that their differences weighs more than their similarities. Of course, it depends on the reference class of properties to be compared . . . .

ACKNOWLEDGMENT

The author thank J. Logan and D. Elias for their comments, and the anonymous referees.

REFERENCES

[1] L. Barsalou, "The instability of graded structure: Implications for the nature of concepts," U. Neisser, Ed. *Concepts and Conceptual Development: Ecological and Intellectual Factors in Categorization.* Cambridge: Cambridge Univ. Press, 1987, ch. 5, pp. 101–140.
[2] R. Bellman and M. Giertz, "On the analytic formalism of the theory of fuzzy sets," *Inform. Sci.,* vol. 5, pp. 149–157, 1973.
[3] P. Cheeseman, "Probabilistic versus fuzzy reasoning," in *Uncertainty in Artificial Intelligence,* L. Kanal and J. Lemmer, Eds. Amsterdam: North-Holland, 1986, pp. 85–102.
[4] D. Cruse, *Lexical Semantics.* Cambridge: Cambridge Univ. Press, 1986.
[5] D. Dubois and H. Prade, *Fuzzy Sets and Systems.* New York: Academic Press, 1980.
[6] G. Fuhrmann, "Note on the integration of prototype theory and fuzzy-set theory," *Synthese,* vol. 86, pp. 1–27, 1991.
[7] B. Gaines, "Fuzzy and probability uncertainty logics," *Inform. Contr.,* vol. 38, pp. 154–169, 1978.
[8] J. Kamp, "Two theories about adjectives," in *Formal Semantics of Natural Language,* E. Keenan, Ed. Cambridge: Cambridge Univ. Press, 1975, pp. 123–155.
[9] D. McNeill and P. Freiberger, *Fuzzy Logic.* New York: Simon & Schuster, 1993.
[10] D. Medin and M. Schaffer, "A context theory of classification learning," *Psycholog. Rev.,* vol. 85, pp. 207–328, 1978.
[11] R. Nosofsky, "Typicality in logically defined categories: Exemplar-similarity versus rule instantiation," *Memory Cogn.,* vol. 17, pp. 444–458, 1991.
[12] G. Oden, "Fuzziness in semantic memory: choosing exemplars of subjective categories," *Memory Cogn.,*, vol. 5, pp. 198–204, 1977.
[13] ——, "Integration of fuzzy logical information," in *J. Exper. Psych.: Human Perc. Perf.,* vol. 3, pp. 565–575, 1977.
[14] D. Osherson and E. Smith, "On the adequacy of prototype theory as a theory of concepts," *Cognition,* vol. 9, pp. 35–58, 1981.
[15] F. Palmer, *Semantics,* 2nd ed. New York: Cambridge University Press, 1981.
[16] E. Rosch, "On the internal structure of perceptual and semantic categories," in *Cognitive Development and the Acquisition of Language,* T. Moore, Ed. New York: Academic, 1973, pp. 111–144.
[17] E. Rosch and C. Mervis, "Family resemblances: Studies in the internal structure of categories," *Cogn. Psychol.,* vol. 7, pp. 573–605, 1975.
[18] E. Sapir, "Grading: A study in semantics," *Philosophy Sci.,* vol. 11, pp. 93–116, 1944.
[19] P. Smets, "Varieties of ignorance and the need for well-founded theories," *Inform. Sci.,* vol. 57–58, pp. 135–144, 1991.
[20] E. Smith and D. Osherson, "Conceptual combination with prototype concepts," *Cogn. Sci.,* vol. 8, pp. 337–361, 1984.
[21] I. Turksen, "Measurement of membership functions and their acquisition," *Fuzzy Sets Syst.,* vol. 40, pp. 5–38, 1991.
[22] A. Tversky, "Features of similarity," *Psycholog. Rev.,* vol. 84, pp. 327–352, 1977.
[23] A. Tversky and D. Kahneman, "Judgment under uncertainty: Heuristics and biases," *Science,* vol. 185, pp. 1124–1131, 1974.
[24] P. Wang, "Belief revision in probability theory," in *Proc. Ninth Conf. Uncertainty in Artificial Intelligence.* San Mateo, CA: Morgan Kaufmann, 1993, pp. 519–526.
[25] ——, "Non-axiomatic reasoning system (version 2.2)," Center for Research on Concepts and Cognition, Indiana University, Bloomington, 1993, Tech. Rep. 75, available via WWW at http://www.cogsci.indiana.edu/farg/pwang_papers.html.
[26] ——, "From inheritance relation to nonaxiomatic logic," *Int. J. Approx. Reas.,* vol. 11, pp. 281–319, 1994.
[27] L. Zadeh, "A fuzzy-set-theoretic interpretation of linguistic hedges," *J. Cybern.,* vol. 2, pp. 4–34, 1972.
[28] ——, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.,* vol. 3, pp. 28–44, 1973.
[29] ——, "The concept of a linguistic variable and its application to approximate reasoning," *Inform. Sci.,* vol. 8, pp. 199–249 and 301–357, vol. 9, pp. 43–80, 1975.
[30] ——, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets Syst.,* vol. 1, pp. 3–28, 1978.
[31] ——, "A theory of approximate reasoning," in *Machine Intelligence,* J. Hayes, D. Michie, and L. Mikulich, Eds. New York: Halstead , 1979, vol. 9, pp. 149–194.
[32] ——, "A note on prototype theory and fuzzy sets," *Cognition,* vol. 12, pp. 291–297, 1982.
[33] ——, "Is probability theory sufficient for dealing with uncertainty in AI: A negative view," in *Uncertainty in Artificial Intelligence,* L. Kanal and J. Lemmer, Eds. Amsterdam: North-Holland, 1986, pp. 103–116.

Learning Fuzzy Inference Systems Using an Adaptive Membership Function Scheme

A. Lotfi and A. C. Tsoi

***Abstract*—An adaptive membership function scheme for general additive fuzzy systems is proposed in this paper. The proposed scheme can adapt a proper membership function for any nonlinear input-output mapping, based upon a minimum number of rules and an initial approximate membership function. This parameter adjustment procedure is performed by computing the error between the actual and the desired decision surface. Using the proposed adaptive scheme for fuzzy system, the number of rules can be minimized. Nonlinear function approximation and truck backer-upper control system are employed to demonstrate the viability of the proposed method.**

I. INTRODUCTION

Fuzzy inference systems have found many applications in recent years. The simplicity of the design procedure of such systems is a dominant attraction in various industrial as well as household products. In most cases, the design of a fuzzy inference system is related to the ways in which an expert or a skilled human operator would operate in that special domain. Among the various successful applications of fuzzy inference systems we can mention are the application of fuzzy theory in the subway system in the city of Sendai, Japan [13]; the detection of load and control of the washing cycle of a washing machine, the automatic focusing of the video camera and nuclear reactor control [1].

Despite the brisk and stimulating promotion of fuzzy theory [14] from academic research to production line, there is still a lack of a fuzzy system theory for the study of fuzzy inference systems. However, some attempts have recently been made [4]. Techniques which have been successfully applied in particular domain may not be applied to problems arising from another domain. Therefore a general design method is required. To move one step in this direction, an

Manuscript received December 12, 1993; revised March 30, 1995.
The authors are with the Department of Electrical and Computer Engineering, University of Queensland, Brisbane 4072, Australia (lotfia@elec.uq.oz.au; act@elec.uq.oz.au).
Publisher Item Identifier S 1083-4419(96)02299-6.

Adaptive Membership Function Scheme (AMFS) for *Fuzzy Inference Systems (FIS)* is proposed in this paper.

The first attempt to provide a general theory for the "realization"[1] of a fuzzy inference system was proposed by Jang [2] who introduced Generalized Neural Networks based fuzzy inference systems. The network was able to adjust its parameters such that the error between the desired and the actual decision surface was gradually reduced. The fuzzy inference method used was based on a specific type of fuzzy inference system introduced earlier by Takagi and Sugeno [11]. The consequent premise of each rule is assumed to be a crisp value rather than a fuzzy value. Our proposed scheme is to employ the same type of network except that we will use a general additive fuzzy system which has already proposed by Pacini and Kosko [10].

Using the proposed adaptive scheme for fuzzy systems, fewer rules are required to correspond the expert exemplar or expert knowledge to the fuzzy system.

The structure of this paper is as follows: first the fuzzy inference systems will be introduced, followed by the description of an adaptive membership function scheme and rule minimization. Application to nonlinear function approximation and the control system for backing-up trucks are employed to illustrate the proposed scheme. Pertinent conclusions will be drawn from the implications of the proposed method.

II. Fuzzy Inference Systems

A crucial step in the design of FIS[2] is determination of appropriate knowledge-base parameters. A knowledge base consists of three major sub-systems which can be varied in the design of a FIS [5]. These three major sub-systems are:

- A data-base containing membership function of linguistic values for both the antecedent and the consequent
- The fuzzy reasoning mechanism
- The number of rules used in the fuzzy rule-base

A general inference mechanism, which is often used in the representation of human reasoning, can be represented as follows:

$$\begin{array}{llllll}
P^1: & \text{If} & \mathrm{X\,is\,}\mathcal{A}^1 & \text{then} & \mathrm{Y\,is\,}\mathcal{B}^1, & \text{else} \\
\cdots & \cdots & \cdots\cdots & \cdots & \cdots\cdots & \cdots \\
P^i: & \text{If} & \mathrm{X\,is\,}\mathcal{A}^i & \text{then} & \mathrm{Y\,is\,}\mathcal{B}^i, & \text{else} \\
\cdots & \cdots & \cdots\cdots & \cdots & \cdots\cdots & \cdots \\
P^n: & \text{If} & \mathrm{X\,is\,}\mathcal{A}^n & \text{then} & \mathrm{Y\,is\,}\mathcal{B}^n, & \text{else} \\
Q: & & \mathrm{X\,is\,}A' & & & \\
\hline
\Delta & & & & \mathrm{Y\,is\,}B' &
\end{array}$$

The antecedent vector $X = [x_1, x_2, \cdots, x_m]$ is an m vector with elements which are linguistic variables in the universe of $U = [U_1, U_2, \cdots, U_m]$. The consequent vector $Y = [y_1, y_2, \cdots, y_k]$ is a k vector with elements which are linguistic variables in the universe of $V = [V_1, V_2, \cdots, V_k]$. Vectors $\mathcal{A}^i = [A_1^i, A_2^i, \cdots, A_j^i, \cdots, A_m^i]$ and $\mathcal{B}^i = [B_1^i, B_2^i, \cdots, B_j^i, \cdots, B_k^i]$ are vectors of linguistic values (linguistic labels) referring to the fuzzy variables X and Y, respectively. Vector $\mathcal{A}'$ is the crisp observation vector and $\mathcal{B}'$ is the crisp conclusion vector. Fuzzy sets correspond to each fuzzy variable can be shown as follows:

$$A_j^i = \{\mu_{A_j^i}(u)/u\} \quad u \in U_j \qquad j = 1, 2, \cdots, m \quad i = 1, 2, \cdots, n \tag{1}$$

[1] The verb "realization" is used here to denote the explicit construction of an implementation of a fuzzy inference system.

[2] In control engineering literature, this may be referred to as a Fuzzy Logic Controller (FLC).

$$B_j^i = \{\mu_{B_j^i}(v)/v\} \quad v \in V_j \qquad j = 1, 2, \cdots, k \quad i = 1, 2, \cdots, n. \tag{2}$$

To use the AMFS, it is desirable that the membership functions employed have a continues first derivative. We have investigated [7] different piecewise continuous membership functions (triangular, trapezoidal, Cauchy and Gaussian) with the Gaussian MF showing the best performance. Therefore, the membership function μ for antecedent and consequent premises in the fuzzy values A_j^i and B_j^i are defined as follows:

$$A_j^i = \exp\left\{-\left[\left(\frac{u-\sigma_{ij}}{\rho_{ij}}\right)^2\right]^{\beta_{ij}}\right\} \qquad j = 1, \cdots, m \quad i = 1, \cdots, n \tag{3}$$

$$B_j^i = \exp\left\{-\left[\left(\frac{v-\hat{\sigma}_{ij}}{\hat{\rho}_{ij}}\right)^2\right]^{\hat{\beta}_{ij}}\right\} \qquad j = 1, \cdots, k \quad i = 1, \cdots, n \tag{4}$$

where σ_{ij}, ρ_{ij}, β_{ij}, $\hat{\sigma}_{ij}$, $\hat{\rho}_{ij}$, $\hat{\beta}_{ij}$ are unknown constant parameters. As will be shown subsequently, these parameters can be adjusted on-line using a gradient decent algorithm. We further assume that the universe of antecedent and consequent i.e., U and V are limited to a specific domain interval, i.e.,

$$\begin{aligned} U_j &= [U_j^- \quad U_j^+], \quad j = 1, \cdots, m; \\ V_j &= [V_j^- \quad V_j^+], \quad j = 1, \cdots, k. \end{aligned} \tag{5}$$

Vector X contains m linguistic variables which are connected together by a "liaison" operator AND. The consequent vector Y comprises of k linguistic variables. It is reasonable to assume that there is no relationship between this linguistic variable y_j and the other linguistic variables y_l, $l \neq j$. Therefore, such an inference might be decomposed into k inferences with antecedent vector X and consequent linguistic variable y_j, $j = 1, 2, \cdots, k$ separately. Without loss of generality we can assume that the consequent premise is just one variable, i.e., $k = 1$, ($\hat{\sigma}_{ij} = \hat{\sigma}_i$, $\hat{\rho}_{ij} = \hat{\rho}_i$, $\hat{\beta}_{ij} = \hat{\beta}_i$).

For making an inference "Y is $\mathcal{B}'$" from a set of rules P and observation Q, different methods of reasoning under different fuzzy implication concepts have been studied e.g., [8]. Since the output of the decision engine should be a crisp value, numerous methods for defuzzification also have been proposed [5]. Among these methods, the *centroid* method has been shown to be more effective. Pacini and Kosko [10] have proven that if correlation product inference determines the output fuzzy values, the global centroid F can be computed from local consequent premise centroids. i.e.,

$$F = \frac{\sum_{i=1}^{n} w_i \overline{C}_i I_i}{\sum_{i=1}^{n} w_i I_i} \tag{6}$$

where w_i, $\overline{C}_i$, and I_i are rule firing weights, local centroid, and area of consequent premise, respectively. Based on our definition for membership functions we have:

$$w_i = \prod_{j=1}^{m} \mu_{A_j^i}(u) \tag{7}$$

$$\overline{C}_i(v) = \frac{1}{I_i(v)} \int_{V-}^{V+} v \exp\left\{-\left[\left(\frac{v-\hat{\sigma}_i}{\hat{\rho}_i}\right)^2\right]^{\hat{\beta}_i}\right\} dv \tag{8}$$

$$I_i(v) = \int_{V-}^{V+} \exp\left\{-\left[\left(\frac{v-\hat{\sigma}_i}{\hat{\rho}_i}\right)^2\right]^{\hat{\beta}_i}\right\} dv. \tag{9}$$

III. Adaptive Membership Function Scheme

There are different approaches for extracting fuzzy if-then rules automatically, based on a fuzzy model of the system [6] or a numerical-fuzzy approach [12]. Still, there is an unknown question regarding the assignment of a membership function to each fuzzy value. It is obvious that altering: a) the membership function of linguistic values, b) fuzzy reasoning mechanisms, or c) the number of rules, will affect the overall input-output mapping. Altering the membership function has a dominant effect on the two other factors [7]. It can be said that for a fixed number of rules and different fuzzy reasoning methods, changing the membership function can achieve the same input-output mapping. Alternatively, for a fixed fuzzy reasoning method we can achieve the same input-output mapping with different number of rules and different membership functions.

Consider the generalized neural network based fuzzy inference system introduced by Jang [2]. This contains a multilayer feedforward network in which each node performs a particular function (node function). There are two types of nodes:

1) the nodes which have fixed parameters (they are called circle node functions by Jang)
2) the nodes which depend on a set of parameters specific to the node (Jang called this type of nodes a square node function)

The performance of the node and consequently the performance of the system changes with altering these parameters.

Our proposed algorithm uses a neural network which contains four layers, with 2 circle and 2 square layers. The first and third layers (containing only square nodes) represent the membership functions given in fuzzy values A_j^i and B^i (in this case $j = 1$), respectively. The other two layers contain only circle nodes. The node function of the second layer is a simple multiplication and the node function of the fourth layer is the actual output of the system governed by (6). Fig. 1 shows the structure of the adaptive network based fuzzy inference system.

The *cost function* for minimizing the error arising from all the square nodes in the first and the third layers, is defined as follows:

$$\begin{aligned} E &= \sum_{p=1}^{P} E^p \\ &= \sum_{p=1}^{P} (T^p - F^p)^2 \end{aligned} \tag{10}$$

where E^p is the square of the difference between the actual F^p and the desired T^p output of the system for the pth training data. We assume the number of exemplars in the training data set is P. The parameters in the first and the third layer/membership functions in the antecedent and the consequent premises are defined as $\Theta_{ij} = [\sigma_{ij}, \rho_{ij}, \beta ij]$, $\hat{\Theta}_i = [\hat{\sigma}_i, \hat{\rho}_i, \hat{\beta}_i]$. To update the parameters, we can use a steepest descent gradient method to minimize the cost function E. The values $\Delta\Theta_{ij}$ and $\Delta\hat{\Theta}_i$ at $(t+1)$th instant, where $\Delta\Theta_{ij}(t) = \Theta_{ij}(t) - \Theta_{ij}(t-1)$, and $\Delta\hat{\Theta}_i(t)$ is defined in a similar fashion. It is given as a function of the values at the tth instant as follows:

$$\Delta\Theta_{ij}(t+1) = -\eta\nabla E_{ij} + \alpha\Delta\Theta_{ij}(t) \tag{11}$$

$$\Delta\hat{\Theta}_i(t+1) = -\eta\nabla\hat{E}_i + \alpha\Delta\hat{\Theta}_i(t) \tag{12}$$

where $\nabla\hat{E}_{ij}$, ∇E_i, and η, are gradients of the parameters and the learning rate, which can be expressed as follows:

$$\nabla E_{ij} = \sum_{p=1}^{P} \nabla E_{ij}^p$$

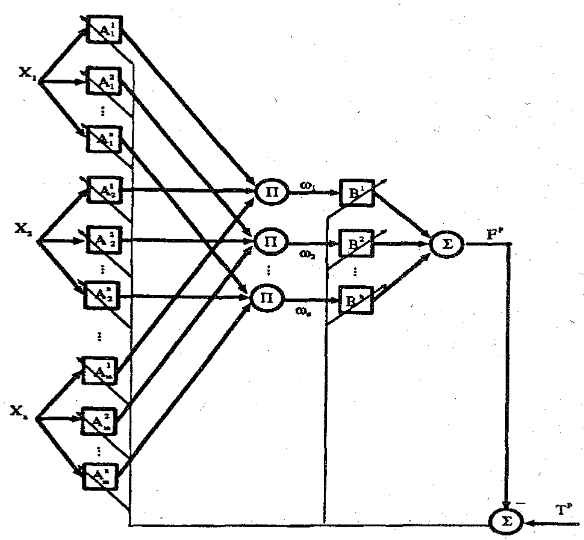


Fig. 1. Adaptive network based fuzzy inference system.

$$\nabla\hat{E}_i = \sum_{p=1}^{P} \nabla\hat{E}_i^p \tag{13}$$

where

$$\begin{aligned} \nabla E_{ij}^p &= \left[\frac{\partial E^p}{\partial\sigma_{ij}}, \frac{\partial E^p}{\partial\rho_{ij}}, \frac{\partial E^p}{\partial\beta_{ij}}\right] \\ \nabla\hat{E}_i^p &= \left[\frac{\partial E^p}{\partial\hat{\sigma}_i}, \frac{\partial E^p}{\partial\hat{\rho}_i}, \frac{\partial E^p}{\partial\hat{\beta}_i}\right] \end{aligned} \tag{14}$$

and

$$\eta = \frac{k}{\sqrt{\sum_{i,j} (\nabla E_{ij})^2 + \sum_i (\nabla\hat{E}_i)^2}} \tag{15}$$

The constant parameter α is the *momentum* of the gradient descent and the constant k is the step size of the gradient descent. The gradients defined in (14) are analytically available (see Appendix) making the presented network realizable.

IV. Rule Minimization

Since acquiring the expert knowledge of a skilled domain specialist in the form of fuzzy value for each fuzzy rule is an arduous step in the design procedures, there should be some methods to determine the proper meaning of related fuzzy values. AMFS gives this opportunity to the controller designer. As long as the parameters of the membership function of fuzzy values are changing, we can obtain the same decision surface with different rules.

Based on our empirical results (these results are shown in Sections V and VI in sequel), for a system without very "spiky" convex nonlinearity (not necessary smooth)[3] the minimum and maximum number of rules can be expressed as follows:

$$2^m \le n \le 3^m \tag{16}$$

Therefore, for a system with two inputs, four to nine rules are sufficient. We can start with a minimum number of rules, and in the case of deficiency, increase it toward the maximum number of rules.

[3] In general, it is difficult to describe exactly what this means in practice. One way in which we can understand this is explained later in Sections V and VI.

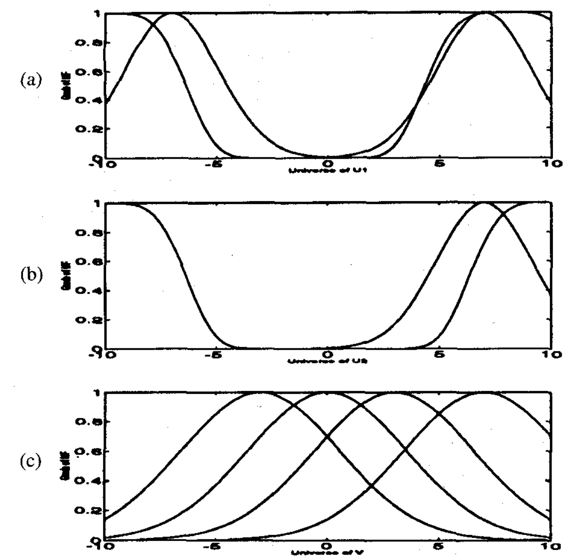


Fig. 2. Initial membership functions with 4 rules for (a) the first input, (b) the second input, and (c) consequent.

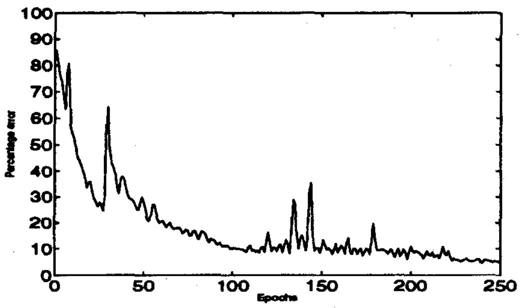


Fig. 3. Error between the target and the trained surface for 250 epochs of training with momentum $\alpha = 0.95$ and step size $\kappa = 0.01$.

There are two types of expert information for justifying membership function of fuzzy values in FIS.

1) *Expert Knowledge:* This is the simplest situation, when some linguistic rules or desired decision surface is accessible. Adjusting membership function of fuzzy values using AMFS would commence with initial membership functions with a minimum number of rules. Rules will be added as required in the design process. The final membership function of fuzzy values, is obtained when the actual decision surface converges to the desired one.
2) *Expert Exemplar:* This is a specific case of (1) and it is more applicable to real systems. A set of input-output pairs of desired system response (expert training) is available. The procedure of training the FIS has been explained earlier. Naturally, the more training exemplars we have the better performance of the overall system will be.

V. Application to Nonlinear Function Approximation

To demonstrate the viability of the proposed method, we use AMFS in this section to approximate a nonlinear function with a set of fuzzy rules. The target nonlinear function which we are going to train our

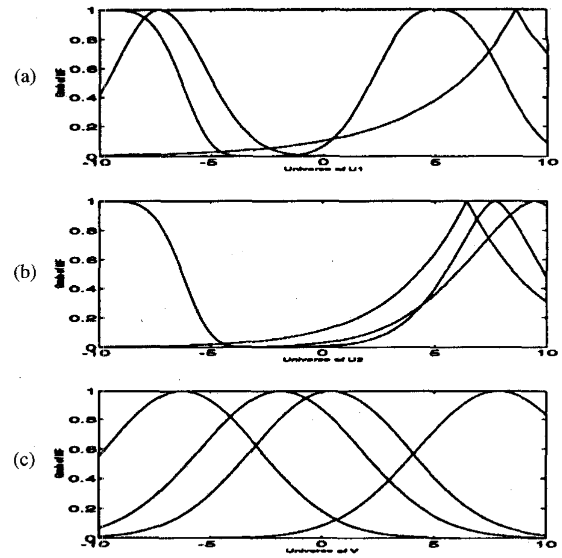


Fig. 4. The membership functions after 250 epochs of training with 4 rules for (a) the first input, (b) the second input, and (c) consequent.

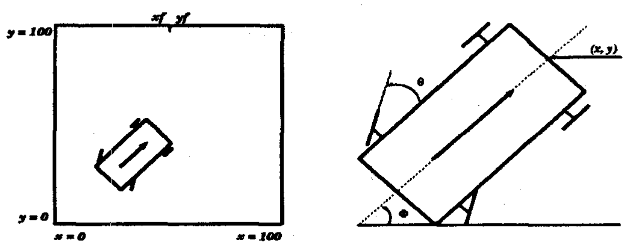


Fig. 5. Diagram of simulated truck and loading zone.

fuzzy system is taken as follows:

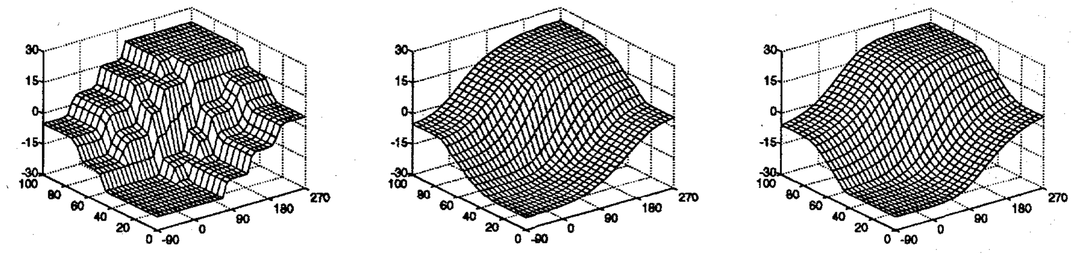
$$T^p = [3e^{x_2/10} - 1]\tanh\left(\frac{x_1}{2}\right) + \frac{2}{30}[4 + e^{x_2/10}]\sin\frac{(x_1+4)\pi}{10} \tag{17}$$

The fuzzy inference system contains 4 rules ($i = 4$) and 2 inputs for antecedent ($j = 2$). The universe U_1 and U_2 are both $[-10 \quad 10]$. Based on some understanding from the desired surface, we can assign initial values for the membership function parameters. With an appropriate combination of the step size and the momentum, the network converges.

The initial membership functions for antecedent and consequent are depicted in Fig. 2. The AMFS has been employed to reduce the error between the desired nonlinear function and fuzzy inference system. The percentage of error is shown in Fig. 3 for 250 epochs of training when the momentum $\alpha = 0$ and step size of gradient descent $\kappa = 0.01$. The membership functions after 250 epochs of training are shown in Fig. 4.

VI. Application to Truck Backer-Upper Control

In the real world, backing a truck to a loading zone is a difficult problem except for a skilled truck driver. If we elicit the skilled driver experience in a fuzzy if-then rule format, and can be assured that the fuzzy controller is working with the same set of rules, we would obtain the same trajectory. For truck backing-up control,



(a) (b) (c)

Fig. 6. Control surface of fuzzy controller after 300 epochs training with: (a) 35 rules, (b) 9 rules, and (c) 4 rules.

Nguyen and Widrow [9] use expert exemplars to train an artificial neural network based controller. Kong and Kosko [3] proposed a fuzzy logic controller with 35 expert rules, and they compared their results obtained from FLC with results achieved by using a neural network controller. FLC has been shown to give more appropriate tracking results. A neural network controller only uses numerical data, whereas FLC employs linguistic rules concluded from expert drivers explicitly. To combine the above two methods, Wang and Mendel [12] utilized numerical-fuzzy approach with almost the same rules as [3], but with different membership functions for fuzzy values.

The simulated truck (which is allowed to move backward only), and the loading zone are depicted in Fig. 5. The truck in our simulation is the cab part of [9] and the same truck for [3] and [12] except for the size of the yard, the definition of steering and the azimuth angle. Since our study is performed in simulation, the dynamics of the truck backing-up system is required. We used the following approximate kinematics [12].

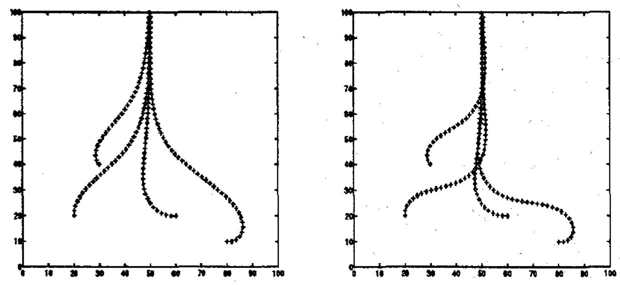
$$\begin{aligned} x(t+1) &= x(t) + v\{\cos[\phi(t)+\theta(t)] + \sin[\theta(t)]\sin[\phi(t)]\} \\ y(t+1) &= y(t) + v\{\sin[\phi(t)+\theta(t)] - \sin[\theta(t)]\cos[\phi(t)]\} \\ \phi(t+1) &= \phi(t) - v\left\{\sin^{-1}\left[\frac{2\sin[\theta(t)]}{\ell}\right]\right\} \end{aligned} \tag{18}$$

where x, y, and ϕ are rear center of truck coordinate and azimuth angle of truck in yard, respectively. They can be considered as state variables of the system which indicate position and direction of the truck in yard at any instant of time. θ is the steering angle to direct the truck to the loading zone x_f and y_f. Constant parameters v and ℓ are truck speed and length of the truck, respectively. The control goal is to steer the truck from any initial position to prespecified loading dock with a right azimuth angle ($\phi_f = 90$) and coincided rear position. The steering angle θ is the control action which is provided by the designed fuzzy controller. Since we presuppose adequate clearance between the truck and the loading dock, state variable y can be abandoned for the reason that it becomes a dependent variable. Therefore, the inputs to the controller are x and ϕ. The range of variables for simulated truck and controller are as follows:

$$\begin{aligned} x &\in [x^- \quad x^+] = [0 \quad 100] \\ \phi &\in [\phi^- \quad \phi^+] = [-90 \quad 270] \\ \theta &\in [\theta^- \quad \theta^+] = [-30 \quad 30]. \end{aligned}$$

The truck speed $v = 5$ and the length of the truck $\ell = 4$. The maximum width of the yard is $y = 100$. Desired loading dock position is $x_f = 50$ and $y_f = 100$. Positive attitude of azimuth angle ϕ is clockwise with respect to the horizontal line. Steering angle θ is positive when the steering wheels rotate counterclockwise.

We start with fuzzy value and rules specified in [3]. There are 35 rules with seven, five, and seven fuzzy values for azimuth angle ϕ, coordinate x, and steering angle θ, respectively. The control surface is depicted in Fig. 6(a).



(a) (b)

Fig. 7. Truck trajectory of fuzzy controller for both expert knowledge and expert exemplars with: (a) 9 rules, and (b) 4 rules.

In the next step, AMFS is used for controllers with 9 and 4 rules. Final decision surface after 300 epochs of training is illustrated in Fig. 6(b) and (c) for 9 and 4 rules, respectively. Truck trajectory of the fuzzy controller with 9 and 4 rules after training from 4 different initial position (x_0, ϕ_0) is shown in Fig. 7.

The afore-mentioned procedure has been repeated for 20 sets of expert exemplars from randomly selected initial points in the (x, y) plane. The results obtained from 9 and 4 rules case studies are almost the same as before. We initiate with the same membership function for fuzzy values as before. Truck trajectory of the fuzzy controller from four initial state with 9 and 4 rules trained through 20 expert exemplars are depicted in Fig. 7. In both cases for 4 and 9 rules the trajectory is followed perfectly after training.

VII. Conclusions

Through illustrated examples, in this paper it has been shown that changing the membership functions of fuzzy values can affect the overall input-output mapping of FIS. This change might be directed to applying the minimum number of rules and consequently simplifying the controller design. The proposed scheme can be used to achieve any continuous nonlinear surfaces, but the gradient descent method is not capable of convergence for very sharp nonlinearities. It has been shown the control surface with fewer rules is more smooth, and this smoothness can be thought of as being more robust and fault-tolerant.

Appendix

In this appendix, the gradient of learning parameters for Gaussian membership functions is derived.

$$\frac{\partial E^p}{\partial \sigma_{ij}} = \frac{(T^p - F^p)(F^p - \overline{C}_i) I_i w_i \beta_{ij}}{(u_j - \sigma_{ij}) A_j^i(u_j) \sum_{i=1}^{n} w_i I_i} \left[\left(\frac{u_j - \sigma_{ij}}{\rho_{ij}}\right)^2\right]^{\beta_{ij}}$$

$$\cdot\, 4 \exp\left\{-\left[\left(\frac{u_j - \sigma_{ij}}{\rho_{ij}}\right)^2\right]^{\beta_{ij}}\right\}$$

$$\frac{\partial E^p}{\partial \rho_{ij}} = \frac{(T^p - F^p)(F^p - \overline{C}_i) I_i w_i \beta_{ij}}{\rho_{ij} A_j^i(u_j) \sum_{i=1}^{n} w_i I_i} \left[\left(\frac{u_j - \sigma_{ij}}{\rho_{ij}}\right)^2\right]^{\beta_{ij}} \cdot 4 \exp\left\{-\left[\left(\frac{u_j - \sigma_{ij}}{\rho_{ij}}\right)^2\right]^{\beta_{ij}}\right\}$$

$$\frac{\partial E^p}{\partial \beta_{ij}} = \frac{-(T^p - F^p)(F^p - \overline{C}_i) I_i w_i}{A_j^i(u_j) \sum_{i=1}^{n} w_i I_i} \left[\left(\frac{u_j - \sigma_{ij}}{\rho_{ij}}\right)^2\right]^{\beta_{ij}} \cdot 2 \ln \left(\frac{u_j - \sigma_{ij}}{\rho_{ij}}\right)^2 \exp\left\{-\left[\left(\frac{u_j - \sigma_{ij}}{\rho_{ij}}\right)^2\right]^{\beta_{ij}}\right\}$$

$$\frac{\partial E^p}{\partial \hat{\sigma}_i} = \frac{2(T^p - F^p) w_i}{\sum_{i=1}^{n} w_i I_i} \left[(F^p - \overline{C}_i) \frac{\partial I_i}{\partial \hat{\sigma}_i} - I_i \frac{\partial \overline{C}_i}{\partial \hat{\sigma}_i}\right]$$

$$\frac{\partial E^p}{\partial \hat{\rho}_i} = \frac{2(T^p - F^p) w_i}{\sum_{i=1}^{n} w_i I_i} \left[(F^p - \overline{C}_i) \frac{\partial I_i}{\partial \hat{\rho}_i} - I_i \frac{\partial \overline{C}_i}{\partial \hat{\rho}_i}\right]$$

$$\frac{\partial E^p}{\partial \hat{\beta}_i} = \frac{2(T^p - F^p) w_i}{\sum_{i=1}^{n} w_i I_i} \left[(F^p - \overline{C}_i) \frac{\partial I_i}{\partial \hat{\beta}_i} - I_i \frac{\partial \overline{C}_i}{\partial \hat{\beta}_i}\right]$$

For the sake of simplicity we will consider the case when $\hat{\beta}_i = 1$ and this variable is not be changed during the training process. Therefore, we have a relatively simple gradient for the other variables.

$$\frac{\partial I_i}{\partial \hat{\sigma}_i} = -\exp\left[-\left(\frac{\hat{\sigma}_i - V^+}{\hat{\rho}_i^2}\right)^2\right] + \exp\left[-\left(\frac{\hat{\sigma}_i - V^-}{\hat{\rho}_i^2}\right)^2\right]$$

$$\frac{\partial I_i}{\partial \hat{\rho}_i} = \frac{-\sqrt{\pi}}{2} \operatorname{erf}\left(\frac{\hat{\sigma}_i - V^+}{\hat{\rho}_i}\right) + \frac{(\hat{\sigma}_i - V^+)}{\hat{\rho}_i} \exp\left[-\left(\frac{\hat{\sigma}_i - V^+}{\hat{\rho}_i^2}\right)^2\right] + \frac{\sqrt{\pi}}{2} \operatorname{erf}\left(\frac{\hat{\sigma}_i - V^-}{\hat{\rho}_i}\right) - \frac{(\hat{\sigma}_i - V^-)}{\hat{\rho}_i} \exp\left[-\left(\frac{\hat{\sigma}_i - V^-}{\hat{\rho}_i^2}\right)^2\right]$$

$$\frac{\partial \overline{C}_i}{\partial \hat{\sigma}_i} = \frac{1}{I_i}\left[(\hat{\sigma}_i + V^+) \exp\left(-\frac{VVp}{\hat{\rho}_i^2}\right) - (\hat{\sigma}_i + V^-) \exp\left(-\frac{VVn}{\hat{\rho}_i^2}\right)\right] + \frac{\hat{\sigma}_i \overline{C}_i}{I_i} + 1 - \frac{\overline{C}_i}{I_i^2} \frac{\partial I_i}{\partial \hat{\sigma}_i}$$

$$\frac{\partial \overline{C}_i}{\partial \hat{\rho}_i} = \frac{1}{I_i^2}\left[\frac{\hat{\rho}_i^2}{2} \exp\left(-\frac{VVp}{\hat{\rho}_i^2}\right) - \frac{\hat{\rho}_i^2}{2} \exp\left(-\frac{VVn}{\hat{\rho}_i^2}\right) - I_i\right] \frac{\partial I_i}{\partial \hat{\rho}_i} + \frac{\hat{\sigma}_i}{\hat{\rho}_i} + \frac{1}{I_i}\left\{-\hat{\rho}_i \exp\left(-\frac{VVp}{\hat{\rho}_i^2}\right) + \frac{\hat{\sigma}_i(\hat{\sigma}_i - V^+)}{\hat{\rho}_i} \exp\left[-\left(\frac{\hat{\sigma}_i - V^+}{\hat{\rho}_i}\right)^2\right]\right\}$$
$$+ \frac{1}{I_i}\left\{\hat{\rho}_i \exp\left(-\frac{VVn}{\hat{\rho}_i^2}\right) - \frac{\hat{\sigma}_i(\hat{\sigma}_i - V^-)}{\hat{\rho}_i} \exp\left[-\left(\frac{\hat{\sigma}_i - V^-}{\hat{\rho}_i}\right)^2\right]\right\} + \frac{1}{I_i}\left[-\frac{VVp}{\hat{\rho}_i} \exp\left(-\frac{VVp}{\hat{\rho}_i^2}\right) + \frac{VVn}{\hat{\rho}_i} \exp\left(-\frac{VVn}{\hat{\rho}_i^2}\right)\right]$$

$$VVp = -2\hat{\sigma}_i V^+ + \hat{\sigma}_i^2 + {V^+}^2$$
$$VVn = -2\hat{\sigma}_i V^- + \hat{\sigma}_i^2 + {V^-}^2$$

References

[1] J. A. Bernard, "Use of a rule-based system for process control," *IEEE Control Systems Mag.*, pp. 3–12, Oct. 1988.

[2] J. S. R. Jang, "Self-learning fuzzy controllers based on temporal back propagation," *IEEE Trans. Syst., Man, Cybern.*, vol. 3, no. 5, pp. 714–723, Sept. 1992.

[3] S.-G. Kong and B. Kosko, "Adaptive fuzzy systems for backing up a truck-and trailer," *IEEE Trans. Neural Networks*, vol. 3, no. 2, pp. 211–233, Mar. 1992.

[4] G. Langari and M. Tomizuka, "Stability of fuzzy linguistic control systems," in *Proc. of the 29th Conference on Decision and Control*, Dec. 1990, pp. 2185–2190.

[5] C. C. Lee, "Fuzzy logic in control systems; fuzzy logic controller part I/II," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 404–435, Mar./Apr. 1990.

[6] S. Liu and S. Hu, "A method of generating control rule model and its application," *Fuzzy Sets Syst.*, vol. 52, pp. 33–37, 1992.

[7] A. Lotfi and A. C. Tsoi, "Importance of membership functions: A comparative study on different learning methods for fuzzy inference systems," in *Proc. Third IEEE Int. Conf. Fuzzy Systems*, Orlando, FL, June 1994, pp. 1791–1796.

[8] M. Mizumoto, "Fuzzy control under various reasoning method," *Inform. Sci.*, vol. 45, pp. 129–151, 1988.

[9] D. H. Nguyen and B. Widrow, "Neural networks for self-learning control systems," *IEEE Control Systems Mag.*, pp. 18–23, Apr. 1990.

[10] P. J. Pacini and B. Kosko, "Adaptive fuzzy systems for target tracking," *Intelligent Syst. Eng.*, pp. 3–21, July 1992.

[11] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.

[12] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1414–1427, Dec. 1992.

[13] S. Yasunobu and S. Miyamoto, "Automatic train operation system by predictive fuzzy control," *Industrial Application of Fuzzy Control*, M. Sugeno, Ed. North-Holland: B. V., Elsevier Science Publishers, 1985, pp. 1–8.

[14] L. A. Zadeh, "Outline of a new approach to the analysis of complex system and decision process," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 1, pp. 28–44, Jan. 1973.