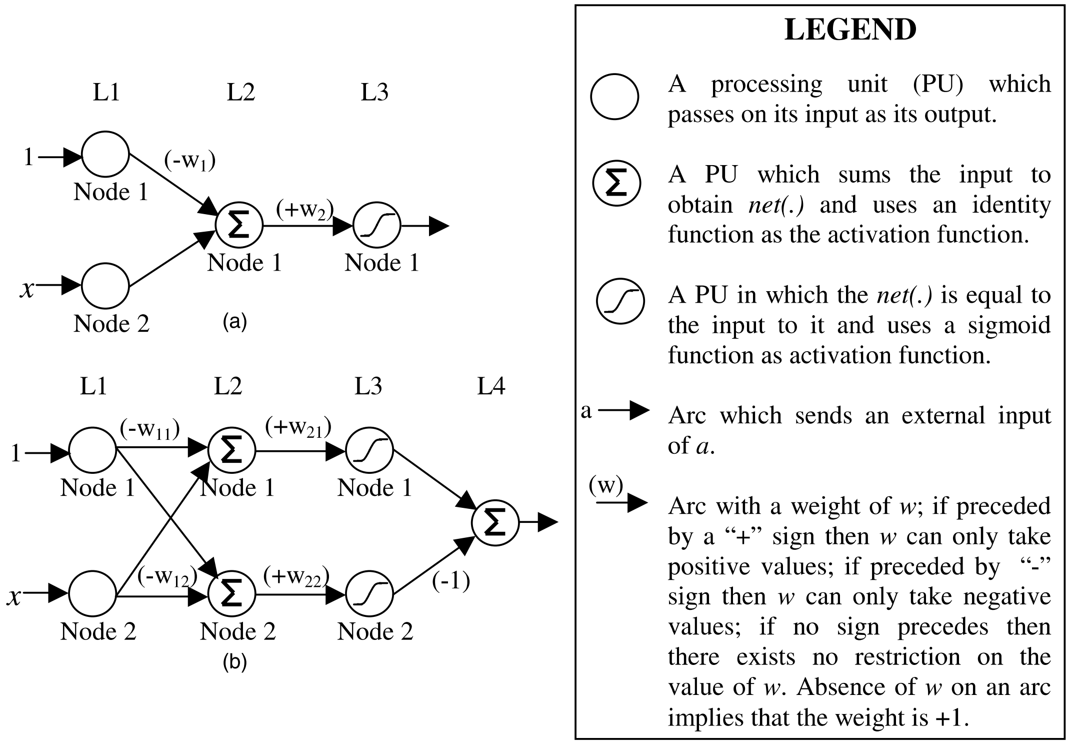


Fig. 6. Membership function construction by an ANN model.

Fig. 6(a) shows how a Type I unbounded fuzzy set can be represented in the structure of an ANN. It is seen that given a value of x, the output from Node 1 of L3 (Layer 3) gives the membership grade of x in the fuzzy set represented by the membership function: $1/\{1+\mathrm{e}^{-(+w_2)(x+(-w_1))}\}$ (Recall that $(+w)$ means a positive number and $(-w)$ means a negative number). The following explains how Node 1 of Layer 3 gives the membership grade, as claimed here, by going through the computations of the nodes of every layer:

Layer 1 Node 1, input x is given; Node 2, input of 1 is given; $\mathrm{var}(i,J,k)$ for these nodes are equal to their inputs as these nodes pass on their inputs as their outputs (see Legend of Fig. 6).

Layer 2 Input to this node are x and $-w_1$ times 1; $\mathrm{net}(i,J,k)$ for this node is obtained by summing the two inputs and is therefore equal to $(x-w_1)$; $\mathrm{var}(i,J,k)$ for this node is equal to $\mathrm{net}(i,J,k)$ as an identity activation function is used in this node.

Layer 3 Input to this node is w_2 multiplied by $(x-w_1)$; $\mathrm{net}(i,J,k)$ for this node is equal to the input as there is only one input; the output from this node (i.e., $\mathrm{var}(i,J,k)$ for this node) is equal to $1/\{1+\mathrm{e}^{-w_2(-w_1+x)}\}$ as a sigmoid activation function is used here.

The structure shown in Fig. 6(a) can also be used to represent membership functions of Type II unbounded fuzzy sets. This can be done by simply stipulating that the weight $(+w_2)$ should be $(-w_2)$; that is the weight can only take negative values as opposed to only positive values.

Fig. 6(b) shows how a bounded fuzzy set can be represented. Given a value of x, the output from Node 1 of L4 (Layer 4), gives the membership value of x in the fuzzy set represented by the membership function: $(1/\{1+\mathrm{e}^{-(+w_{21})(x+(-w_{11}))}\})-(1/\{1+\mathrm{e}^{-(+w_{22})(x+(-w_{12}))}\})$.

6.2. ANN representation of the entire fuzzy inference system

This section presents a detailed description of the ANN that represents the FIS, described in Section 3. Specifically, the description is based on the ANN representation of the FIS shown in Eq. (3.1) with the following characteristics: (i) $l=3$, $m=3$, and $n=4$, (ii) $N=3\times3\times4=36$, (iii) $\tilde{A}_1$, $\tilde{B}_1$, and $\tilde{C}_1$ are Type II unbounded fuzzy sets, (iv) $\tilde{A}_3$, $\tilde{B}_3$, and $\tilde{C}_4$ are Type I unbounded fuzzy sets, and (v) $\tilde{A}_2$, $\tilde{B}_2$, $\tilde{C}_2$, and $\tilde{C}_3$ are bounded fuzzy sets.

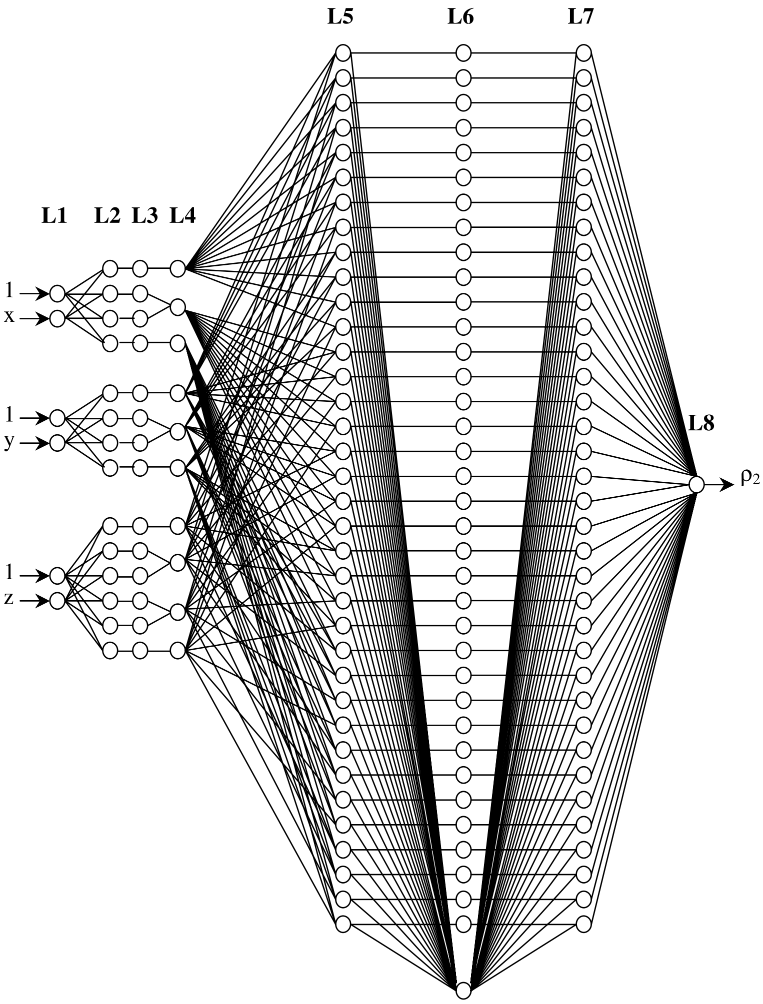


Fig. 7. Representation of the FIS described in Section 3 as an ANN model.

Table 1
Description of the tasks and outputs of the various processing units (nodes) of the ANN representation shown in Fig. 7

Layer	Node number(s)	Characteristics	Remarks (meaning of the output)
L1	1, 3, 5	Similar to Node 1 of L1 of Fig. 6(a) or (b). (Nodes which receive an external input of 1.)	Each node passes on its input as its output.
	2, 4, 6	Similar to Node 2 of L1 of Fig. 6(a) or (b). (Nodes which receive the input vector $\{x_k, y_k, z_k\}$. That is, Node 2 receives x_k, Node 4 receives y_k, etc.)	Each node passes on its input as its output.
L2	1, 5, 9	Similar to Node 1 of L2 of Fig. 6(a) when it represents Type II unbounded fuzzy set.	
	4, 8, 14	Similar to Node 1 of L2 of Fig. 6(a) when it represents Type I unbounded fuzzy set.	
	2, 6, 10, 12	Similar to Node 1 of L2 of Fig. 6(b).	
	3, 7, 11, 13	Similar to Node 2 of L2 of Fig. 6(b).	
L3	1, 5, 9	Similar to Node 1 of L3 of Fig. 6(a) when it represents Type II unbounded fuzzy set.	Gives membership in sets, $\tilde{A}_1$, $\tilde{B}_1$, and $\tilde{C}_1$, respectively.
	4, 8, 14	Similar to Node 1 of L3 of Fig. 6(a) when it represents Type I unbounded fuzzy set.	Gives membership in sets $\tilde{A}_3$, $\tilde{B}_3$, and $\tilde{C}_4$, respectively.
	2, 6, 10, 12	Similar to Node 1 of L3 of Fig. 6(b).	
	3, 7, 11, 13	Similar to Node 2 of L3 of Fig. 6(b).	
L4	1, 3, 4, 6, 7, 10	Units which get a single input (hence $\text{net}(\cdot) = \text{input}$) and use an identity function as the activation function.	Each node passes on its input as its output.
	2, 5, 8, 9	Similar to Node 1 of L4 of Fig. 6(b).	Gives membership in sets $\tilde{A}_2$, $\tilde{B}_2$, $\tilde{C}_2$, and $\tilde{C}_3$, respectively.
L5	1–36	Each node uses $m(\cdot)$ to determine $\text{net}(\cdot)$ from the inputs it receives. Each of them uses an identity function as the activation function. Node 1 receives as input the memberships in sets, $\tilde{A}_1$, $\tilde{B}_1$, and $\tilde{C}_1$; Node 2 receives as input the memberships in sets, $\tilde{A}_1$, $\tilde{B}_1$, and $\tilde{C}_2$; and so on.	Each gives the truth of a rule (i.e., $w_r(k)$ of Eq. (3.2)) for the kth input vector. For example, Node 1 gives the truth of Rule 1 (in Eq. (3.1)), Node 2 of Rule 2, etc.
L6	1–36	Units which get a single input (hence $\text{net}(\cdot) = \text{input}$) and use an identity activation function as the activation function.	Each node passes on its input as its output.
	37	Uses a summation operator to obtain $\text{net}(\cdot)$. Uses an identity function as the activation function.	Gives the sum of all the truth values.
L7	1–36	Receives input from the corresponding node number of L6 and also from Node 37 of L6. Uses a division operator (the former upon the latter) to obtain $\text{net}(\cdot)$. Uses an identity function as the activation function.	Gives the normalized truth values for all the rules; i.e., each node gives $w_r(k)/\Sigma_r w_r(k)$ for a given r.
L8	1	Uses a summation operator to obtain $\text{net}(\cdot)$.	Gives a defuzzified value of the output from the FIS (i.e., $\rho_2(k)$) for a given input vector.

Fig. 7 shows the ANN structure that exactly replicates the FIS (in Eq. (3.1) with the characteristics mentioned in the previous paragraph).

Each line in Fig. 7 is an arc (connection) leaving from the node to the left and going to the node on the right. The processes carried out at the various nodes and the weights of the arcs are explained in Tables 1 and 2, respectively. Table 1 also provides the meanings of the outputs from the nodes in the *Remarks* column, wherever applicable. Note that, the nodes are numbered (although

Table 2
Weights on connections between nodes of successive layers in Fig. 7

From layer (value of i)	From node number(s) (value of J)	To node number(s) (value of K)	Weight $w(i,J,K)$
L1 ($i=1$)	1	1–4	$-w(1,1,K)$
	2	1–4	$+1$
	3	5–8	$-w(1,3,K)$
	4	5–8	$+1$
	5	9–14	$-w(1,5,K)$
	6	9–14	$+1$
L2	2–14 except 5 and 9	J	$+w(2,J,J)$
	1, 5, 9	J	$-w(2,J,J)$
L3	1, 2	J	$+1$
	4, 5, 6	$J-1$	$+1$
	8, 9, 10	$J-2$	$+1$
	12	9	$+1$
	14	10	$+1$
	3	2	-1
	7	5	-1
	11	8	-1
	13	9	-1
L4	1	1–12	$+1$
	2	13–24	$+1$
	3	25–36	$+1$
	4	1–4, 13–16, 25–28	$+1$
	5	5–8, 17–20, 29–32	$+1$
	6	9–12, 21–24, 33–36	$+1$
	7	1, 5, 9, 13, 17, 21, 25, 29, 33	$+1$
	8	2, 6, 10, 14, 18, 22, 26, 30, 34	$+1$
	9	3, 7, 11, 15, 19, 23, 27, 31, 35	$+1$
	10	4, 8, 12, 16, 20, 24, 28, 32, 36	$+1$
L5	1–36	J	$+1$
	1–36	37	$+1$
L6	1–36	J	$+1$
	37	1–36	$+1$
L7	1–36	1	$w(7,J,1)$

not specified in Fig. 7) sequentially from top to bottom; i.e., Node n of a layer refers to the nth node in that layer from the top.

Table 2 presents the weights of all the connections (arcs) shown in Fig. 7. The table has four columns. The first column gives the layer from which the arcs emanate. The second column gives the node number from which an arc emanates, the third column gives the node number (of the immediately subsequent layer) to which the arc connects. The last column gives the weight associated with the arc described through the first three columns. A "+" before a weight means that the weight cannot be negative. A "−" sign means that the weight cannot be positive. If no sign is mentioned then it means that the weight could be any real number. Further, all weights that are stated using the notation $w(i,J,K)$ mean that these weights are variables and are to be calibrated. The rest of the weights are fixed. If a particular connection between two nodes is not mentioned in the table, then it means that no connection exists (and should not exist) between these two nodes. Another clarification regarding the weights of arcs emanating from Layer 7 needs to be given: note that, $w(7,J,1)$'s, represent ρ^r's; in fact, $w(7,1,1)$, represents ρ^1, $w(7,2,1)$, represents ρ^2, and so on.

Fig. 7 together with Tables 1 and 2 describes the ANN that replicates the FIS of Eq. (3.1). The following description shows how the different layers replicate the different computations of the FIS.

Given the input values of x, y, and z the following operations take place in the ANN:

- Layers 1, 2, 3 and 4, which are constructed from the basic modules presented in Fig. 6, compute the membership grades of the input vector in each of the 10 ($= 3 + 3 + 4$) fuzzy sets used in the FIS. This corresponds to the Step 1 explained in Section 3.
- Each node of Layer 5 collects inputs from three different nodes (which represent the membership grades of the input vector in the three different fuzzy sets constituting the antecedent of a rule). Each node then performs the minimum operation (corresponding to Eq. (3.2)) on the inputs it receives. Each node thus provides (as its output) the truth value of the antecedent of a particular rule. This operation corresponds to Step 2 explained in Section 3.
- Layers 6 and 7 calculate the normalized weight for each Rule, r; the normalized weight of a rule is the parenthetically enclosed term in Eq. (3.3) (and also in Eq. (3.5)).
- Layer 8 produces the predicted defuzzified output based on Eq. (3.5).

7. Application of the generalized delta rule and bases for calibrating the membership functions

Once the ANN representation of the inference system is complete, we "train" the neural network using the *generalized delta rule* based back-propagation algorithm. The training involves correction of the weights. This identifies the weights that ought to be corrected. First, the generalized delta rule is used to derive the weight modifiers (i.e., amount by which the weights should be modified or corrected) when there exists a mismatch between the predicted output and the target. Second, a discussion is provided to show how these weight modifiers implement the bases (described in Section 4.2) for modification of an FIS.

7.1. Weight modifiers, $\Delta w(i,J,K)$

The main task in determining the $\Delta w(i,J,K)$ terms is to calculate the values of the $\delta(i,J,k)$ terms. A more detailed and complete derivation of these terms can be found in Chakroborty

(1993). In order to determine the weight modifiers, first the error term (the differences between the predicted and the target) needs to be defined. The error term used here is as follows:

$$E(k) = \tfrac{1}{2}(T(k) - \mathrm{var}(8,1,k))^2 \tag{7.1}$$

The $\delta(i,J,k)$ terms are determined using the above definition of the error term, the structure of the ANN described in Section 6.2, and the definition of $\delta(i,J,k)$ given in Eq. (5.7).

The $\delta(i,J,k)$ expressions for the nodes of different layers are provided in the following

Layer 8

$$\delta(8,1,k) = T(k) - \mathrm{var}(8,1,k) \tag{7.2}$$

Layer 7

$$\delta(7,J,k) = \delta(8,1,k)w(7,J,1) \tag{7.3}$$

Layer 6

As explained earlier, this layer is a dummy layer used for clarity of representation. Hence, the $\delta(6,J,k)$ values are not calculated.

Layer 5

$$\delta(5,J,k) = \frac{1}{\sum_{\forall J} \mathrm{var}(5,J,k)}\{T(k) - \mathrm{var}(8,1,k)\}\{w(7,J,1) - \mathrm{var}(8,1,k)\} \tag{7.4}$$

In order to obtain this expression, the error term of Eq. (7.1) is directly written in terms of $\mathrm{var}(5,J,k)$ and $w(7,J,1)$. For a detailed derivation of the above term one may refer to Chakroborty (1993).

Layer 4

$$\delta(4,J,k) = \sum_{\forall K \in \Psi} \delta(5,K,k)\frac{\partial}{\partial \mathrm{var}(4,J,k)} m(\mathrm{var}(4,J,k), \mathrm{var}(4,F,k), \mathrm{var}(4,G,k)) \tag{7.5}$$

The set Ψ refers to the set of nodes of the fifth layer to which the Jth node of the fourth layer is connected. The node numbers F and G refer to the other two nodes (other than the Jth node) of the fourth layer which also send their outputs to the Kth node of the fifth layer. Note that given a particular J, the set Ψ can be obtained from Table 2. Further, for a given $K \in \Psi$, the F and G nodes can also be obtained from either Table 2 or Fig. 7.

Layer 3

$$\delta(3,J,k) = \begin{cases} \mathrm{var}(3,J,k)(1 - \mathrm{var}(3,J,k))\delta(4,K,k) & \text{for } J = 1,2,4,5,6,8,9,10,12,14 \\ -\mathrm{var}(3,J,k)(1 - \mathrm{var}(3,J,k))\delta(4,K,k) & \text{for } J = 3,7,11,13 \end{cases} \tag{7.6}$$

In the above expression the value of K (the node number of Layer 4 to which the Jth node of Layer 3 connects) for a given J can be obtained from Table 2.

Layer 2

$$\delta(2,J,k) = \delta(3,J,k)w(2,J,J) \tag{7.7}$$

The above description of $\delta(i,J,k)$ terms provide the necessary groundwork for developing the weight modifiers, $\Delta w(i,J,K)$. Note, only the weight modifiers for the weights of arcs between

Layers 1 and 2, Layers 2 and 3, and Layers 7 and 8 are computed and presented because all the other weights in the ANN representation are fixed (refer to Table 2). Recall that the weights being modified are the ones which represent the parameters of the membership functions of the fuzzy sets of the FIS.

7.1.1. Weight modifiers for weights from Layer 1 to Layer 2: $\Delta w(1,J,K)$

From Eq. (5.5) and the expression for $\mathrm{net}(2,K,k)$ (which can be obtained based on the descriptions in the Sections 6.1 and 6.2), the expression for $\Delta w(1,J,K)$ can be obtained.

$$\Delta w(1,J,K) = \eta\delta(2,K,k)\mathrm{var}(1,J,k) \tag{7.8}$$

Further, note that only the weights emanating from Nodes 1, 3, and 5 of Layer 1 are modified, because the rest are fixed. For the Nodes 1, 3, and 5, the value of $\mathrm{var}(1,J,k)$ is always one. Hence, the above expression can be simplified to the following:

$$\Delta w(1,J,K) = \eta\delta(2,K,k) \tag{7.9}$$

7.1.2. Weight modifiers for weights from Layer 2 to Layer 3: $\Delta w(2,J,J)$

From Eq. (5.5) and the expression for $\mathrm{net}(3,J,k)$ (which can be obtained based on the descriptions in the Sections 6.1 and 6.2), the expression for $\Delta w(2,J,J)$ can be obtained.

$$\Delta w(2,J,J) = \eta\delta(3,J,k)\mathrm{var}(2,J,k) \tag{7.10}$$

7.1.3. Weight modifiers for weights from Layer 7 to Layer 8: $\Delta w(7,J,1)$

From Eq. (5.5) and the expression for $\mathrm{net}(8,1,k)$ (which can be obtained based on the descriptions in the Sections 6.1 and 6.2), the expression for $\Delta w(7,J,1)$ can be obtained.

$$\Delta w(7,J,1) = \eta\delta(8,1,k)\mathrm{var}(7,J,k) \tag{7.11}$$

7.2. Implementation of the bases for modification of the FIS

This section shows how the bases for modifying the FIS parameters, which was developed in Section 4.2. Before proceeding further, recall (i) there are five bases (as explained in Section 4.2), and (ii) the following correspondence between the notation used in Section 4.2 and this section: the variables ρ^r, $\rho_2(k)$, and $w_r(k)$ are $w(7,J,1)$, $\mathrm{var}(8,1,k)$, and $\mathrm{var}(5,J,k)$, respectively.

7.2.1. Implementation of Basis 1(b) and Basis 2

Eq. (7.11) shows that the sign of $\Delta w(7,J,1)$ is the same as $\delta(8,1,k)$ since $\mathrm{var}(7,J,k)$ is always non-negative. Therefore, the following can be stated:

When $T(k) > \rho_2(k)$, then $\delta(8,1,k)$ is positive, and therefore $\Delta w(7,J,1)$ is also positive. Hence, $w(7,J,1)$ (or ρ^r) is increased (thus implementing Basis 2).

When $T(k) < \rho_2(k)$, then $\delta(8,1,k)$ is negative, and therefore $\Delta w(7,J,1)$ is also negative. Hence, $w(7,J,1)$ (or ρ^r) is decreased (thus implementing Basis 2).

Further, when a rule does not fire (i.e., when $\mathrm{var}(7,J,k) = \mathrm{var}(5,J,k) = 0$), then $\Delta w(7,J,1) = 0$ and therefore the weights $w(7,J,1)$ (or ρ^r's) are left unchanged (thus implementing Basis 1(b)).

7.2.2. Implementation of Basis 1(a), Basis 3, Basis 4 and Basis 5

Recall, the expression for $\delta(4, J, k)$ given in Eq. (7.5) and the fact that the derivative of $m(v_1, v_2, v_3)$ with respect to v_1 takes a value of 0 if v_1 is not the minimum, 1 if it is the minimum, and between 0 and 1 if v_1 is very close to the minimum of v_2 and v_3.

Therefore, it can be stated that if the truth of a particular Rule r (obtained from, say, Node K of Layer 5) is determined by the membership function of Node J of Layer 4, then the whole of $\delta(5, K, k)$ is assigned to $\delta(4, J, k)$. Similarly, if the truth value is not determined by the membership function of Node J of Layer 4, then none of $\delta(5, K, k)$ is assigned to $\delta(4, J, k)$. Only when it becomes difficult to ascertain whether Node J is the sole determinant of the truth value (for example, when membership values of more than one fuzzy set of the antecedent are very close or equal to the truth value) then a part of $\delta(5, K, k)$ is assigned to $\delta(4, J, k)$ (thus implementing Basis 5).

Also, note that the summation in Eq. (7.5) captures the fact that the same membership function (fuzzy set) appears in many rules and therefore the net effect of Node J should be obtained by adding all the contributions of this node to the error via its presence in many rules.

Further, from (i) the sign of the part of $\delta(4, J, k)$ which comes from Node K of Layer 5 is the same as that of $\delta(5, K, k)$, (ii) the description of $\mathrm{var}(2, J, k)$, and (iii) Eqs. (7.6), (7.7), (7.9) and (7.10), it can be shown that:

- If $\delta(4, J, k) > 0$ and the membership grade of the input is higher than 0.5, then the slope of the membership function is made steeper (so as to increase the membership grade). If $\delta(4, J, k) > 0$ and the membership grade is less than 0.5, then the slope is made milder (so as to increase the membership grade).
- If $\delta(4, J, k) > 0$, then the crossover point (of the membership function) is shifted to the left or to the right so as to increase the membership grade of the input. The direction of shift depends on which half of the membership function is pertinent to the input.
- If $\delta(4, J, k) < 0$ then the opposite of the above actions are taken.
- The slope and the crossover point of only the relevant half (i.e., the half in which the given input lies) of the membership function of a bounded fuzzy set is modified. This is true because (for bounded fuzzy sets), either $\delta(3, J, k)$ or $\delta(3, J+1, k)$ is 0.

These actions together with the fact that the sign of $\delta(5, J, k)$ is determined (recall Eq. (7.4)) according to the conditions stated in Basis 3 and Basis 4 assure that these bases are correctly implemented. One additional condition on $\delta(5, J, k)$, that of setting it equal to zero whenever $\mathrm{var}(5, J, k) = 0$, ensures that Basis 1(a) is also implemented. For a detailed proof of the above one may refer to Chakroborty (1993).

8. Implementation of the entire calibration process

This section gives an overview of the implementation of the calibration procedure for practical application. Fig. 8 gives the flowchart of the calibration process. As can be seen from the flowchart, the calibration process is iterative. In each iteration (in the figure, counter "I", represents this iteration) all the input–output vectors are presented one at a time. For each of the input–output vectors all relevant quantities of the ANN are calculated, the error is computed, the

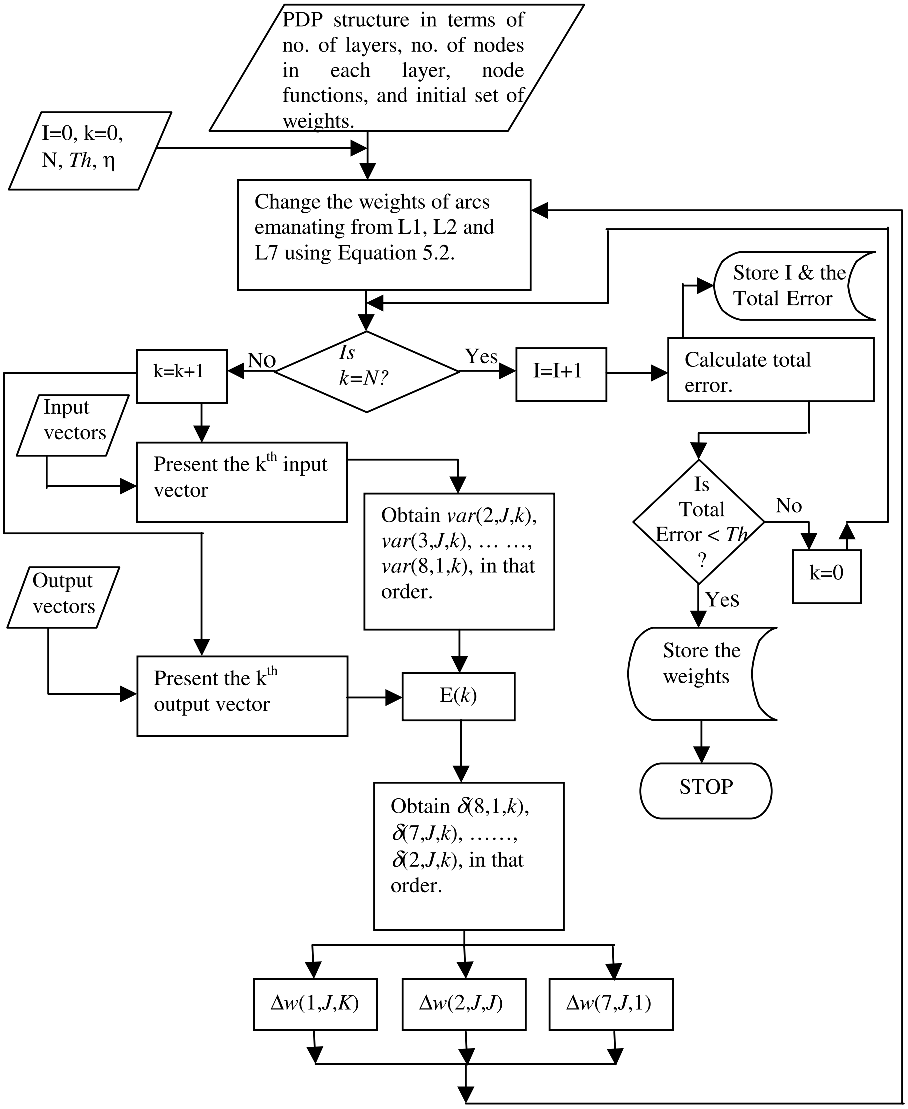


Fig. 8. Flowchart showing the implementation of the calibration process.

weight modifiers are determined, and the relevant weights are modified. Once all the input–output vectors have been presented, and the weights modified, the total error (i.e., the sum of the errors from each of the input–output vectors) is calculated. If this error, is less than a pre-defined threshold value, Th, then the calibration process is stopped; otherwise, the next iteration of the calibration process starts with the presentation of all the input–output vectors once again. Note that, η of Eqs. (7.9)–(7.11) is kept small so that the system does not over-correct based on the target from a single input–output vector and thereby create instability in the calibration procedure.

9. Examples

In this section, the proposed method is applied to calibrate the membership functions of a FIS that performs the complex control task of car-following. The model was developed by the authors Kikuchi and Chakroborty (1992), Chakroborty (1993), and Chakroborty and Kikuchi (1999). The situation being modeled using the FIS is as follows; a vehicle pair traveling in the same direction (they are called the leading vehicle (LV) and the vehicle following (FV)) is considered, the action (in terms of acceleration and deceleration) of the FV is predicted based on the relative speed between the two vehicles, the current speed of the FV, the distance between the LV and the FV, and the actions of the LV.

The inputs to the model are the relative speed between FV (following vehicle) and LV (the leading vehicle), the speed of FV, the distance between LV and FV, and the actions of LV, at an earlier time. The output of the model is the action of the following vehicle (in terms of acceleration or deceleration) at the current time. The model is in the "If the relative speed is $\tilde{A}$, and the speed of FV is $\tilde{B}$, and the distance between LV and FV is $\tilde{C}$, then the action of FV is $\tilde{D}$", where $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$ are given in fuzzy sets. Two of the antecedent variables have 6 fuzzy classes (sets) each, and one has 11 fuzzy classes, whose definition depends on the condition of the factors that change with the driving condition. There are a total of 396 rules ($= 6 \times 6 \times 11$). The parameters of the membership functions of the fuzzy sets, total 23 ($= 6 + 6 + 11$) are calibrated using the proposed process.

Real world data on the inputs and the output are collected using an instrumented vehicle capable of measuring the headway between LV and FV, speeds of LV and FV, and acceleration (deceleration) of LV and FV at intervals of 20 ms. The input conditions obtained from the real world data are presented to the FIS model; it then predicts the actions (acceleration/deceleration) of the FV. The mismatch between the predicted and the actual actions (obtained from the real world data) is used to calibrate the parameters of the membership functions used in the FIS.

In the rest of this section, the predicted outputs before-calibration and after-calibration are presented. Four cases are presented here. The first three cases present the results from the calibration procedure using three different drivers (called Subjects 1, 2 and 3). The fourth case presents results using data from various driving runs with one driver. The changes in the membership functions after calibration are also shown for the fourth case. Many other cases were tested but only a sample set is shown here.

9.1. Case I

Fig. 9 plots the observed actions of the FV (in terms of acceleration or deceleration) and the predicted actions (obtained using the FIS) at every second. In Fig. 9(a) the predicted actions are from the FIS before calibration and in Fig. 9(b) the predicted actions are from the FIS after calibration. The predictions from the FIS, is drawn as a band and the observed actions are plotted as points. The band represents plus–minus 1 ft/s^2 to the predicted modal value of the actions of FV. Because, in the calibration process presented in this example, only the modal value of the fuzzy sets in the consequents are corrected, their supports (or the bases of the membership functions) do not change with correction; hence, the width of the band in the figure remains the same before and after calibration.

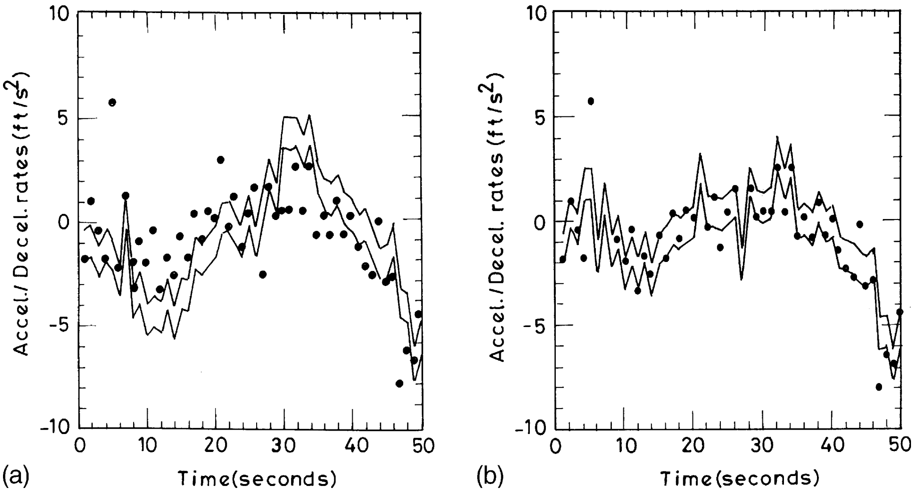


Fig. 9. Predicted pattern of actions of the FV (Subject 1, Experiment 1): (a) before calibration, (b) after calibration.

As seen in the figure, before calibration most of the observed values lay outside the predicted band, after calibration, however, the predicted band houses most of the observed values. This illustrates that the calibration process is able to modify the membership functions of the FIS appropriately.

9.2. Case II

Fig. 10 is similar to that of Fig. 9, but the data used here is obtained using a driver different from the one used in Case I. Fig. 10(a) shows the predicted actions before calibration and Fig. 10(b) shows the predicted actions after calibration. The same observations as in Case I are made here.

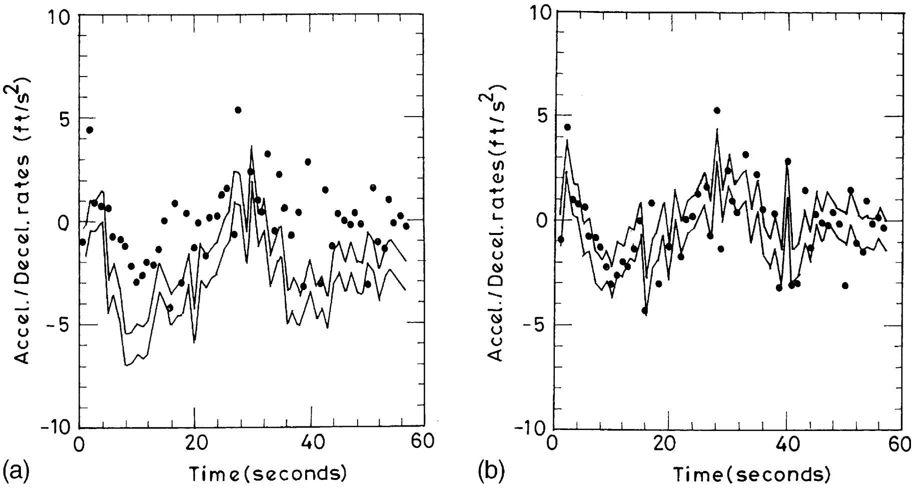


Fig. 10. Predicted pattern of actions of the FV (Subject 2, Experiment 1): (a) before calibration, (b) after calibration.

9.3. Case III

Fig. 11 is similar to that of Figs. 9 and 10, but the data used here is obtained using a third driver. Fig. 11(a) shows the predicted actions before calibration and Fig. 11(b) shows the predicted actions after calibration. The same observations as in Cases I and II are made here.

9.4. Case IV

This case uses the data from various driving runs by one driver. The data is divided into two parts, the first part is used to calibrate the model and the second part is to check the performance

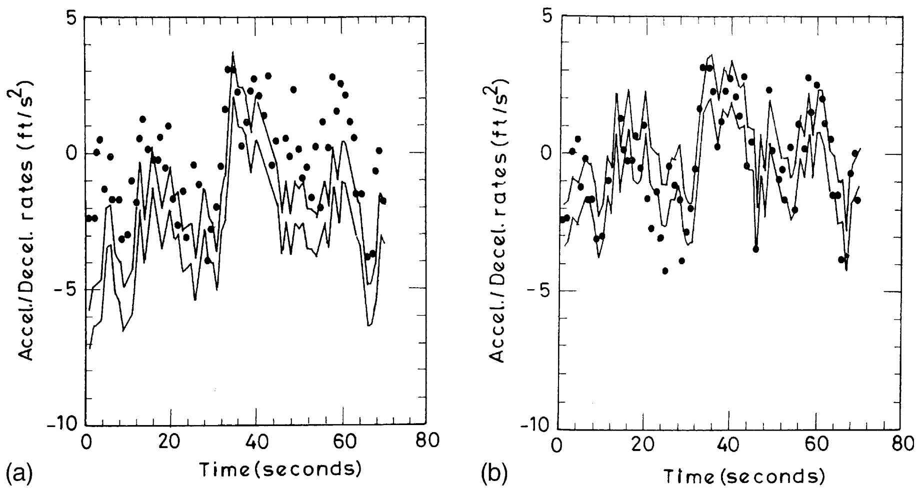


Fig. 11. Predicted pattern of actions of the FV (Subject 3, Experiment 1): (a) before calibration, (b) after calibration.

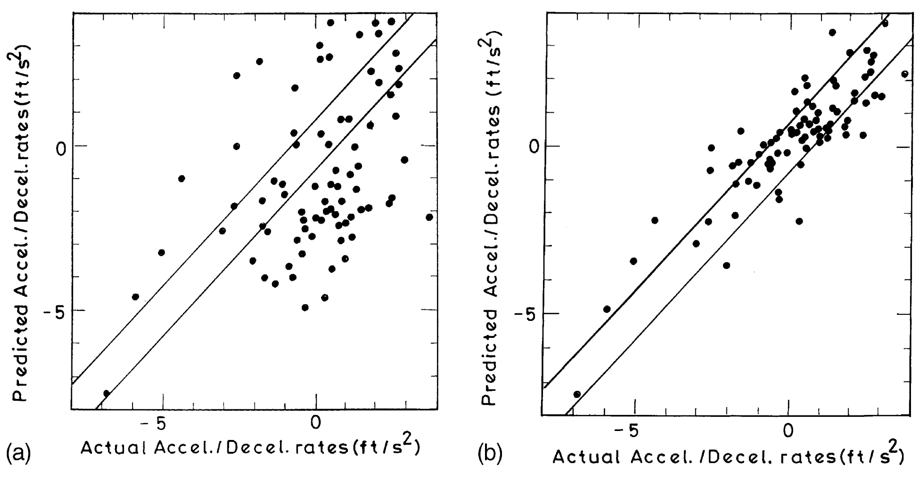


Fig. 12. Predicted actions of the FV (Subject 1, training set): (a) before calibration, (b) after calibration.

of the calibrated FIS model. Fig. 12 shows the comparison between the observed and predicted actions, before and after calibration of the FIS. In Fig. 12(a) (where the predicted values are from the FIS before calibration) the observed values lay outside the 45° band. However, in Fig. 12(b) (where the predicted values are from the FIS after calibration) the band houses most of the observed values. This again shows that the calibration procedure modifies the membership functions of the FIS appropriately.

The second part of the data is used to study how well the calibrated FIS model can predict the actions of the FV, even when the data has not been used in the calibration process. Fig. 13 highlights the fact that once the FIS model is calibrated, it can predict the actions of the FV

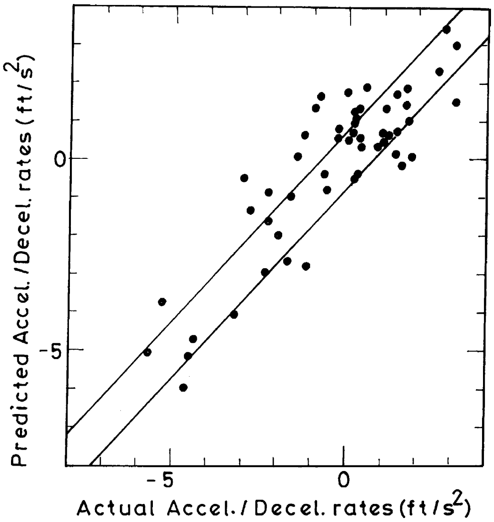


Fig. 13. Predicted actions of the FV (Subject 1, checking set).

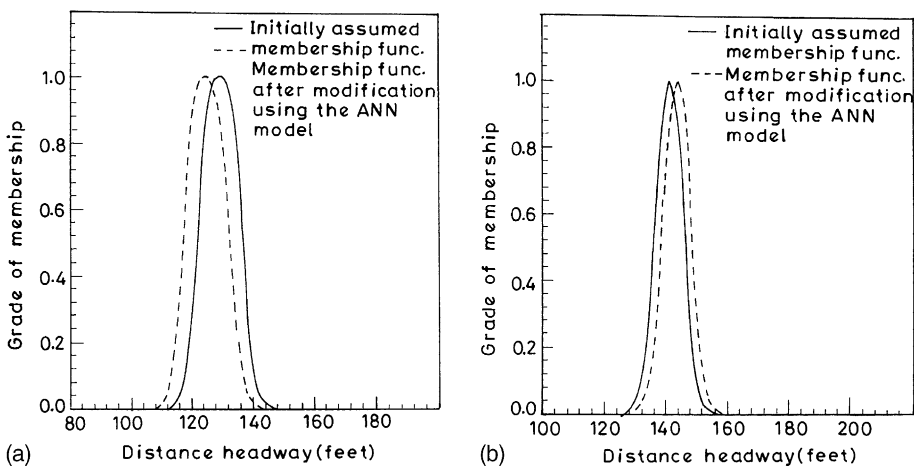


Fig. 14. Two examples of modified membership functions for categories of distance headway (Subject 1).

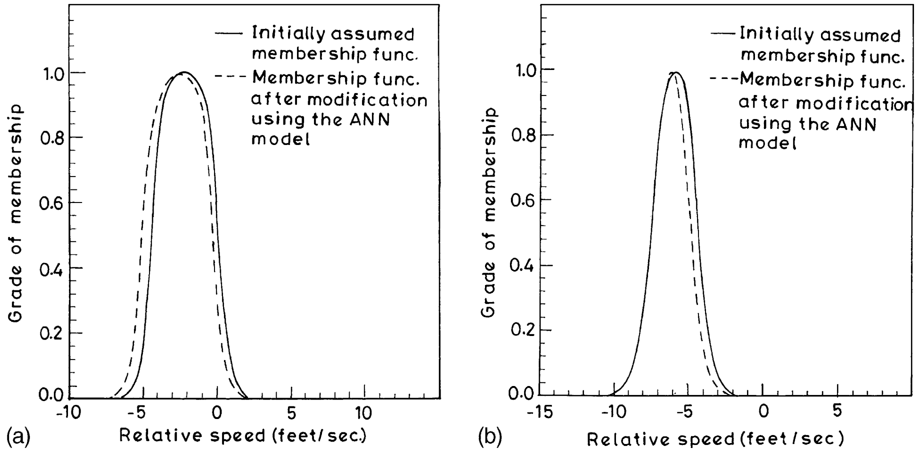


Fig. 15. Two examples of modified membership functions for categories of relative speed (Subject 1).

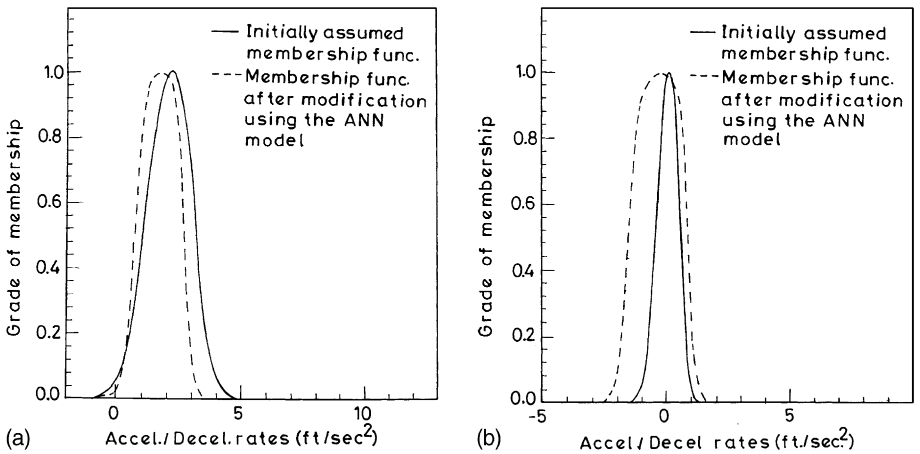


Fig. 16. Two examples of modified membership functions for categories of acceleration/deceleration rate of LV (Subject 1).

accurately. Most of the points in the figure lie within the 45° band although none of these observed actions are used in the calibration process. This shows that the calibration process makes meaningful changes to the model, rather than changes that merely allow the model to follow the data used for calibration more closely.

Figs. 14–17 are presented to show, as an example, the type of changes the proposed procedure has made to the initial fuzzy sets of in the antecedents and the consequents of the rules by the calibration process. Each of the first three figures, shows the examples of the changes made to some of the fuzzy sets of the antecedents during calibration.

Fig. 17 shows the changes made to the modal value of the consequents during calibration. The line in Fig. 17 is a 45° line; points above the line represent the consequents which are increased and points below the line represent consequents which are decreased. For the scenario presented in

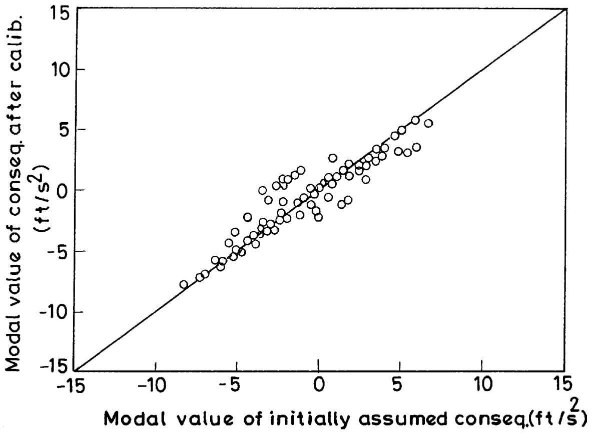


Fig. 17. Modified modal values of the consequences (Subject 1).

this figure, a total of 97 rules fired and therefore the consequents of these rules are the ones that have been modified. Upon calibration, on an average, the consequents which suggested acceleration are decreased by 0.57 ft/s^2; however, the consequents which suggested deceleration are increased (i.e., to a lesser deceleration rate) by 0.56 ft/s^2. Such changes in the consequents are expected because before calibration, as can be seen from Fig. 12(a), the predicted acceleration/deceleration rates were generally more than the actual actions of the FV.

10. Conclusions

This paper has developed a procedure that calibrates the membership functions of the fuzzy sets used in a FIS. The inference system is transformed into an ANN format; in the network, the shape and placement of the membership functions are controlled by the weights of the connections at the nodes. Using the generalized delta rule based back-propagation correction mechanism, the weights are corrected repeatedly in light of a set of input and output data. The final weight values then represent the calibrated parameters of the membership functions. The procedure was tested to calibrate the membership functions used in a fuzzy inference model that represents the driver's control behavior under car-following.

The paper shows that the membership function of a FIS can be logically and systematically calibrated. Many problems in transportation are conducive to fuzzy inference modeling due to the approximate reasoning nature of human behavior such as choice process and driving behavior. Hence, we hope that what is often perceived as the arbitrary nature of membership calibration, can no longer be the main deterrence for applying FIS to modeling transportation phenomena. This procedure will add credibility to the fuzzy inference approach when modeling systems that involve approximate stimulus–response or cause–effect relationships. The procedure tunes the membership functions of the fuzzy sets logically based on the real world data. We plan to apply the FIS to represent causalities involving in the elements of a large-scale civil infrastructure systems, such as a large port, airport, or urban transport system.

References

Abe, S., Lan, M.-S., 1995. A method for fuzzy rules extraction directly from numerical data and its application to pattern classification. IEEE Transactions on Fuzzy Systems 3 (1), 18–28.

Baldwin, J.F., Pillsworth, B.W., 1980. Axiomatic approach to implication for approximate reasoning with fuzzy logic. Fuzzy Sets and Systems 3, 193–219.

Chakroborty, P., 1993. A model of car-following: a fuzzy-inference-based system. Ph.D. Dissertation, University of Delaware, Newark.

Chakroborty, P., Kikuchi, S., 1999. Car-following behavior: a review of existing models and proposed model. Transportation Research, Part C 7, 209–235.

Dijker, T., Hoogendoorn, S.P., 1998. Perspective of Fuzzy Logic in Traffic Engineering: Reviews and Annotated Bibliography. Ministry of Transport, Public Works and Water Management, The Netherlands.

Hecht-Nielsen, R., 1990. Neurocomputing. Addison Wesley Publishing Company, Massachusetts.

Homaifar, A., McCormick, E., 1995. Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. IEEE Transactions on Fuzzy Systems 3 (1), 129–139.

Hoogendoorn, S.P., Copinga, G., Kaymak, U., 1998. Perspective of Fuzzy Logic in Traffic Engineering: Main Document. Ministry of Transport, Public Works and Water Management, The Netherlands.

Horikawa, S., Furuhashi, T., Uchikawa, Y., Tagawa, T., 1991. A study on fuzzy modelling using fuzzy neural networks. Proceedings of the International Fuzzy Engineering Symposium, 562–573.

Kikuchi, S., Chakroborty, P., 1992. Car-following Model based on Fuzzy Inference System. Transportation Research Record (1365), 82–91.

Klir, G.J., Folger, T.A., 1988. Fuzzy Sets, Uncertainty and Information. Prentice Hall, Englewood Cliffs, New Jersey.

Pappis, C., Mamdani, E., 1977. A fuzzy logic controller for a traffic junction. IEEE Transactions on Systems, Man, and Cybernetics SMC-7, 707–717.

Rumelhart, D.E., McClelland, J.L., the PDP Research Group, 1986a. In: Parallel Distributed Processing: Explorations into the Microstructure of Cognition, vol. 1. The MIT Press, Cambridge, MA.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986b. In: Learning Internal Representation by Error Propagation Parallel Distributed Processing: Explorations into the Microstructure of Cognition, vol. 1. The MIT Press, Cambridge, MA, pp. 318–362.

Sugeno, M., 1989. In: Fuzzy Control. Nikkan Kogyo Shimbun, Tokyo, pp. 128–132.

Teodorovic, D., Vukadinovic, K., 1998. Traffic Control and Transport Planning: A Fuzzy Sets and Neural Network Approach. Kluwer Academic Publishers, Boston.

Wang, L.-X., Mendel, J., 1992a. Generating fuzzy rules by learning from examples. IEEE Transactions on systems, Man and Cybernetics 22, 1414–1427.

Wang, L.-X., Mendel, J., 1992b. Back-propagation of fuzzy systems as nonlinear dynamic system identifiers. Proceedings IEEE International Conference on Fuzzy Systems, San Diego. pp. 807–813.

Wang, L.-X., Mendel, J., 1992c. Fuzzy basis functions, universal approximation, and orthogonal least squares learning. IEEE Transactions on Neural Networks 3, 807–813.

Wasserman, P.D., 1989. Neural Computing: Theory and Practice. Van Nostrand Reinhold, New York.

Williams, R.J., 1986. In: The Logic of Activation Functions Parallel Distributed Processing: Explorations into the Microstructure of Cognition, vol. 1. The MIT Press, Cambridge, MA, pp. 423–443.

Zimmermann, H.-J., 1996. Fuzzy Set Theory and Its Applications, Third Ed. Kluwer Academic Publishers, Boston.