Design of Adaptive Fuzzy Logic Controller Based on Linguistic-Hedge Concepts and Genetic Algorithms

Bin-Da Liu, *Senior Member, IEEE*, Chuen-Yau Chen, *Student Member, IEEE*, and Ju-Ying Tsao

***Abstract*—In this paper, we propose a novel fuzzy logic controller, called *linguistic hedge fuzzy logic controller*, to simplify the membership function constructions and the rule developments. The design methodology of linguistic hedge fuzzy logic controller is a hybrid model based on the concepts of the linguistic hedges and the genetic algorithms. The linguistic hedge operators are used to adjust the shape of the system membership functions dynamically, and can speed up the control result to fit the system demand. The genetic algorithms are adopted to search the optimal linguistic hedge combination in the linguistic hedge module. According to the proposed methodology, the linguistic hedge fuzzy logic controller has the following advantages: 1) it needs only the simple-shape membership functions rather than the carefully designed ones for characterizing the related variables; 2) it is sufficient to adopt a fewer number of rules for inference; 3) the rules are developed intuitionally without heavily depending on the endeavor of experts; 4) the linguistic hedge module associated with the genetic algorithm enables it to be adaptive; 5) it performs better than the conventional fuzzy logic controllers do; and 6) it can be realized with low design complexity and small hardware overhead. Furthermore, the proposed approach has been applied to design three well-known nonlinear systems. The simulation and experimental results demonstrate the effectiveness of this design.**

***Index Terms*—Adaptive fuzzy logic controller, genetic algorithm, linguistic hedge.**

I. INTRODUCTION

CONVENTIONAL control system designs heavily rely on the linearized models or the mathematical descriptions of the controlled plants. As to the real world nonlinear and more complex systems which have no adequate mathematical expression to describe them, their related controller models are therefore difficult to construct. The fuzzy logic controller (FLC) first implemented by Mamdani [1] on the basis of the fuzzy logic system generalized from the fuzzy set theory originated by Zadeh [2] has appeared to offer a feasible solution to various control problems [3]–[7]. In an FLC, the inference engine plays the role of a kernel. It explores the fuzzy rules preconstructed by experts to accomplish inference. Since the rules specify the implication relationships between the input variables and output variables characterized by their corresponding membership functions, the choice of the rules along with the membership functions makes significant impacts on the final performance of the FLC and therefore becomes the major control strategy in FLC design. The more the membership functions are used, the more the rules emerge, and the finer the results of inference.

Much research has concentrated on the rule construction in an FLC design. Simpson [8] proposed a fuzzy min–max neural network for pattern classification. He used a single pass expansion–contraction process of fuzzy set hyperboxes to learn nonlinear class boundaries. Based on his hyperbox method, Abe [9] proposed a method for fuzzy rule extraction directly from the numerical data. In the meantime, Wang [10] proposed general methods to generate the fuzzy control rules automatically according to the input–output (I/O) data pairs of the control system. With these methods, the rule extraction process can be done by one-pass to reduce the system construction complexity, and the experience of a human expert can be combined with the rules obtained from automatic learning. On the other hand, the tree structure is such a simple and easily understood method for modeling the problem that many applications are solved [11]–[14]. Turksen [15] proposed a two-level tree search method for a fuzzy expert system. With his method, taking advantage of the tree approach, the computational complexity and the search time of the fuzzy system can be reduced. In addition, Liu [16] proposed a tree-based FLC design methodology. By means of this method, not only can the control rules be extracted automatically but the search time can also be significantly reduced.

With regard to the membership functions, instead of having the membership functions constructed manually by skilled operators or experts, several researchers have proposed methods for automatically selecting the high performance membership functions for FLC's. Karr [17] properly specified the membership functions to ensure efficient FLC performance by using the genetic algorithms (GA) [18], [19] which have the ability of searching the near-optimal or optimal solutions in the solution space and are widely used in many applications [20]–[23]. Chang [23] adopted the GA-based tuning methods to membership function tuning. Iokibe [24] automatically generated the membership function by means of the fuzzy clustering method which produces a much richer construction efficiency than the neural network approach [25]. Krishnapuram [26] relied on the properties of possibilistic clustering to develop an approach for generating membership functions. Kim [27] applied the concept of inductive reasoning to generate the membership function. In his scheme, no extra information is needed except the experienced data describing the input and output relationships. For the

Manuscript received January 24, 1999; revised September 23, 2000. This work was supported by the National Science Council, R.O.C., under Grant NSC-89-2215-E-006-021.

B.-D. Liu and C.-Y. Chen are with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan 70101, R.O.C. (e-mail: bdliu@cad.ee.ncku.edu.tw).

J.-Y. Tsao was with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan 70101, R.O.C. She is now with EE-Solutions, Inc., Hsinchu, Taiwan 30052, R.O.C.

Publisher Item Identifier S 1083-4419(01)00088-7.

multivariable fuzzy systems, Chen [28] developed a technique to decompose a system with complicated I/O relationships into the accumulation of simple I/O relationships, and to extract a suitable membership function for each simple subsystem. Wu [29] proposed an algorithm to generate fuzzy rules and train membership functions based on the defined format of fuzzy rules.

On the other hand, Zadeh [30] proposed the fuzzy linguistic hedges such as *very, more or less, much, essentially*, and *slightly* to modify the membership functions of the fuzzy sets. Since the linguistic hedges proposed by Zadeh in 1973, only a small amount of literature dealing with these concepts has been published [31]–[39]. Banks [31] used hedge operations to better qualify and emphasize the crisp variables to mix crisp and fuzzy logic in applications. Bouchon-Meunier [32] investigated several interesting properties of linguistic hedge, such as

1) being compatible with simple symbolic rules;
2) avoiding computations and being compatible with the fuzzy logic;
3) enhancing the comparison of various available fuzzy implication;
4) managing gradual rules in the context of deductive rules.

Modifying the existing linguistic hedge models, Novák [33] proposed a horizon-shifting model of linguistic hedges, by which the membership function can be shifted as well as its steepness modified. In addition, the concept of extended hedge algebras and their application in approximate reasoning was discussed by Ho [34], [35]. To maintain the completeness of the set of the linguistic hedges, Liu [36]–[39] proposed several hedge operators. Their related hardware realizations in current-mode approach are also found.

Although the rule developments and the membership function constructions can be accomplished automatically by a variety of algorithms, this major strategy in an FLC design still becomes a considerable challenge when the number of mentioned variables is increased, when the parameters of the controlled plants are varied, or when the conditions of the environment are changed. To aim directly at this point, an adaptive FLC with ease of membership function constructions and rule developments is necessary. In this paper, we take advantage of the superior characteristics inherent in the linguistic hedges and the search ability of GA to design a novel FLC called a *linguistic hedge fuzzy logic controller* (LHFLC). In this controller, each variable utilized is characterized by only three fuzzy sets with simple-shape membership functions; therefore, the maximum number of fuzzy rules required for a system with N input variables is 3^N. Moreover, a module called the *linguistic hedge module* embedded in this controller plays the role of a linguistic modifier. It is used to dynamically modify the shape of simple-shape membership functions according to the feedback signal from the controlled plants. This modification action allows the LHFLC to utilize only fewer rules and simple-shape membership functions without the satisfactory performance degrading. In addition, to prevent the foul factors from damaging the system design, the adjustment of the linguistic hedge module through the attached GA module makes this controller adaptive. Finally, three well-known examples including the nonlinear plant model control system [40], the truck backer-upper control system [41], and the cart-pole balance system [42] will be used to verify the feasibility of this LHFLC.

This paper is organized as follows. In Section II, we will briefly review the fuzzy set theory and introduce the concept of fuzzy linguistic hedges. Besides, the FLC will be also mentioned. In Section III, the architecture of the proposed LHFLC will be presented. The GA algorithms used for searching the optimal hedge combination will be presented in Section IV. In Section V, three well-known examples along with their simulation results will be presented to verify the feasibility of this LHFLC. In addition, the experimental results acquired by controlling the real cart-pole balance system will also be demonstrated. Section VI concludes this work.

II. Fuzzy Sets, Fuzzy Linguistic Hedges, and Fuzzy Logic Controller

A. Fuzzy Set Theory and Fuzzy Set Operations

Fuzzy sets have been interpreted as membership functions μ_F that associate with each element x of the universe of discourse X a number $\mu_F(x)$ in the interval $[0,1]$. In essence, a fuzzy set F may be represented in the form of

$$F = \int_X \mu_F(x)/x. \tag{1}$$

In the case of F having a finite support $x_1, x_2, \ldots, x_n$, the discrete form of (2) is [30]

$$F = \sum_{i=1}^{n} \frac{\mu_i}{x_i} = \frac{\mu_1}{x_1} + \cdots + \frac{\mu_n}{x_n} \tag{2}$$

where $\mu_i (i = 1, \ldots, n)$ is the grade of membership of x_i in F.

Unlike the crisp set logic that distinguishes the members of a given set from no-members by binary decision, the fuzzy sets are characterized by their membership functions. In order to manipulate the fuzzy sets as well as ordinary sets with Boolean operations, Zadeh [30] proposed the extension of the ordinary set theory for fuzzy sets. Let A and B be two fuzzy sets in X with membership functions μ_A and μ_B, respectively. The fuzzy set operations of *union*, *intersection*, and *complement* are defined as follows.

Union:

$$\mu_C(x) = \mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x)), \quad x \in X. \tag{3}$$

Intersection:

$$\mu_C(x) = \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x),\ \mu_B(x)), \quad x \in X. \tag{4}$$

Complement:

$$\overline{\mu_A}(x) = 1 - \mu_A(x). \tag{5}$$

These fundamental fuzzy operations are often used to build the other fuzzy logic functions.

B. Fuzzy Linguistic Hedges

In a fuzzy logic based system, the information is described linguistically. The linguistic hedge is an operator with an operation like a modifier used to modify the shape of membership functions. According to the statement in [30], linguistic hedge operations can be classified into three categories: *concentration*, *dilation*, and *contrast intensification*. In this paper, we only focus on the *concentration* type and the *dilation* type hedge operations.

1) Concentration: Applying a concentration operator to a fuzzy set A results in the reduction in the magnitude of the grade of membership of x in A which is relatively small for those x with a high grade of membership in A and relatively large for those x with low membership. The hedge operation of "*concentration* x" defined by Zadeh [30] is

$$\mathrm{CON}(x) \triangleq x^{\alpha}; \quad \alpha > 1. \tag{6}$$

Based on the above definition, a few related hedge operations such as *absolutely*, *very*, *much more*, *more*, and *plus* can be defined as [36], [39]

$$\text{absolutely } x \triangleq x^{4} \tag{7}$$
$$\text{very } x \triangleq x^{2} \tag{8}$$
$$\text{much more } x \triangleq x^{1.75} \tag{9}$$
$$\text{more } x \triangleq x^{1.5} \tag{10}$$
$$\text{plus } x \triangleq x^{1.25}. \tag{11}$$

2) Dilation: In contrast, the effect of dilation is opposite to that of concentration. The hedge operation of "*dilation* x" defined by Zadeh [30] is

$$\mathrm{DIL}(x) \triangleq x^{\alpha}; \quad \alpha < 1. \tag{12}$$

Similarly, some related hedge operations such as *minus*, *more or less*, and *slightly* can be defined as [38], [39]

$$\text{minus } x \triangleq x^{0.75} \tag{13}$$
$$\text{more or less } x \triangleq x^{0.5} \tag{14}$$
$$\text{slightly } x \triangleq x^{0.25}. \tag{15}$$

In order to consider the hedge effect on the fuzzy set, the hedge operator *very* is used to stand for the *concentration* type operation; the hedge operator *more or less* is used to stand for the *dilation* type operation. The fuzzy sets *cold*, *very cold*, and *more or less cold* characterized by their membership functions $\mu_{\text{cold}}(t), \mu_{\text{very cold}}(t)$, and $\mu_{\text{more or less cold}}(t)$ are shown in Fig. 1. In this figure, the membership function of the fuzzy set *very cold* is generated by applying the hedge operator *very* to that of the fuzzy set *cold* while the membership function of the fuzzy set *more or less cold* is generated by applying the hedge operator *more or less* to that of the fuzzy set *cold*. Obviously, the linguistic hedge *very* tends to narrow the shape of the membership function and decrease the membership degree; the linguistic hedge *more or less* tends to widen the shape of the membership function and increase the membership degree. That is, the members in the fuzzy set *very cold* are closer to the temperature of *cold* while the members in the fuzzy set *more or less cold* are farther far away from the temperature of *cold*.

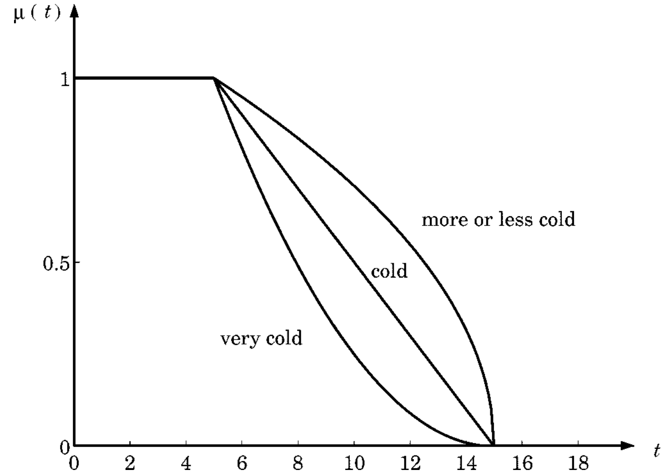


Fig. 1. Effects of the fuzzy linguistic hedge "*very*" and "*more or less*".

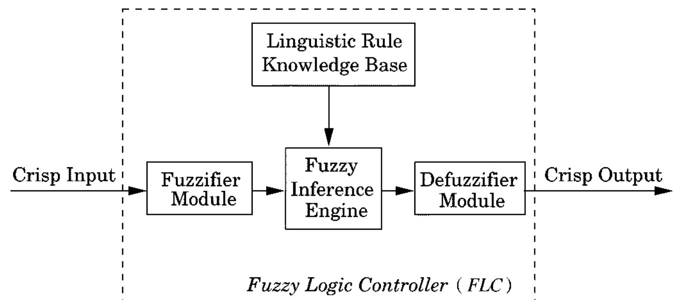


Fig. 2. Basic configuration of a fuzzy logic controller.

C. Fuzzy Logic Controller

A fuzzy logic controller designed on the basis of the fuzzy logic is an approximate reasoning-based controller, which does not require exactly analytical models and is much closer in spirit to human thinking and natural language than the traditional logic system. Fig. 2 shows the block diagram of an FLC consisting of four principal units: the *fuzzifier module*, *fuzzy inference engine*, *knowledge base*, and the *defuzzifier module*. In fuzzy control applications, the observed data are usually crisp. Since the data manipulation in an FLC is based on fuzzy set theory, fuzzification is necessary during an earlier stage. Fuzzification is related to the vagueness and imprecision in a natural language, which translates the input crisp data into the fuzzy representation for further processing. The most outstanding feature of fuzzy set theory which made it very attractive for applications is its ability to model the meaning of natural language expressions. A fuzzy system is characterized by a set of linguistic statements according to expert knowledge that is usually represented in the form of "IF–THEN" rules expressed as

IF (a set of conditions are satisfied)
THEN (a set of consequences can be inferred). (16)

The antecedent and the consequence of these IF–THEN rules are associated with fuzzy concepts, so they are often called *fuzzy* *conditional statements*. In fact, the antecedent is a condition in its application domain and the consequence is a control action for the system under control. Above all, the fuzzy control rules provide a convenient way for expressing control policy and domain knowledge. The knowledge base module is used to specify the control rules, which comprises a knowledge of the application domain and the attendant control goals. Moreover, to deal with the fuzzy information described above, the fuzzy inference engine employs the fuzzy knowledge base to simulate human decision making and infer fuzzy control actions. Finally, the defuzzifier module is used to translate the processed fuzzy data into the crisp data suited to real world applications.

Consider a multivariable fuzzy control system with three inputs and three outputs. The linguistic description of the system control rules can be expressed as [5] [see (17) at the bottom of the page], where $X_{k(i)}$ is the fuzzy value of the k-th input variable defined in the universe of discourse $U^k, k = 1,2,3$; and $Y_{j(i)}$ is the fuzzy value of the j-th output variable defined in the universe of discourse $W^j, j = 1,2,3$.

Assume that the outputs depend only on the inputs and have nothing to do with the other outputs. Thus, the linguistic description of the control rules can be rewritten as

$$\begin{aligned}&\text{IF } X_{1(1)} \text{ AND } X_{2(1)} \text{ AND } X_{3(1)} \text{ THEN } Y_{j(1)}\\&\text{or}\\&\vdots\\&\text{IF } X_{1(i)} \text{ AND } X_{2(i)} \text{ AND } X_{3(i)} \text{ THEN } Y_{j(i)}\\&\text{or}\\&\vdots\\&\text{IF } X_{1(n)} \text{ AND } X_{2(n)} \text{ AND } X_{3(n)} \text{ THEN } Y_{j(n)}\end{aligned} \tag{18}$$

where $j = 1,2,3$. The three-input/three-output system can be decomposed into 3 three-input/one-output systems. Regarding the three-input/ one-output system, the fuzzy relation of this system defined by Zadeh [30] can be expressed as

$$R_1 = \bigvee_{i=1}^{n} \{X_{1(i)} \wedge X_{2(i)} \wedge X_{3(i)} \wedge Y_{1(i)}\}. \tag{19}$$

If the present inputs are X'_1, X'_2, and X'_3, then the present output Y'_1 can be determined by the compositional rule of inference [30]

$$Y'_1 = (X'_1, X'_2, X'_3) \circ R_1. \tag{20}$$

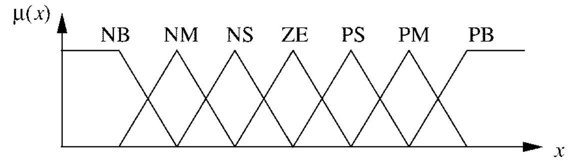


Fig. 3. Membership functions of the fuzzy sets NB, NM, NS, ZE, PS, PM, and PB.

Essentially, the control strategies in the FLC are based on expert experience, so the fuzzy logic controller can be regarded as the simulation of a humanoid control model. When designing a FLC, the control strategies have to be based on the determination of the fuzzy membership function of control variables and the linguistic control rules. Therefore, after finishing the design of a controller, if the control result fails to meet the system requirements due to a change in the outside environment of the control system, the system control strategies have to be modified to fit the control objective. The possible solution to this problem is that we can adjust either the membership function of the fuzzy sets or the control rules to achieve the control objective.

To explain the linguistic hedges from a more physical perspective, we consider their effects on the inference performance of an FLC whose goal is to produce the suitable control actions to control the controlled plants reaching the desired situations. In general, the schedule of the membership functions of the fuzzy sets in this case are in the form shown in Fig. 3, in which the fuzzy sets are labeled with linguistic variables Negative–Big (NB), Negative–Medium (NM), Negative–Small (NS), Zero (ZE), Positive–Small (PS), Positive–Medium (PM), and Positive–Big (PB). What an FLC should do is lead the controlled plant to the state such that the input variables and the output variables of this FLC enter the range around the fuzzy set ZE or reach ZE, which indicates that a balanced condition is met. For simply explaining the effect of linguistic hedges on the membership functions, we concentrate only on the fuzzy sets NB, ZE, and PB. The dashed lines in Fig. 4 are the membership functions of the fuzzy sets NB, ZE, and PB. The solid lines in Fig. 4(a) represent the effect of the hedge operator *more or less* on the fuzzy sets NB, ZE, and PB, while those in Fig. 4(b) reveal the effect owning to the hedge operator *very*. Clearly, the effect of *more or less* can be viewed as the stress of fuzzy sets in physical meaning. Alternatively, if x is located in NB or PB, the *more or less* effect increases the membership degrees and stresses the meaning of NB or PB. This action forces FLC to consider that

$$\begin{aligned}&\text{IF } X_{1(1)} \text{ AND } X_{2(1)} \text{ AND } X_{3(1)} \text{ THEN } Y_{1(1)} \text{ AND } Y_{2(1)} \text{ AND } Y_{3(1)}\\&\text{or}\\&\vdots\\&\text{IF } X_{1(i)} \text{ AND } X_{2(i)} \text{ AND } X_{3(i)} \text{ THEN } Y_{1(i)} \text{ AND } Y_{2(i)} \text{ AND } Y_{3(i)}\\&\text{or}\\&\vdots\\&\text{IF } X_{1(n)} \text{ AND } X_{2(n)} \text{ AND } X_{3(n)} \text{ THEN } Y_{1(n)} \text{ AND } Y_{2(n)} \text{ AND } Y_{3(n)}.\end{aligned} \tag{17}$$

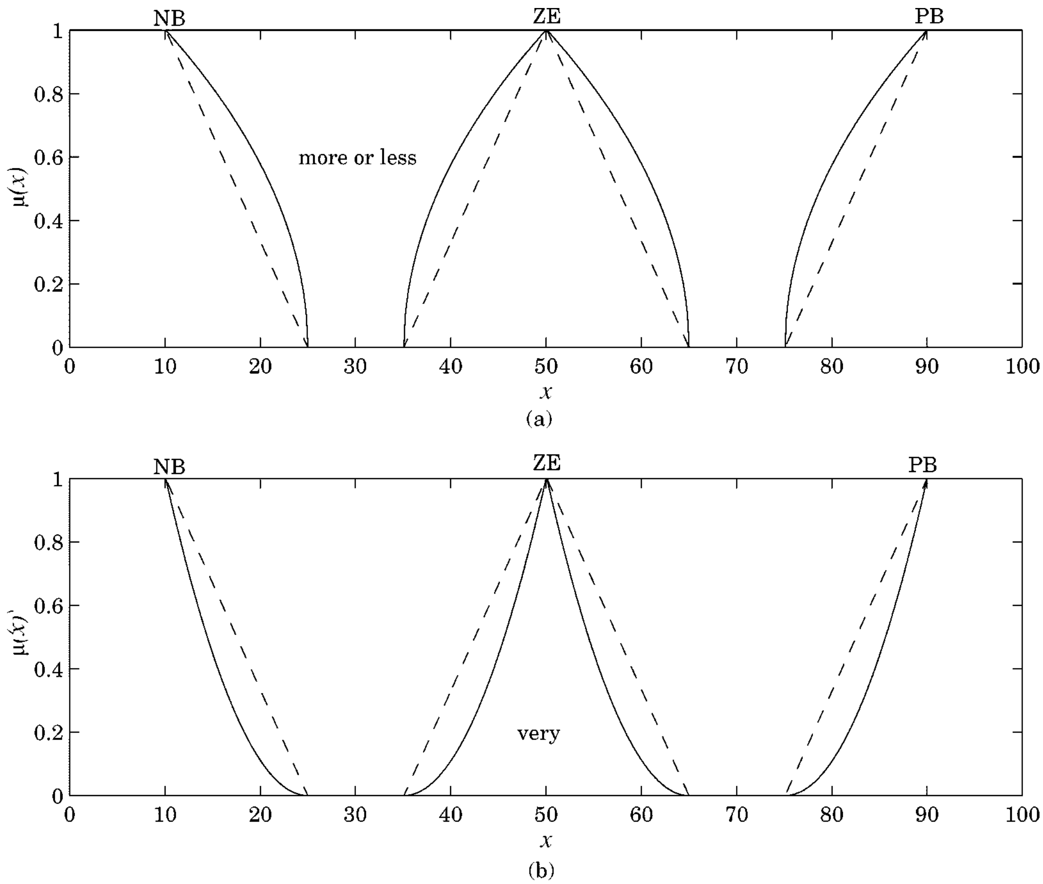


Fig. 4. Diagrammatic sketch of effects with *very* and *more or less*. (a) Effect of *more or less*. (b) Effect of *very*.

the input state is still far from the target ZE. Furthermore, the FLC stresses the output control action to reach the target earlier. The situation of x located in ZE is the same as stated above. The stress of ZE forces FLC to consider that the input state is very close to the target and to tune the output control action in a finer manner to fit control demand. On the other hand, opposite to that of *more or less*, the effect of *very* suppresses the input state. This means that if x is located in NB or PB, the suppression forces FLC to consider that the input state is not far from the target ZE and to control the output action to approach the target in a finer manner. As usual, if x is located in ZE, suppression forces FLC to consider that the input state is not close to the target and to tune the output action in a coarser manner to fit the system demand.

III. Linguistic Hedge Fuzzy Logic Controller Architecture

The LHFLC is designed by taking advantage of the superior characteristics inherent in the linguistic hedges which can be used to modify the shape of the fuzzy membership functions in order to achieve better inference performance. The major difference between this proposed LHFLC and the conventional FLC is that a module called *linguistic hedge module* is inserted into the conventional one to adjust the shape of fuzzy membership functions dynamically according to the feedback signal from the controlled plant. The emerged interesting result is that this LHFLC maintains better performance even though the number of the inference rules is reduced to a number as small as possible such as that only nine rules are used. Fig. 5 is the block diagram of this LHFLC, which consists of several modules similar to those in a conventional FLC except for the linguistic hedge module attached to the fuzzifier module. Relying on the benefits described, the number of inference rules used in this LHFLC is nine. These rules are usually scheduled in a 3×3 rule table. As shown in Fig. 6, three fuzzy sets labeled NB, ZE, and PB are used in this architecture, which are the most general and universal representations of membership functions used in FLC's. The Z-shape membership function $\mu_{\rm NB}(x)$ of fuzzy set NB can be expressed as

$$\mu_{\rm NB}(x) = \begin{cases} 1, & -\infty < x \le x_{\rm NB} \\ -\frac{x - x_{\rm NB}}{x_{\rm ZE} - x_{\rm NB}} + 1, & x_{\rm NB} \le x \le x_{\rm ZE} \\ 0, & x_{\rm ZE} \le x < +\infty \end{cases} \tag{21}$$

The $\wedge$-shape membership function $\mu_{\rm ZE}(x)$ of fuzzy set ZE can be expressed as

$$\mu_{\rm ZE}(x) = \begin{cases} 0, & -\infty < x \le x_{\rm NB} \\ -\frac{|x - x_{\rm ZE}|}{x_{\rm PB} - x_{\rm ZE}} + 1, & x_{\rm NB} \le x \le x_{\rm PB} \\ 0, & x_{\rm PB} \le x < +\infty \end{cases} \tag{22}$$

The S-shape membership function $\mu_{\rm PB}(x)$ of fuzzy set PB can be expressed as

$$\mu_{\rm PB}(x) = \begin{cases} 0, & -\infty < x \le x_{\rm ZE} \\ \frac{x - x_{\rm PB}}{x_{\rm PB} - x_{\rm ZE}} + 1, & x_{\rm ZE} \le x \le x_{\rm PB} \\ 1, & x_{\rm PB} \le x < +\infty \end{cases} . \tag{23}$$

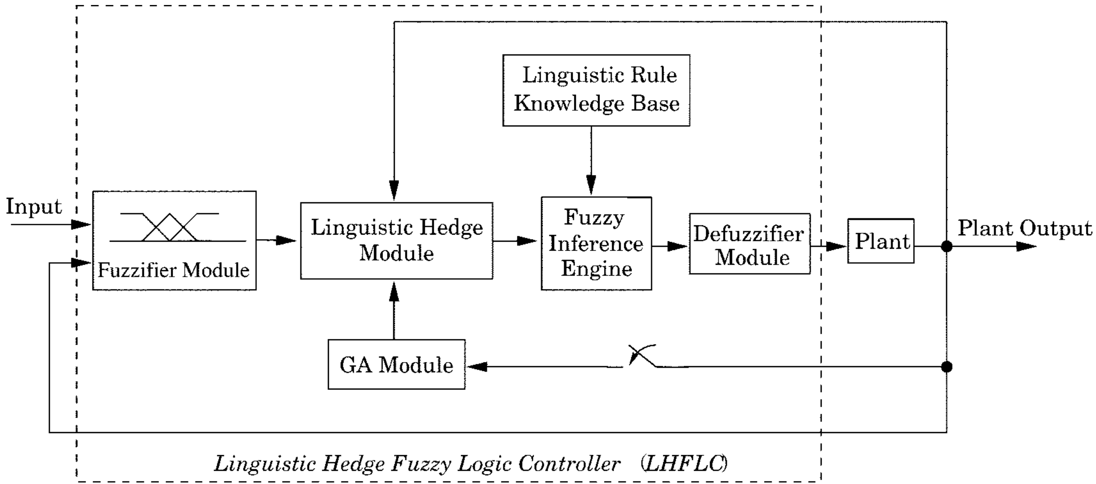


Fig. 5. Architecture of linguistic hedge fuzzy logic controller.

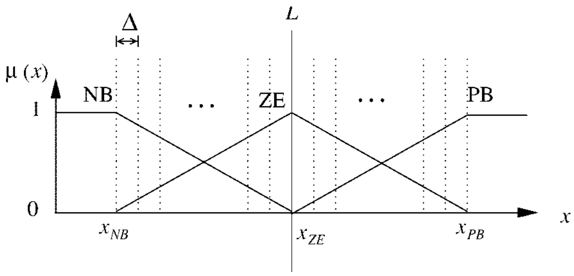


Fig. 6. Membership functions applied to linguistic hedge fuzzy logic controller.

In order to apply the hedge operations to the proposed FLC, the domains of the input variables are partitioned into n intervals. From the mathematical point of view, the membership functions $\mu_{\rm NB}(x), \mu_{\rm ZE}(x)$, and $\mu_{\rm PB}(x)$ seem to be assembled by n piecewise linear functions. These partitioned membership functions denoted as $\mu_{\rm NB'}(x), \mu_{\rm ZE'}(x)$, and $\mu_{\rm PB'}(x)$ can be expressed as

$$\begin{bmatrix}\mu_{\rm NB'}(x)\\ \mu_{\rm ZE'}(x)\\ \mu_{\rm PB'}(x)\end{bmatrix} = {\rm tr}(\mathbf{P}(x))\begin{bmatrix}\mu_{\rm NB}(x)\\ \mu_{\rm ZE}(x)\\ \mu_{\rm PB}(x)\end{bmatrix} \tag{24}$$

where ${\rm tr}(\cdot)$ denotes the trace of a matrix, and $\mathbf{P}(x)$ is the partition matrix defined as (25) shown at the bottom of the page. in which $\triangle$ denotes the step size of the input domain partition, and $u(x)$ is the unit step function of x defined as

$$u(x) = \begin{cases} 0 & -\infty < x < 0 \\ 1 & 0 \le x < +\infty \end{cases}. \tag{26}$$

The membership function in each interval $i (i = 1, \ldots, n)$ is now modified by its corresponding hedge operator h_i which is the i-th element of the hedge combination vector $\mathbf{h} = [h_1 \ldots h_n]$ defining the proper hedge operators of the n intervals of the whole input domain. For the sake of the convenience of mathematical expression, we define the hedge combination matrix $\mathbf{H}$ as

$$\mathbf{H} = [h_1\mathbf{v}_1 \ldots h_n\mathbf{v}_n] = \begin{bmatrix} h_1 & 0 & \cdots & 0 \\ 0 & h_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & h_n \end{bmatrix}_{n\times n} \tag{27}$$

where $\mathbf{v}_i$ is the i-th basis of n dimensional vector space, which is defined as

$$\mathbf{v}_i = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}_{n\times 1} \tag{28}$$

and

$$v_j = \begin{cases} 1, & j = i \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

That is, every entry of $\mathbf{H}$ is 0 except the diagonal entries h_is which give the hedge operators of the corresponding interval of membership function. Since the matrix $\mathbf{H}$ is diagonal, the membership functions $\mu_{H\circ{\rm NB}}(x), \mu_{H\circ{\rm ZE}}(x)$, and $\mu_{H\circ{\rm PB}}(x)$ resulting from modification by corresponding hedge operators can be expressed as

$$\begin{bmatrix}\mu_{H\circ{\rm NB}}(x)\\ \mu_{H\circ{\rm ZE}}(x)\\ \mu_{H\circ{\rm PB}}(x)\end{bmatrix} = \begin{bmatrix}{\rm tr}(\mathbf{P}(x)(\mu_{\rm NB}(x))^{\mathbf{H}})\\ {\rm tr}(\mathbf{P}(x)(\mu_{\rm ZE}(x))^{\mathbf{H}})\\ {\rm tr}(\mathbf{P}(x)(\mu_{\rm PB}(x))^{\mathbf{H}})\end{bmatrix}. \tag{30}$$

After processing in the fuzzifier module and the linguistic hedge module, we send the resulting signals to the succeeding stage re-

$$\mathbf{P}(x) = \begin{bmatrix} u(x - x_{\rm NB}) - u(x - x_{\rm NB} - \triangle) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & u[x - x_{\rm NB} - (n-1)\triangle] - u(x - x_{\rm NB} - n\triangle) \end{bmatrix}_{n\times n} \tag{25}$$

ferred to as the inference engine. This stage infers the fuzzy control actions employing fuzzy implication and rules constructed by the expert experience. The fuzzy reasoning method adopted in the LHFLC is Mamdani's Minimum Operation Rule [43]. The final stage is the defuzzifier module whose function is to transfer the signal from the fuzzy set into the real world for obtaining the actual control actions. The widely used method, central of gravity (COG), is adopted in the proposed LHFLC. In this case, the crisp output z_0 can be derived by [5]

$$z_0 = \frac{\sum_{i=1}^{n} \mu_z(w_i) \cdot w_i}{\sum_{i=1}^{n} \mu_z(w_i)} \tag{31}$$

where w_i is the abscissa at which the membership function reaches the maximum value $\mu_z(w_i)$ and n is the number of quantization levels of the output variable z.

According to the above descriptions, we can find that the characteristic of this architecture simplifies the complexity of the LHFLC design both on architecture itself and the hardware realization. From the viewpoint of the LHFLC architecture itself, inserting a linguistic hedge module allows us to use the simple triangle-like membership functions and a fewer number of rules instead of the more carefully designed membership functions and the large number of rules to reach the control goals. As a result, the membership function constructions and the rule developments become simpler works. From the viewpoint of the hardware realization, by comparing this LHFLC and the conventional FLC's, we can find that only one extra module called a linguistic hedge module is inserted. Besides, the fuzzifier circuit (membership function generator) becomes simpler than those of the conventional FLC's because it only has to generate fewer and simpler triangle-like membership functions; thus, size of memory is decreased dramatically because fewer rules are needed storing. Therefore, this LHFLC can be realized with low design complexity and small hardware overhead.

IV. Optimization Mechanisms

In LHFLC, we must tune the linguistic hedge combinations which are difficult to be contributed according to human experience and knowledge. To acquire an optimal combination, we adopt the GA's as the search method. In this work, the GA module works offline. That is, it searches the optimal linguistic hedge combination vector according to the controlled plants specified at first, and then provides this solution to the linguistic hedge module to make the LHFLC adaptive. Among all the various GA's, the simple GA is the simplest one without loss of efficiency. In this work, we adopted a modified version of a simple GA for increasing the linguistic hedge combination variety while searching the optimal solution. Fig. 7 shows the flow chart of the modified simple GA. In this algorithm, ten individuals with lower fitness among the whole population are removed, and ten newly generated individuals fill the resulting vacancies in the population. This operation increases the variety of the combination of linguistic hedges and enhances the search ability. Before proceeding with this GA approach, there are two preliminaries to be finished.

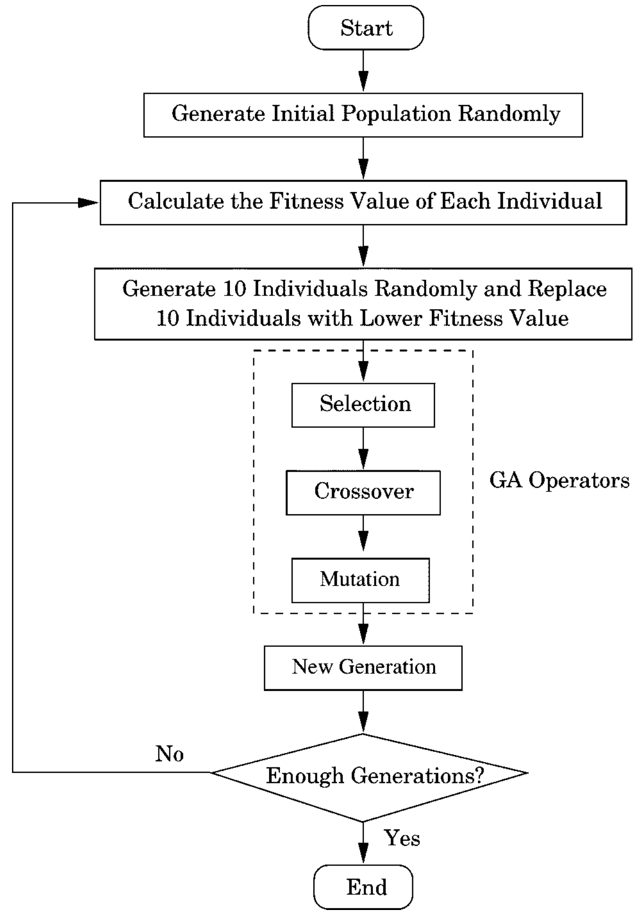


Fig. 7. Modified simple GA approach for searching the optimal hedge combinations.

A. Two Preliminaries in GA

1) Definition of Suitable Coding: One of the most attractive problems in GA's is coding the solution space. According to the eight hedge operations mentioned before and the *empty-hedge* operation defined as

$$\text{empty-hedge } x \triangleq x^1 \tag{32}$$

there are nine possible hedge operations to be adopted in the linguistic hedge module for modifying the corresponding membership functions properly. Consider the input domain partitioned into n intervals with their own hedge operations. In the case of a system with three fuzzy sets, the overall number of linguistic hedge combinations is as many as 3×9^n. That is, there are as many as 3×9^n kinds of encoding representations to be searched. To simplify this search problem, we investigate these three mentioned membership functions as shown in Fig. 6 again. The demonstrated significant result is their symmetrical property. That is, the membership function $\mu_{\text{ZE}}(x)$ is self-symmetrical with respect to the line $L : x = x_{\text{ZE}}$; the membership functions $\mu_{\text{NB}}(x)$ and $\mu_{\text{PB}}(x)$ are symmetrical to each other with respect to the line L. Therefore, the hedge combination of the fuzzy set ZE ranging from the first interval to the $n/2$-th interval and those ranging from the $((n/2)+1)$-st interval to the n-th interval must be symmetrical with respect

TABLE I
PARAMETERS FOR MODIFIED SIMPLE GA

Number of generations	60
Population size	100
Crossover rate	0.6
Mutation rate	0.01

to the line L. Accordingly, the hedge combination vector $\mathbf{h}$ used to specify the membership function of fuzzy set ZE must be in the form of

$$\mathbf{h} = [\mathbf{A}|\mathbf{A}^r] \quad (33)$$

where $\mathbf{A} = [a_1 \ldots a_{n/2}]$ specifies the hedge operators corresponding to the $n/2$ intervals to the left of the line L, and $\mathbf{A}^r$ is the vector whose elements are in the reverse order with respect to those of $\mathbf{A}$, i.e.,

$$\mathbf{A}^r = \mathbf{AR} = [a_{n/2} \ldots a_1] \quad (34)$$

where the transfer matrix $\mathbf{R}$ is defined as

$$\mathbf{R} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix}_{\frac{n}{2} \times \frac{n}{2}} . \quad (35)$$

Similarly, consider the fuzzy sets NB and PB. The hedge combination vector $\mathbf{B}$ specifying the hedge operators ranging from the first interval to the $n/2$-th interval of the fuzzy set NB and the hedge combination vector $\mathbf{C}$ specifying the hedge operators ranging from the $((n/2)+1)$-st interval to the n-th interval of the fuzzy set PB must be also symmetrical with respect to the line L. That is

$$\mathbf{C} = \mathbf{B}^r = \mathbf{BR} = [b_{n/2} \ldots b_1]. \quad (36)$$

Obviously, once the vector $\mathbf{A}$ and the vector $\mathbf{B}$ are determined, the vectors $\mathbf{A}^r$ and $\mathbf{B}^r$ $(=\mathbf{C})$ can be also obtained in turn. The hedge combination vector $\mathbf{h}$ used to specify the fuzzy sets NB and PB becomes

$$\mathbf{h} = [\mathbf{B}|\mathbf{C}] = [\mathbf{B}|\mathbf{B}^r] \quad (37)$$

where $\mathbf{B}$ specifies the hedge operators ranging from the first interval to the $n/2$-th interval of the fuzzy set NB while $\mathbf{B}^r$ specifies the hedge operators ranging from the $((n/2)+1)$-st interval to the n-th interval of the fuzzy set PB. Accordingly, the hedge combination vectors we have to determine are $\mathbf{A}$ and $\mathbf{B}$ with the dimension of $n/2$. For GA processing, more natural representations are more efficient and produce better solutions. Hence, the real-coded representation is used to manipulate the floating point hedge operators. Each individual to be considered is encoded as a vector of floating point numbers, i.e.,

$$\mathbf{h} = [\mathbf{A}|\mathbf{A}^r] \quad \text{or} \quad \mathbf{h} = [\mathbf{B}|\mathbf{B}^r] \quad (38)$$

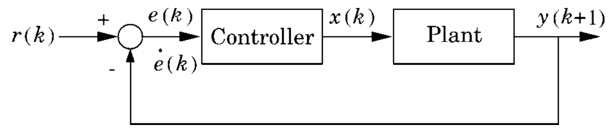


Fig. 8. Control loop of nonlinear plant model control system.

$\dot{e}$ \ e	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NM	NM	NS	NS	ZE
NM	NB	NM	NM	NS	NS	ZE	PS
NS	NM	NM	NS	NS	ZE	PS	PS
ZE	NM	NS	NS	ZE	PS	PS	PM
PS	NS	NS	ZE	PS	PS	PM	PM
PM	NS	ZE	PS	PS	PM	PM	PB
PB	ZE	PS	PS	PM	PM	PB	PB

Fig. 9. The 7×7 rule table of nonlinear plant model control system.

where

$$\mathbf{A} = [a_1 \ldots a_{n/2}] \quad (39)$$
$$\mathbf{B} = [b_1 \ldots b_{n/2}] \quad (40)$$
$$a_i, b_i \in S, i = 1, \ldots \frac{n}{2} \quad (41)$$

and

$$S = \{\text{all hedge operators such as } 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, \text{ and } 4\}. \quad (42)$$

This kind of encoding representation reduces the number of the possible individuals in the search space dramatically from 3×9^n to $2 \times 9^{n/2}$.

2) Choice of Fitness Function: The second preliminary to be finished is choosing the problem-dependent fitness function. Different fitness functions promote different GA behaviors, which generate fitness values providing a performance measure of the problem considered. The general form of a fitness function consists of two functions and can be expressed as the composition of a scaling function $g(\mathbf{h})$ and an objective function $m(\mathbf{h})$, i.e.,

$$f(\mathbf{h}) = g \circ m(\mathbf{h}) \quad (43)$$

where $m(\mathbf{h})$ is the objective function returning a cost value, and $g(\mathbf{h})$ is the scaling function transferring the cost value to the fitness. In this work, we choose the fitness function using the power scaling function, which can be expressed as

$$f(\mathbf{h}) = \exp(-\sigma \cdot c(\mathbf{h})) \quad (44)$$

where $c(\mathbf{h})$ stands for the cost function which varies from problem to problem, and σ can be viewed as a discernment measure.

After deciding these two preliminaries, we should choose the genetic operators. This modified simple GA consists of three kinds of genetic operations which are *selection*, *crossover*, and *mutation*.

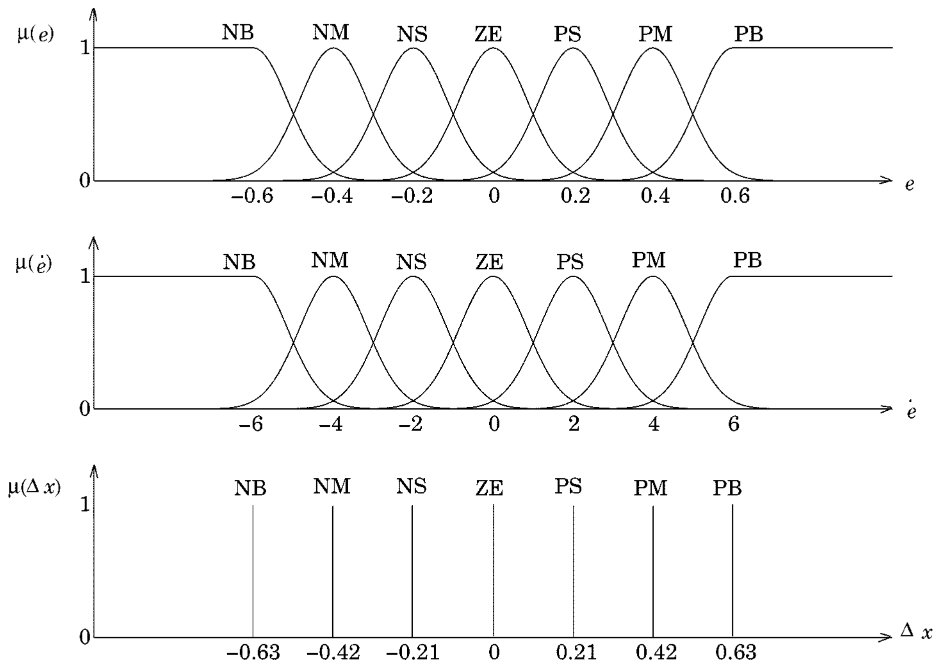


Fig. 10. Fuzzy membership functions for $e(k)$, $\dot{e}(k)$, and Δx of conventional fuzzy logic controller with 7×7 rules.

B. GA Operators

1) Selection: Selection chooses the individuals in the population as parent individuals to create offspring for the next generation, whose purpose is to emphasize the fitter individuals in the population in hopes that their offspring will in turn have even higher fitness. In this work, the implementation method of fitness-proportionate selection is adopted. The selective probability $p_S(\mathbf{h}_i)$ of the i-th individual $\mathbf{h}_i$ is

$$p_S(\mathbf{h}_i) = \frac{f(\mathbf{h}_i)}{\sum_{j=1}^{n} f(\mathbf{h}_j)}. \tag{45}$$

2) Crossover: Instead of the single-point crossover, we adopt the two-point crossover to increase the candidate population variety of the linguistic hedge combinations. For example, the parent individuals $\mathbf{h}_1$ and $\mathbf{h}_2$ given to be crossovered at the points k^+ and l^+ with the crossover probability p_c results in the new offspring $\mathbf{h}_1'$ and $\mathbf{h}_2'$ expressed as

$$\mathbf{h}_1' : h_{1i}' = \begin{cases} h_{2i}, & k^+ < i < l^+ \\ h_{1i}, & \text{otherwise} \end{cases} \tag{46}$$

and

$$\mathbf{h}_2' : h_{2i}' = \begin{cases} h_{1i}, & k^+ < i < l^+ \\ h_{2i}, & \text{otherwise.} \end{cases} \tag{47}$$

3) Mutation: Each element in a hedge combination string is a possible candidate for the mutated element that may be randomly replaced by all the hedge operators according to the mutation probability p_m. As an illustration, the individual $\mathbf{h}_1'$ mutated in the k-th element and the l-th element results in the new offspring $\mathbf{h}_1''$ expressed as (48) shown at the bottom of the next page.

In order to acquire better performance, several parameters for GA's should be set appropriately. In this work, the parameters suggested by De Jong [44] are adopted, which are widely used in GA community. These parameters are shown in Table I.

ė \ e (Δx)	NB	ZE	PB
NB	NB	NB	ZE
ZE	NB	ZE	PB
PB	ZE	PB	PB

Fig. 11. The 3×3 rule table of the nonlinear plant model control system.

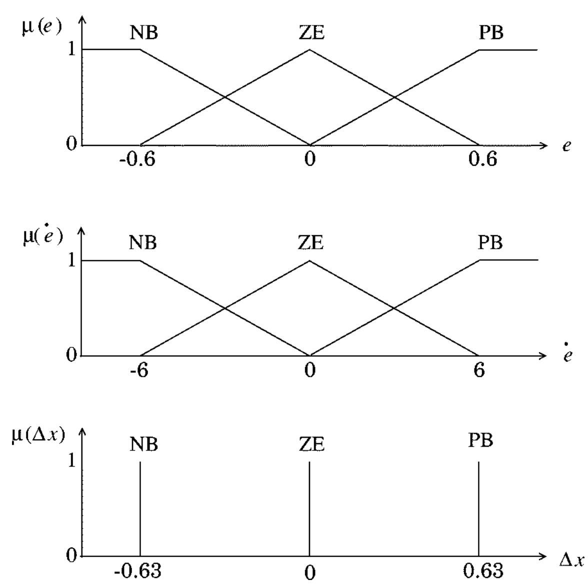


Fig. 12. Fuzzy membership functions for $e(k)$, $\dot{e}(k)$, and Δx of conventional fuzzy logic controller with 3×3 rules.

V. Demonstrative Examples and Experimental Results

The capability and feasibility of the proposed LHFLC are demonstrated in this section. The focus of this work is to emphasize that the LHFLC with fewer rules can work better than a conventional FLC with more rules. To do this,

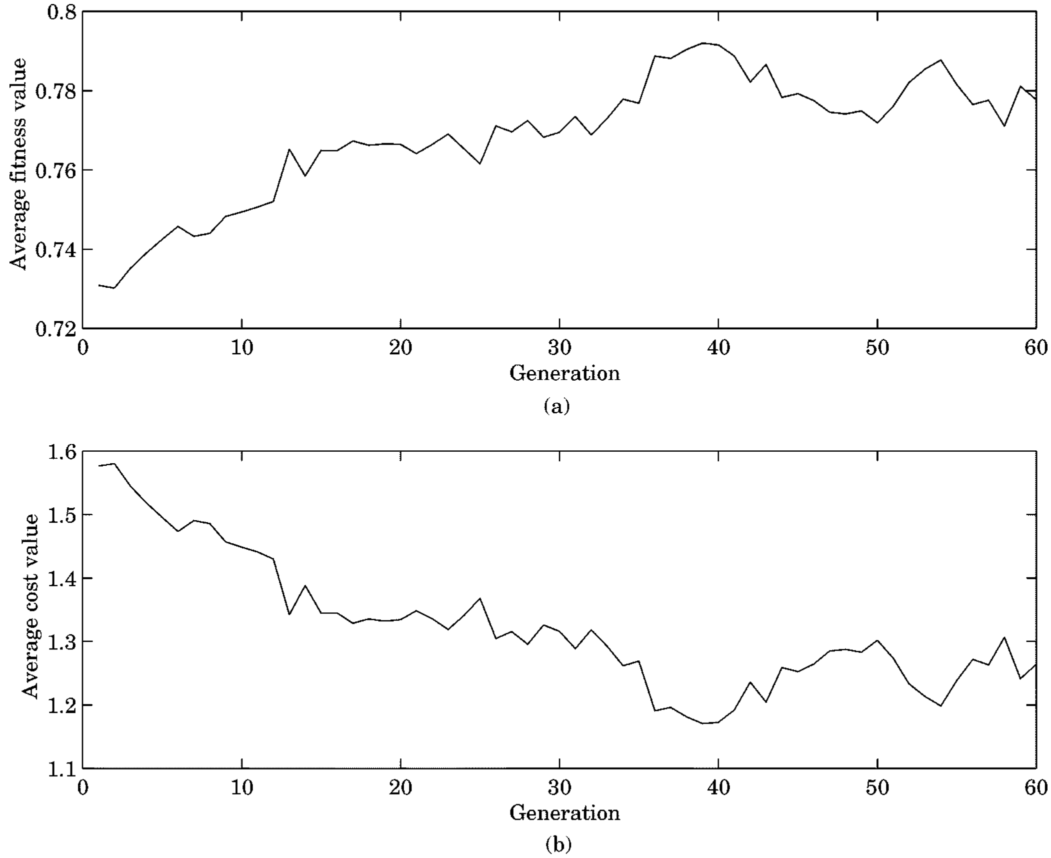


Fig. 13. Genetic algorithm performance at each generation of nonlinear plant model control system. (a) Average fitness value. (b) Average cost value.

three well-known nonlinear systems including the nonlinear plant model control system, the truck backer-upper control system, and the cart-pole balance system are used to verify the performance of the LHFLC. The number of rules chosen in this LHFLC is nine; the domain of each input variable is divided into 16 equal intervals. Throughout this work, all simulations are performed with MATLAB [45]. In addition, the physical cart-pole balance system is also used to demonstrate the feasibility of this LHFLC.

A. Nonlinear Plant Model Control System

1) Problem Description: The first example in this work is a non-BIBO nonlinear plant [40] with the plant model

$$\begin{aligned} y(k+1) = {} & 0.2y^2(k) + 0.2y(k-1) \\ & + 0.4\sin[0.5(y(k)+y(k-1))] \\ & \times \cos[0.5(y(k)+y(k-1))] + 1.2x(k) \end{aligned} \tag{49}$$

where x is the input signal of the plant and y is its output signal. The reference model that the plant output will track is chosen as

$$r(k) = 1. \tag{50}$$

The diagram of the plant control loop is plotted in Fig. 8. The goal of this system is to determine the plant input $x(k)$ such that

$$\lim_{k\to\infty} |r(k) - y(k)| < \epsilon \tag{51}$$

where ϵ is a suitably chosen constant. That is, the faster $y(k)$ tracks the reference signal $r(k)$, the better the controller will perform. In Fig. 8, the variables $e(k)$ and $\dot{e}(k)$ represent the error input and the change rate of error input of the controller, respectively, which are expressed as

$$e(k) = r(k) - y(k) \tag{52}$$

and

$$\dot{e}(k) = \frac{e(k) - e(k-1)}{\Delta t}. \tag{53}$$

where the time step Δt is chosen as 0.1 seconds. Furthermore, $x(k)$ in (49) can be expressed as

$$x(k) = x(k-1) + \Delta x \tag{54}$$

where Δx represents the increment of the plant input x at each iteration. Since the chosen plant model is nonlinear, it is difficult to handle it when the controller is determined by the clas-

$$\mathbf{h}_1'' : h_{1i}'' = \begin{cases} h_{1i}', & i \neq k, l \\ \text{hedge operators randomly selected from } S \text{ according to } p_m, & \text{otherwise.} \end{cases} \tag{48}$$

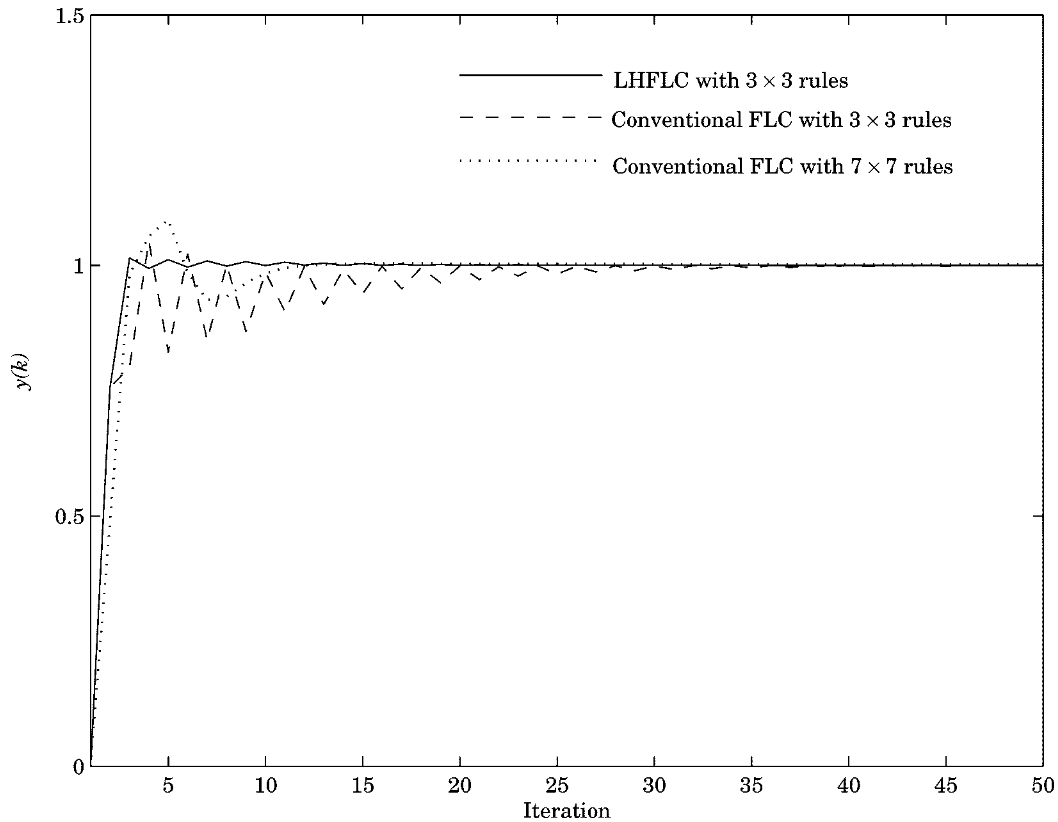


Fig. 14. Simulation result of nonlinear plant model control system.

TABLE II
PERFORMANCE FACTORS OF NONLINEAR PLANT MODEL CONTROL SYSTEM

	Conventional FLC with 7×7 rules	Conventional FLC with 3×3 rules	LHFLC with 3×3 rules
Shape of membership functions	Gauss-like	Triangle-like	Triangle-like
Settling times (s)	1.1	2.7	0.3

sical control theory. The FLC is therefore a suitable choice to replace the role played by this controller. The input variables of the FLC are the error e and its change rate $\dot{e}$; the output variable is Δx. In order to stress the power of the proposed LHFLC, the performance of the system controlled by the conventional FLC with 7×7 rules and 3×3 rules is also concerned.

2) Simulation Results:

- **Conventional FLC with 7×7 Rules**
  Fig. 9 shows the rule table specifying the implication relationships between the input variables (e and $\dot{e}$) and the increment of the output variable Δx in the FLC. As shown in Fig. 10, the input variables e and $\dot{e}$ are characterized by seven fuzzy sets with the Gauss-like membership functions distributed in the interval $e \in [-0.6, 0.6]$ and $\dot{e} \in [-6, 6]$, respectively; the increment of the output variable Δx is characterized by seven fuzzy singletons over the interval $\Delta x \in [-0.63, 0.63]$ with the support values (with membership equal to 1) located at $-0.63, -0.42, -0.21, 0, 0.21, 0.42,$ and 0.63, respectively.
- **Conventional FLC with 3×3 Rules**
  In this case, the number of rules in the conventional FLC is reduced from 7×7 to 3×3 to run the same simulation. The rule table specifying the I/O relationships is shown in Fig. 11. Fig. 12 shows the membership functions for the input variables (e and $\dot{e}$) scheduled by only three fuzzy sets with the simple shape membership functions linguistically labeled as NB, ZE, and PB distributed over the intervals $e \in [-0.6, 0.6]$ and $\dot{e} \in [-6, 6]$, respectively; the output variable Δx is characterized by three fuzzy singletons NB, ZE, and PB over the interval $\Delta x \in [-0.63, 0.63]$.
- **LHFLC with 3×3 Rules**
  The major difference between the LHFLC and the FLC with 3×3 rules is the inserted linguistic hedge module.

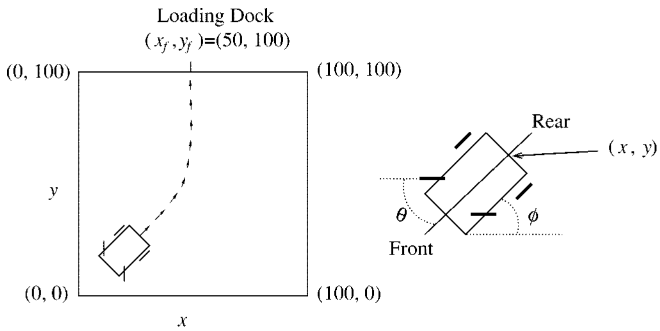


Fig. 15. Diagram of simulated truck and loading zone.

ϕ \ x (θ)	NB	NM	ZE	PM	PB
NB	PS	PM	PM	PB	PB
NM	NS	PS	PM	PB	PB
NS	NM	NS	PS	PM	PB
ZE	NM	NM	ZE	PM	PM
PS	NB	NM	NS	PS	PM
PM	NB	NB	NM	NS	PS
PB	NB	NB	NM	NM	NS

Fig. 16. The 5×7 rule table of truck backer-upper control system.

The linguistic hedge combination is searched by GA and the fitness function is chosen as

$$f(\mathbf{h}) = \exp(-\sigma \cdot c_e(\mathbf{h})) \tag{55}$$

where σ is selected as 0.2 and $c_e(\mathbf{h})$ is the cost function expressed as

$$c_e(\mathbf{h}) = \sum_{i=1}^{m} {e_i}^2(\mathbf{h}) \tag{56}$$

in which m is the number of iterations during simulation. In the GA searching phase, the performance is measured according to its corresponding cost value. The lower the cost value, the better the linguistic hedge combination searched. Hence, the variable $y(k)$ will track the reference signal $r(k)$ in the best manner possible. During this phase, the state $(e, \dot{e}) = (1, 0)$ is chosen as the initial state to search the optimal linguistic hedge combination by GA. Fig. 13 shows the GA performance. From this figure, the maximum average fitness value of 0.79 can be achieved at the 39th generation. The resultant optimal linguistic hedge combination vectors are shown in (57) at the bottom of the page for the fuzzy set ZE and in (58) shown at the bottom of the next page for the fuzzy sets NB and PB.

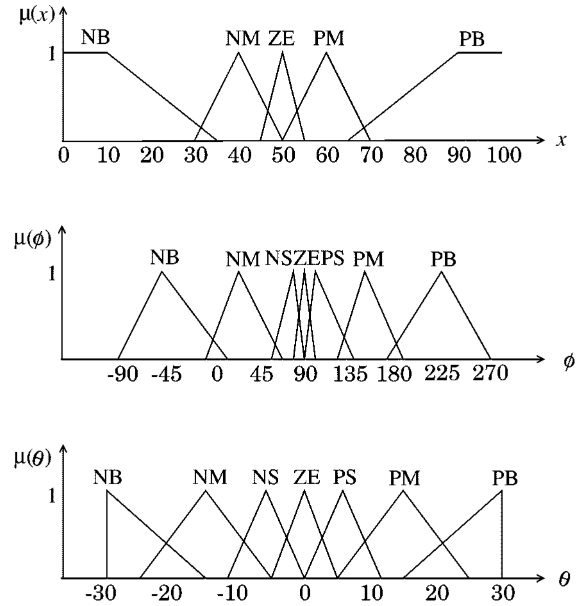


Fig. 17. Fuzzy membership functions for x, ϕ, and θ of conventional FLC with 5×7 rules.

ϕ \ x (θ)	NB	ZE	PB
NB	NB	PB	PB
ZE	NB	ZE	PB
PB	NB	NB	PB

Fig. 18. The 3×3 rule table of truck backer-upper control system.

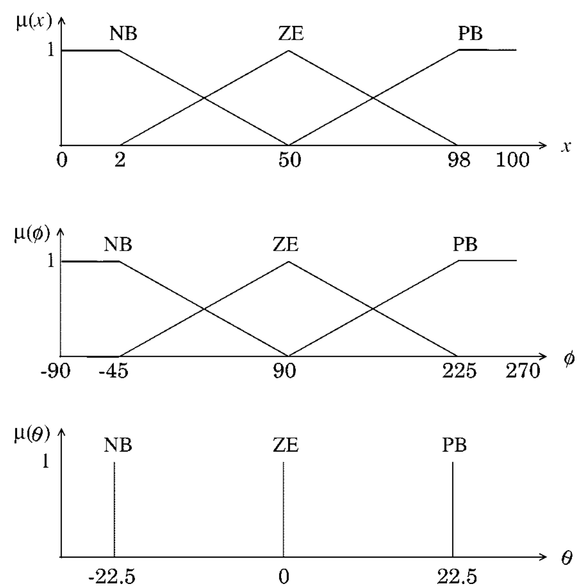


Fig. 19. Fuzzy membership functions for x, ϕ, and θ of conventional FLC with 3×3 rules.

$$\mathbf{h} = [0.25 \quad 0.5 \quad 1 \quad 2 \quad 2 \quad 2 \quad 1 \quad 1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2 \quad 1 \quad 0.5 \quad 0.25] \tag{57}$$

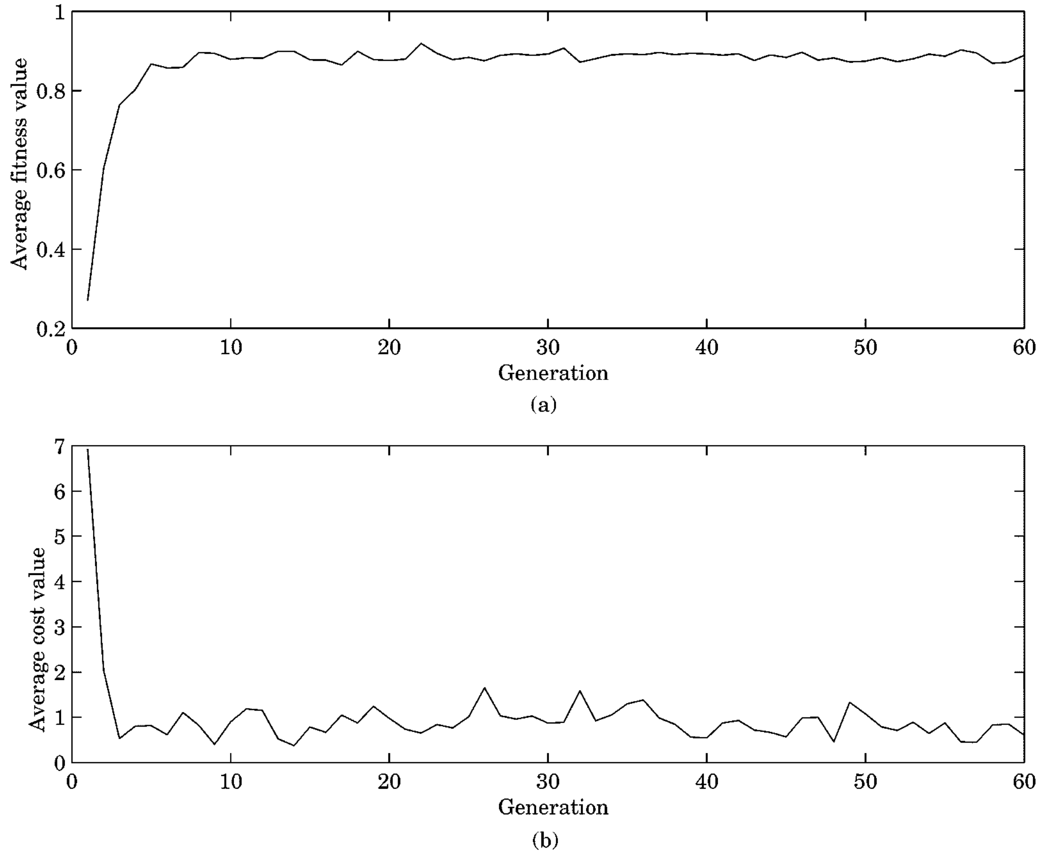


Fig. 20. GA performance at each generation of truck backer-upper control system. (a) Average fitness value. (b) Average cost value.

The performance of the plant model controlled by this LHFLC is indicated by the solid line in Fig. 14. We define the settling time as the time for response to settle to within $\pm 0.1\%$ of the steady-state value. Therefore, the plant output $y(k)$ tracks the reference signal $r(k)$ with 0.3 s settling time in the proposed LHFLC system. In contrast, the plant output $y(k)$ tracks the reference signal $r(k)$ with 1.1 s settling time when it is controlled by the conventional FLC with 7×7 rules. Also, $y(k)$ tracks $r(k)$ with 2.7 s settling time when it is controlled by the conventional FLC with 3×3 rules. Clearly, the LHFLC which adopts the least number of rules (3×3 rules rather than 7×7 rules as those in the conventional FLC) and the simplest shape membership functions (the triangle-like membership functions rather than the Gauss-like ones as those in the conventional FLC) possesses the best performance. Table II summaries these results.

B. Truck Backer-Upper Control System

1) Problem Description: The truck backer-upper control system with the goal of parking the truck in a prescribed parking lot is shown in Fig. 15. Three variables x, ϕ, and y describe this system well, where the variable ϕ specifies the angle of the truck to the horizontal while the coordinate pair (x, y) specifies the position of the rear center of the truck in the plane $[0, 100] \times [0, 100]$. The truck moves backward by some fixed distance at each step. The experiment is targeted to drive the truck to the loading dock $(x_f, y_f) = (50\ 100)$ at a right angle ($\phi = 90°$).

At each stage, the fuzzy logic controller produces the steering angle θ which causes the truck to back up to the loading zone from any initial position with any angle in the plane. The dynamic equations describing the truck moving backward from (x, y) to (x', y') at each iteration can be expressed as [41]

$$x' = x + r\cos(\phi') \tag{59}$$

$$y' = y + r\sin(\phi') \tag{60}$$

$$\phi' = \phi + \theta \tag{61}$$

where r is the fixed moving distance of the truck at each iteration. The constraints of these mentioned variables are

$$0 \le x \le 100 \tag{62}$$

$$-90° \le \phi \le 270° \tag{63}$$

$$-30° \le \theta \le 30° \tag{64}$$

$$\mathbf{h} = [0.5 \quad 2 \quad 2 \quad 0.25 \quad 4 \quad 1 \quad 2 \quad 0.25 \quad 0.25 \quad 2 \quad 1 \quad 4 \quad 0.25 \quad 2 \quad 2 \quad 0.5] \tag{58}$$

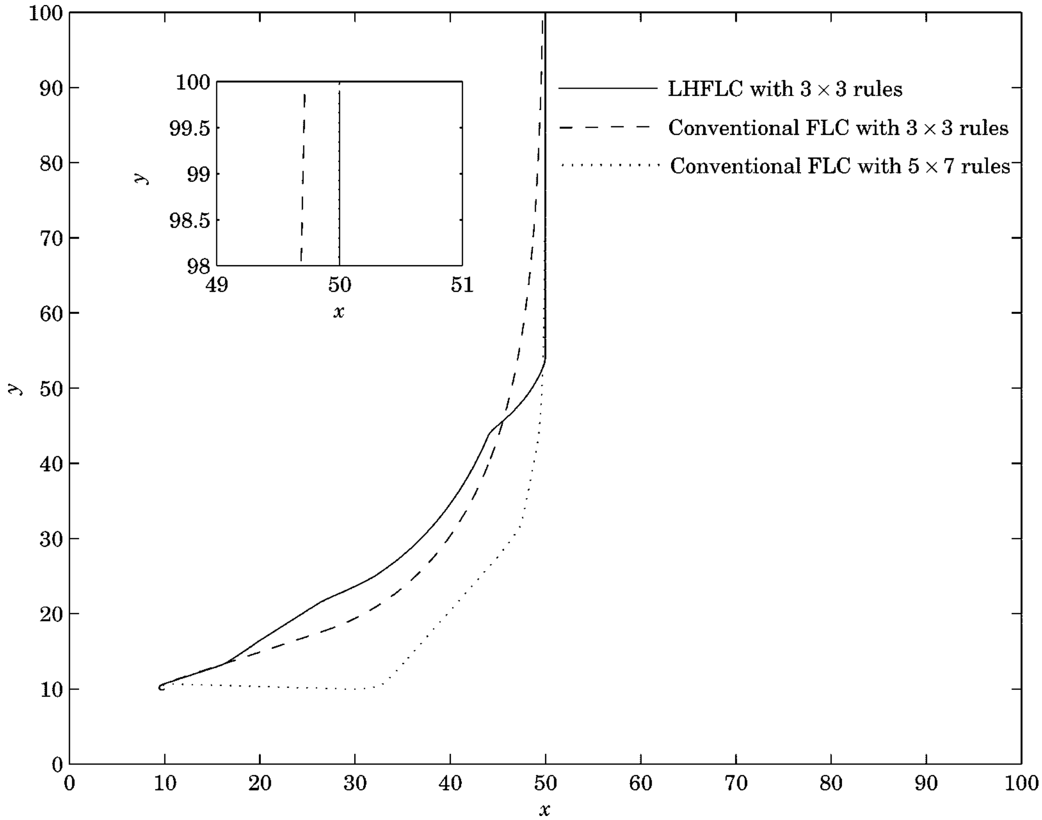


Fig. 21. Trajectories of truck with the initial point $(x, y, \phi) = (10, 10, 260°)$.

where the positive values of θ represent clockwise rotations of the steering wheel while the negative ones represent counter-clockwise rotations.

The performance of the FLC is measured in terms of the *docking error* defined as the Euclidean distance between the actual final position (ϕ, x, y) and the desired final position (ϕ_f, x_f, y_f) of the truck, i.e.,

$$\text{docking error} = \sqrt{(\phi_f - \phi)^2 + (x_f - x)^2 + (y_f - y)^2}. \quad (65)$$

For the FLC, the input variables are the x-position x and the truck angle ϕ; the output variable is the steering angle θ. The y-position can be ignored because x and y are both the functions of ϕ. In this example, the results acquired from the conventional FLC with 5×7 rules and 3×3 rules are also mentioned to demonstrate the outstanding behavior of this proposed LHFLC.

2) Simulation Results:

- **Conventional FLC with 5×7 Rules**
Fig. 16 shows the rule table specifying the implication relationships between the input variables (x and ϕ) and the output variable θ in the FLC. As shown in Fig. 17, five fuzzy sets with the carefully designed membership functions linguistically labeled as NB, NM, ZE, PM, and PB corresponding to the input variables x are designed over interval $x \in [0, 100]$; seven fuzzy sets with the carefully designed membership functions linguistically labeled as NB, NM, NS, ZE, PS, PM, and PB corresponding to the input variables ϕ are designed over interval $\phi \in [-90°, 270°]$; seven fuzzy sets with carefully designed membership functions linguistically labeled as NB, NM, NS, ZE, PS, PM, and PB corresponding to the output variable θ are designed over interval $\theta \in [-30°, 30°]$.
- **Conventional FLC with 3×3 Rules**
Similar simulation work is performed in the other conventional FLC in which the number of rules is reduced from 5×7 to 3×3 to emphasize the capability of the LHFLC. The rule table is shown in Fig. 18. As shown in Fig. 19, the input variable either x or ϕ is scheduled by only three fuzzy sets with the triangle-like membership functions linguistically labeled as NB, ZE, and PB distributed over the intervals $x \in [0, 100]$ and $\phi \in [-90°, 270°]$, respectively; three fuzzy singletons with related linguistic labels corresponding to the output variable θ are specified in the interval $\theta \in [-30°, 30°]$.
- **LHFLC with 3×3 Rules**
Similar to the LHFLC designed in the first example, the optimal linguistic hedge combination is searched by GA. The major concern is whether the truck backs at the right site at a right angle or not; hence, the docking-error is included in the fitness function to search the optimal combinations. The fitness function is chosen as

$$f(\mathbf{h}) = \exp(-\sigma \cdot c_d(\mathbf{h})) \quad (66)$$

where σ is selected as 0.8 and $c_d(\mathbf{h})$ is the cost function of the docking-error expressed in (65). In this example, the GA performance is measured according to the docking-error oriented cost value. That is, the smaller the

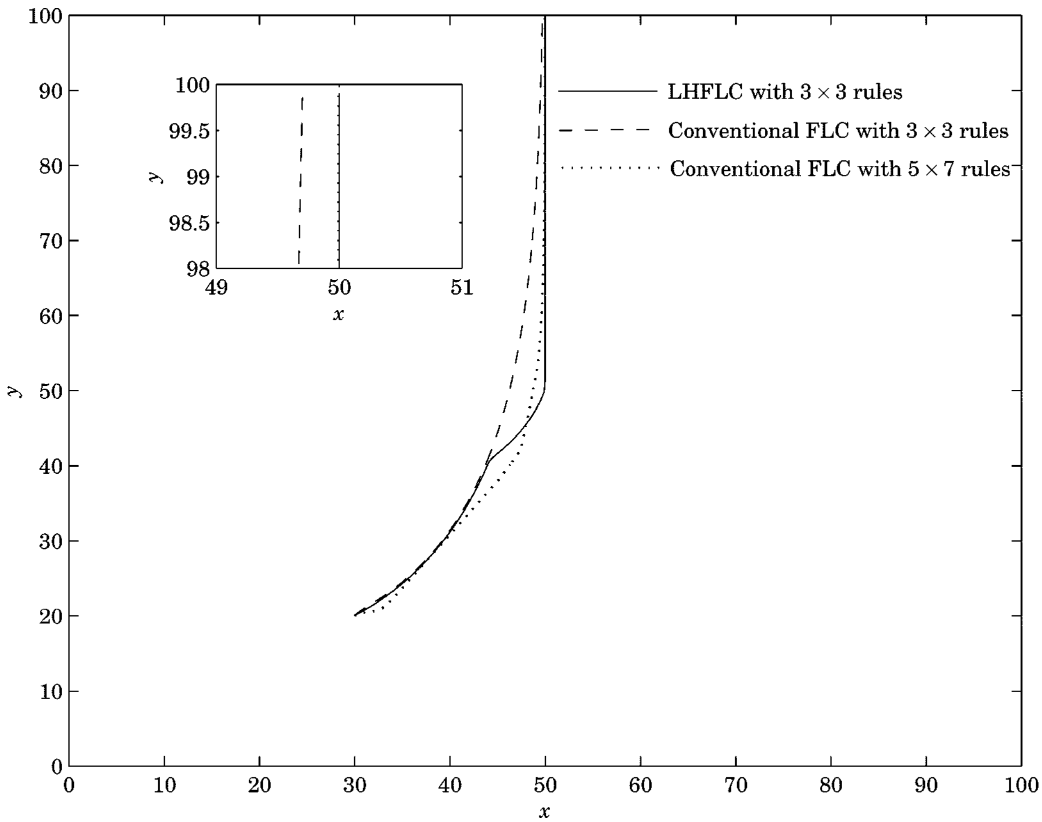


Fig. 22. Trajectories of truck with the initial point $(x, y, \phi) = (30, 20, 60°)$.

docking-error, the better the performance of this LHFLC. The truck state $(x, y, \phi) = (10, 10, 260°)$ is chosen as the initial state to search the optimal hedge combinations by GA. The GA performance is shown in Fig. 20. This figure indicates that the maximum average fitness value of 0.9 can be achieved at the 22nd generation. The resultant optimal linguistic hedge combination vectors are shown in (67) at the bottom of the page for the fuzzy set ZE and in (68) at the bottom of the page for the fuzzy sets NB and PB.

Fig. 21 shows the trajectory of the truck with the initial state $(x, y, \phi) = (10, 10, 260°)$. The performance factors are listed in Table III. According to the results in Table III, the proposed LHFLC possesses the lowest docking-error with the smallest number of iterations. To verify the generality of this linguistic hedge combination, the other initial point $(x, y, \phi) = (30, 20, 60°)$ is also chosen to simulate this truck backer-upper control system. The simulation results in Fig. 22 and Table III reveal that the LHFLC still gets the best performance of the truck backer-upper control system. These results meet our claims again. That is, the LHFLC that adopts the least number of rules (3×3 rules rather than 5×7 rules as those in the conventional FLC) and the simplest shape membership functions (the triangle-like membership functions rather than the carefully designed ones as those in the conventional FLC) possesses the best performance.

C. Cart-Pole Balance System

1) Problem Description: The cart-pole balance system is really a well-known nonlinear system the goal of which involves both vertically balancing a pole hinged to a motor-driven cart and causing the cart to be stopped at the specified position by applying forces on it either left or right. Fig. 23 represents the cart-pole balance system, which can be described by the following nonlinear differential equations [46] [see (69) and (70) at the bottom of the next page], where the related parameters are

$g = -9.8$ m/s^2 acceleration due to gravity;
$M = 1.1$ kg mass of cart;
$m = 0.1$ kg mass of pole;

$$\mathbf{h} = [0.5 \quad 1 \quad 4 \quad 1 \quad 0.5 \quad 2 \quad 2 \quad 1 \quad 1 \quad 2 \quad 2 \quad 0.5 \quad 1 \quad 4 \quad 1 \quad 0.5] \tag{67}$$

$$\mathbf{h} = [4 \quad 1 \quad 1 \quad 1 \quad 0.25 \quad 2 \quad 1 \quad 4 \quad 4 \quad 1 \quad 2 \quad 0.25 \quad 1 \quad 1 \quad 1 \quad 4] \tag{68}$$

TABLE III
PERFORMANCE FACTORS OF TRUCK BACKER-UPPER CONTROL SYSTEM

Initial points	$(x, y, \phi) = (10, 10, 260°)$			$(x, y, \phi) = (30, 20, 60°)$		
Various schemes	Conventional FLC with 5×7 rules	Conventional FLC with 3×3 rules	LHFLC with 3×3 rules	Conventional FLC with 5×7 rules	Conventional FLC with 3×3 rules	LHFLC with 3×3 rules
Shape of membership functions	Carefully designed ones	Simple triangle-like	Simple triangle-like	Carefully designed ones	Simple triangle-like	Simple triangle-like
Docking-error	0.0762	0.8676	0.0029	0.0509	0.9142	0.0093
Iterations	1185	1095	1087	868	852	846

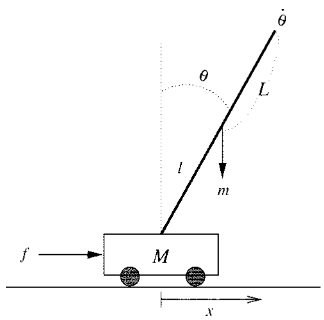


Fig. 23. Diagram of simulated cart-pole balance system.

$\dot{s}$ \ s	NB	NS	ZE	PS	PB
NB	NB	NB	NM	NS	ZE
NS	NB	NM	NS	ZE	PS
ZE	NM	NS	ZE	PS	PM
PS	NS	ZE	PS	PM	PB
PB	ZE	PS	PM	PB	PB

Fig. 24. The 5×5 rule table of the cart-pole balance system.

$L = 0.5$ m half-pole length;
$\mu_c = 0.1$ coefficient of friction of cart on track;
$\mu_p = 0.01$ coefficient of friction of pole on cart;
$\Delta t = 0.01$ s the sample time;

the four state variables and the output variable are

x position of the cart on the track (in meters);
$\dot{x}$ cart velocity (in meters per second);
θ angle of the pole with the vertical (in radians);
$\dot{\theta}$ rate of change of the angle (in radians per second)
f force (in Newtons) applied to cart's center of mass.

In this example, we use the *switching-type* fuzzy sliding mode controller (FSMC) to control the cart-pole balance system as our platform. The switching-type FSMC proposed by Li [47] is a method based on FLC and sliding mode controller (SMC) [48]–[50], which achieves asymptotic stability of the system. The dynamics of the cart-pole balance system is divided into *approaching condition* and *departure condition*. Two different FSMCs to solve control problems for these two conditions should be designed, each characterized by the associated sliding surface chosen. The sliding surface of the cart-pole balance system for the approaching mode is designed as

$$s = e_\theta + 0.4e_{\dot{\theta}} - 0.3(e_x + 0.4e_{\ddot{\theta}}) = 0 \tag{71}$$

while that for the departure mode is chosen as

$$s = e_\theta + 0.4e_{\dot{\theta}} + 0.3(e_x + 0.4e_{\ddot{\theta}}) = 0 \tag{72}$$

in which the error vector $\mathbf{E} = [e_\theta \; e_{\dot{\theta}} \; e_x \; e_{\dot{x}}]^T$ is defined as the difference between the actual state vector $\mathbf{V} = [\theta \; \dot{\theta} \; x \; \dot{x}]^T$ and the desired state vector $\mathbf{V}_d = [\theta_d \; \dot{\theta}_d \; x_d \; \dot{x}_d]^T$. That is

$$\mathbf{E} = \mathbf{V} - \mathbf{V}_d = \begin{bmatrix} e_\theta \\ e_{\dot{\theta}} \\ e_x \\ e_{\dot{x}} \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} - \begin{bmatrix} \theta_d \\ \dot{\theta}_d \\ x_d \\ \dot{x}_d \end{bmatrix}. \tag{73}$$

The input variables of the switching-type FSMC are s and its time derivative $\dot{s}$; the output variable is the force f applied to

$$\ddot{\theta} = \frac{(M+m)g\sin\theta - \cos\theta[f + mL\dot{\theta}^2\sin\theta - \mu_c \mathrm{sgn}(\dot{x})] - \frac{\mu_p (M+m)\dot{\theta}}{mL}}{\frac{4}{3}(M+m)L - mL\cos^2\theta} \tag{69}$$

$$\ddot{x} = \frac{f + mL(\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta)}{M+m} - \mu_c \mathrm{sgn}(\dot{x}) \tag{70}$$

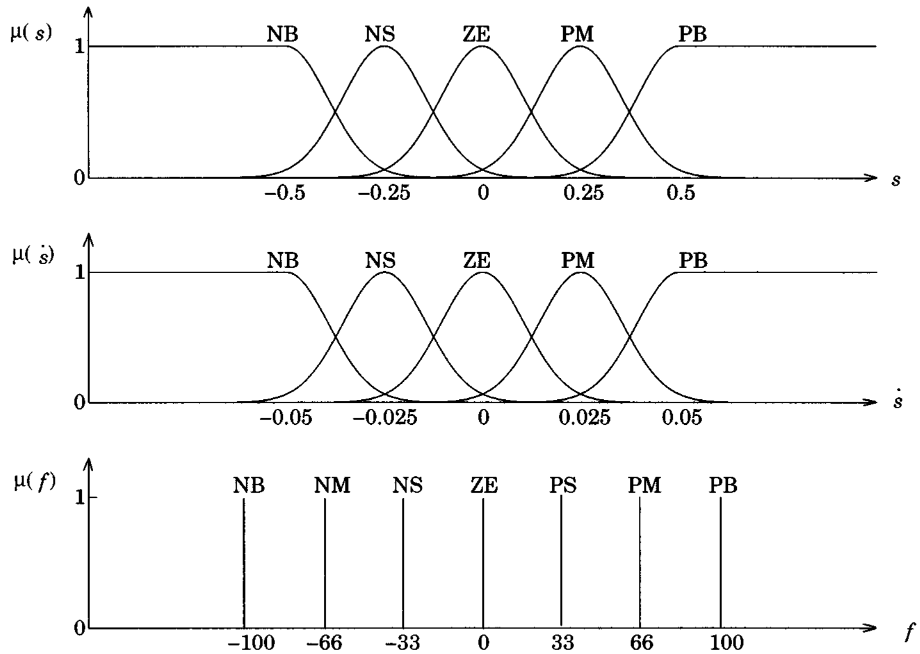


Fig. 25. Fuzzy membership functions for s, $\dot{s}$, and f of switching-type fuzzy sliding mode controller with 5×5 rules.

f \ s / $\dot{s}$	NB	ZE	PB
NB	NB	NB	ZE
ZE	NB	ZE	PB
PB	ZE	PB	PB

Fig. 26. The 3×3 rule table of the cart-pole balance system.

the cart. In the simulation as well as the experiment, the objective with which we are concerned is to control the pole being balanced at the position $x = 0.25$ m with the angle $\theta = 0$ rad.

2) Simulation Results:

- **Switching-Type FSMC with 5 $\times$ 5 Rules**
  Fig. 24 shows the rule table which specifies the implication relationships between the input variables (s and $\dot{s}$) and the output variable f in this controller. The five fuzzy sets with Gauss-like membership functions linguistically labeled as NB, NS, ZE, PS, and PB corresponding to the input variable s and $\dot{s}$ are designed over the intervals $s \in [-0.5, 0.5]$ and $\dot{s} \in [-0.05, 0.05]$, respectively, as shown in Fig. 25. The output variable f is characterized by seven fuzzy singletons designed over the interval $f \in [-100\,100]$.
- **Switching-Type FSMC with 3 $\times$ 3 Rules**
  To emphasize the capability of the LHFLC, the number of rules in the original switching-type FSMC is reduced from 5×5 to 3×3 to run the same simulation. In this case, the rule table is shown in Fig. 26. Each input variable is defined by three fuzzy sets with triangle-like membership functions distributed over the intervals $s \in [-0.5, 0.5]$ and $\dot{s} \in [-0.05, 0.05]$, respectively, as shown in Fig. 27. The output variable f is defined by three fuzzy singletons distributed over the interval $f \in [-100\,100]$.

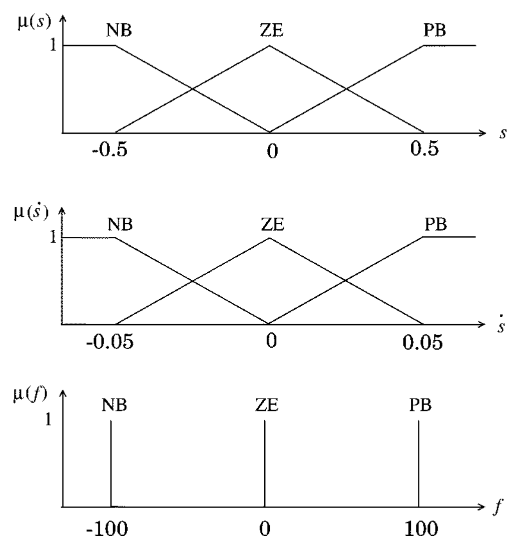


Fig. 27. Fuzzy membership functions for s, $\dot{s}$, and f of switching-type FSMC with 3×3 rules.

- **LHFLC with 3 $\times$ 3 Rules**
  In this case, the linguistic hedge concept is applied to the design of switching-type FSMC in order to modify the membership functions of the input variables s and $\dot{s}$. The optimal linguistic hedge combination is searched by GA according to the fitness function defined as

$$f(\mathbf{h}) = \exp(-\sigma(c_\theta(\mathbf{h}) + c_x(\mathbf{h}))) \tag{74}$$

where σ is selected as 0.1; $c_\theta(\mathbf{h})$ and $c_x(\mathbf{h})$ are the cost functions expressed as

$$c_\theta(\mathbf{h}) = \sum_{i=1}^{m} e_{\theta_i}^2(\mathbf{h}) \tag{75}$$

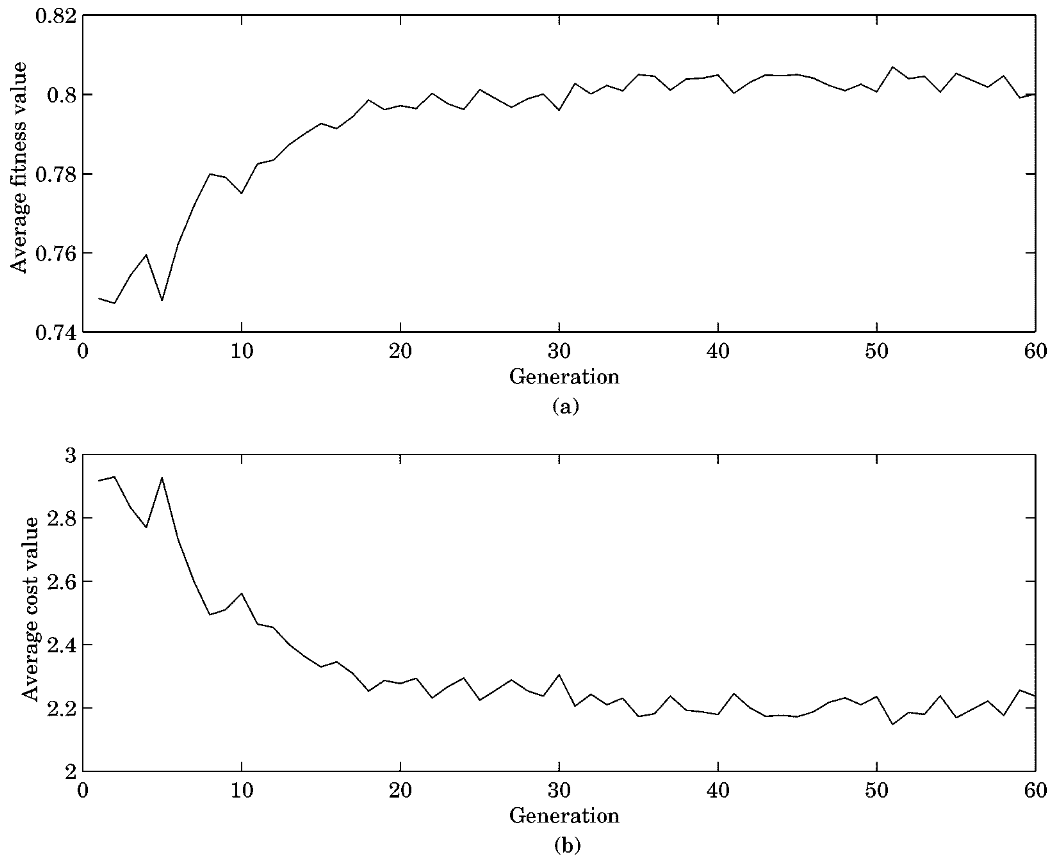


Fig. 28. Genetic algorithm performance at each generation of cart-pole balance system. (a) Average fitness value. (b) Average cost value.

and

$$c_x(\mathbf{h}) = \sum_{i=1}^{m} e_{x_i}^2(\mathbf{h}) \tag{76}$$

in which m is the number of iterations during simulation. During the search process, the state $(x, \dot{x}, \theta, \dot{\theta}) = (0, 0, 0, 0)$ is chosen as the initial state to search the optimal linguistic hedge combination by GA. The cost values returned by GA reflect the qualities of the linguistic hedge combinations searched. The lower the cost value, the better the result searched. That is, we expect the pole angle θ to reach 0 rad and the cart to be located at the desired position $x = 0.25$ m as fast as possible. Fig. 28 shows the GA performance. Obviously, the maximum average fitness value of 0.8 can be achieved at the 40th generation. The resultant optimal linguistic hedge combination vectors are shown in (77) at the bottom of the page for the fuzzy set ZE and in (78), shown at the bottom of the page, for the fuzzy sets NB and PB.

Fig. 29 shows the response of the cart-pole balance system controlled by the switching-type FSMC with either 5×5 or 3×3 rules and the LHFLC with associated $\mathbf{h}$. Obviously, either the pole angle response or the cart position response reveals that the LHFLC adopting the least number of rules and the simplest shape membership functions really enhances the performance of the FLC adopting 5×5 rules and Gauss-like membership functions or the FLC adopting 3×3 rules and triangle-like membership functions.

3) Experimental Results: To apply the proposed LHFLC to the real experimental system, the cart-pole balance system manufactured by Phimatic Enterprise Co. Ltd. is used to demonstrate this work. The arrangement of the whole experimental setup is illustrated in Fig. 30. The specifications of the related hardware are listed as follows.

$$\mathbf{h} = [0.5 \quad 4 \quad 4 \quad 0.5 \quad 0.5 \quad 0.25 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0.25 \quad 0.5 \quad 0.5 \quad 4 \quad 4 \quad 0.5] \tag{77}$$

$$\mathbf{h} = [0.25 \quad 2 \quad 1 \quad 0.25 \quad 0.5 \quad 0.5 \quad 4 \quad 2 \quad 2 \quad 4 \quad 0.5 \quad 0.5 \quad 0.25 \quad 1 \quad 2 \quad 0.25] \tag{78}$$

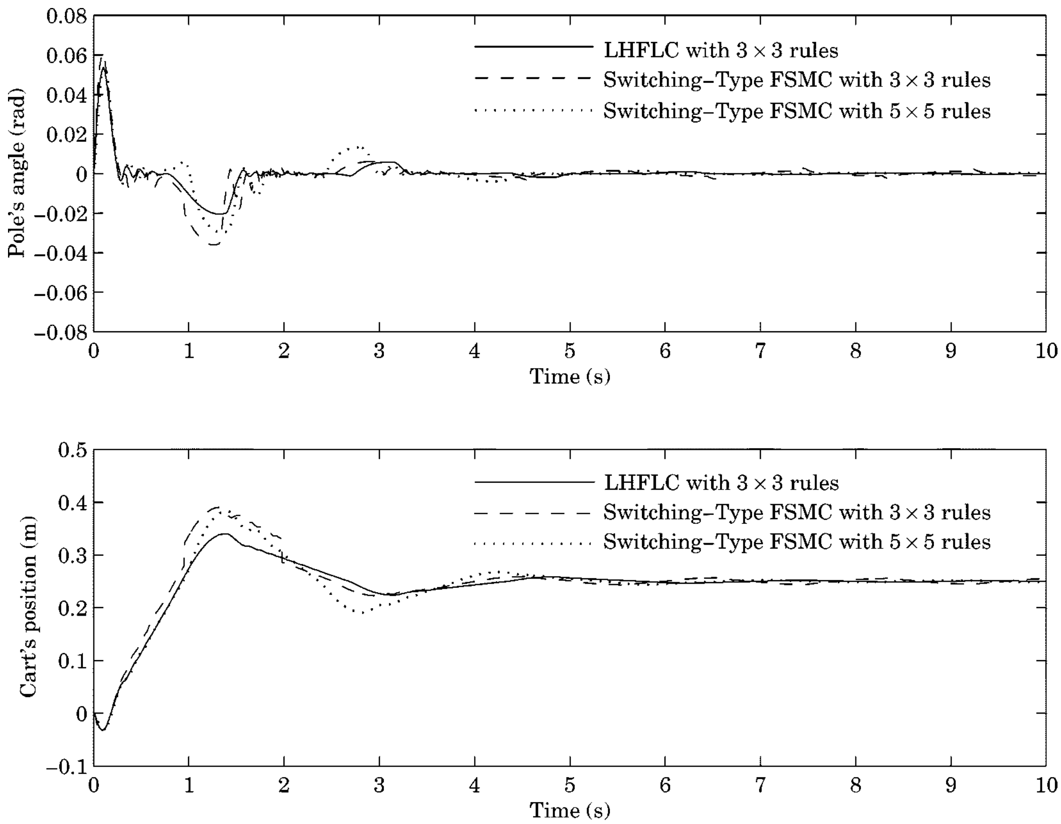


Fig. 29. Simulation result of the cart-pole balance system with initial conditions $(\theta, \dot{\theta}, x, \dot{x}) = (0, 0, 0, 0)$. (a) Response of pole angle. (b) Response of cart position.

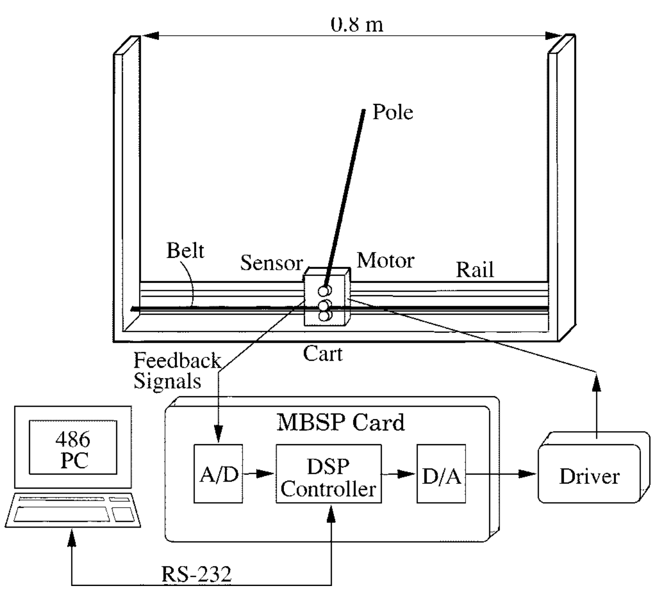


Fig. 30. Experimental setup of the cart-pole balance system.

Pole: 0.5 m length;
Drive force: dc motor (15 W);
Sensor: Photo encoder (500 pulse per rotation);
Micro-computer: 486 personal computer;
A/D, D/A & digital signal control card: MBSP card [51].

Initially, the LHFLC algorithm is programmed in C language. The control signals generated by the resultant execution code compiled from the C program are sent to the DSP controller of the MBSP card through the 25-pin RS-232 transmission line. The DSP controller processes the received signals to produce the control actions. These control actions are delivered to a dc motor via the D/A converter to apply the suitable force to the cart. The state of the cart-pole balance system is sensed by a photo encoder and fed back to the A/D converter of the MBSP card. This routine is run continuously until the system demand is met.

The responses of the pole angle and the cart position are recorded by computer and plotted in Fig. 31. Fig. 32 exhibits the photograph of the cart-pole balance system controlled by the resultant LHFLC with an exposure of about 8 s.

VI. Conclusion

In this paper, we have proposed the LHFLC. By means of adjusting the membership functions dynamically through the linguistic hedge concept, we can employ fewer rules and simple-shape membership functions to achieve a better performance than a conventional FLC does. Moreover, the GA module attached to this system with the ability of searching the optimal linguistic hedge combination allows this LHFLC to confront the variations due to internal or external factors; in other words, this LHFLC is adaptive. To verify the feasibility of this LHFLC, we have simulated three famous examples including the nonlinear plant model control system, the truck backer-upper control system, and the cart-pole balance system. The membership functions used in these systems are S-shape, $\wedge$-shape, and

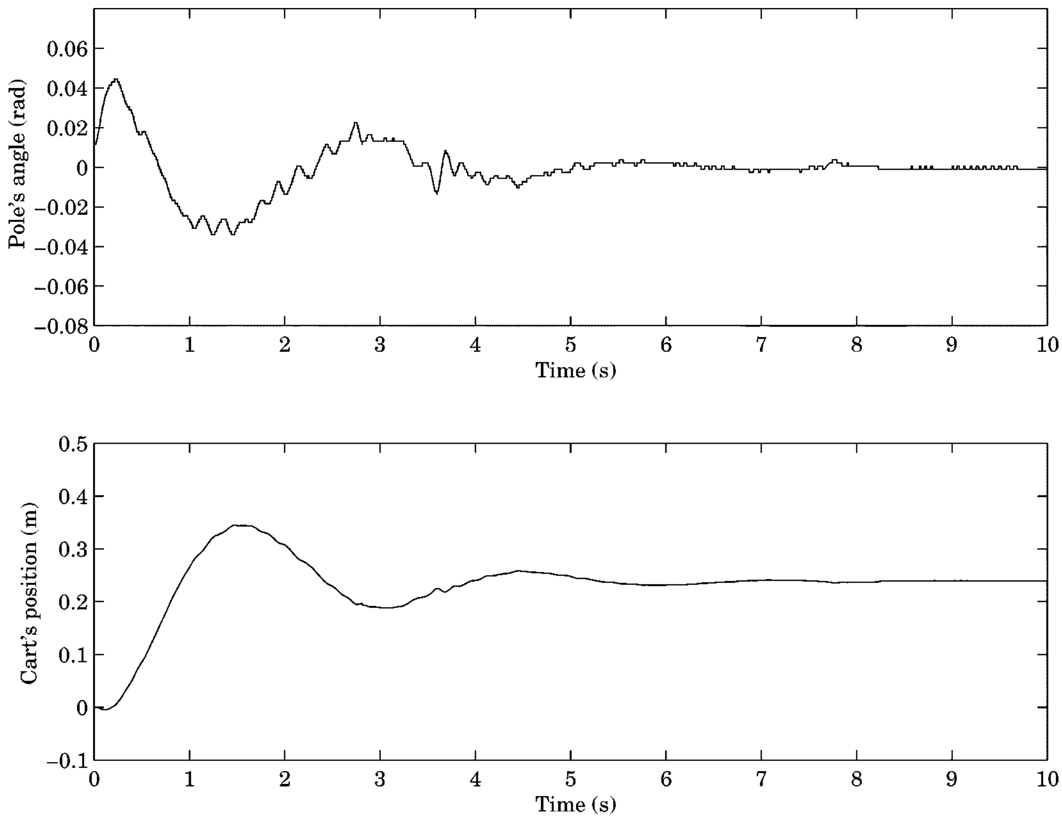


Fig. 31. Experimental result of the cart-pole balance system.

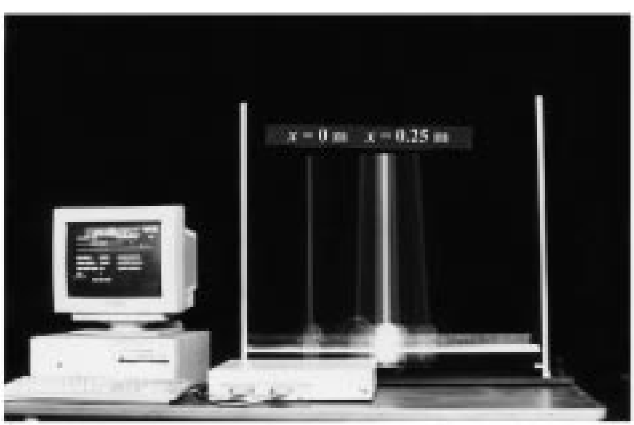


Fig. 32. Photograph of cart and pole during control with initial conditions $(\theta, \dot{\theta}, x, \dot{x}) = (0, 0, 0, 0)$.

Z-shape. The number of rules is nine and the number of partitions of each input variable is 16. We have also performed experiment on the real cart-pole balance system to prove this LHFLC to be practical. Both the simulation and experimental results reveal that the proposed scheme is really feasible.

Due to the benefits mentioned such as characterizing the related variables by simple-shape membership functions, inferring control actions based on fewer rules, and its adaptability, this LHFLC is attractive especially in the systems with more variables. Furthermore, to be suited to the real-world applications, the study on developing a hardware realization of this LHFLC in VLSI to achieve a real-time adaptive FLC will be continued in the future.

REFERENCES

[1] P. J. King and E. H. Mandani, "The application of fuzzy control systems to industrial processes," *Automatica*, vol. 13, no. 3, pp. 235–242, 1977.

[2] L. A. Zadeh, "Fuzzy sets," *Inform. Contr.*, vol. 8, pp. 338–353, 1965.

[3] M. Sugen and K. Murakami, "Fuzzy parking control of model car," in *Proc. 23rd IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 1984, pp. 902–903.

[4] O. Itoh, K. Gotoh, T. Nakayama, and S. Takamizawa, "Application of fuzzy control to activated sludge process," in *Proc. 2nd Int. Fuzzy Syst. Association Congr.*, Tokyo, Japan, July 1987, pp. 282–285.

[5] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller—Part I, II," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, pp. 404–432, Mar./Apr. 1990.

[6] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*. Norwell, MA: Kluwer, 1991.

[7] D. G. Schwartz and G. J. Klir, "Fuzzy logic flowers in Japan," *IEEE Spectrum*, vol. 29, no. 7, pp. 32–35, July 1992.

[8] P. K. Simpson, "Fuzzy min–max neural networks—Part 1: Classification," *IEEE Trans. Neural Networks*, vol. 3, pp. 776–786, Sept. 1994.

[9] S. Abe and M. S. Lan, "A method for fuzzy rules extraction directly from numerical data and its application to pattern classification," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 18–28, Feb. 1995.

[10] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 1414–1427, Nov. 1992.

[11] R. L. P. Chang and T. Pavlidis, "Fuzzy decision tree algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 28–35, Jan. 1977.

[12] E. T. Lee, "Fuzzy tree automata and syntactic pattern recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 445–449, July 1982.

[13] S. Morishima and H. Harashima, "Automatic rule extraction from statistical data and fuzzy tree search," *Syst. Comput. Jpn.*, vol. 19, no. 5, pp. 26–37, May 1988.

[14] M. Delgado, J. L. Verdegay, and M. A. Vila, "On fuzzy tree definition," *Eur. J. Oper. Res.*, vol. 22, no. 2, pp. 243–249, Nov. 1985.

[15] I. B. Turksen and Y. Tian, "Two-level tree search in fuzzy expert system," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 555–568, Apr. 1995.

[16] B. D. Liu and C. Y. Huang, "Design and implementation of the tree-based fuzzy logic controller," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, pp. 475–487, June 1997.

[17] C. L. Karr, "Design of an adaptive fuzzy logic controller using a genetic algorithm," in *Proc. 4th Int. Conf. Genetic Algorithms*, San Diego, CA, July 1991, pp. 450–457.

[18] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.

[19] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[20] L. B. Booker, D. E. Goldberg, and J. H. Holland, "Classifier systems and genetic algorithms," *Artif. Intell.*, vol. 40, no. 1–3, pp. 235–282, Sept. 1990.

[21] G. A. Vignaux and Z. Michalewicz, "A genetic algorithm for the linear transportation problem," *IEEE Trans. Syst. Man, Cybern.*, vol. 21, pp. 445–454, Mar./Apr. 1991.

[22] D. S. Weile, E. Michielssen, and D. E. Goldberg, "Genetic algorithm design of Pareto optimal broadband microwave absorbers," *IEEE Trans. Electromagn. Compat.*, vol. 38, pp. 518–524, Aug. 1996.

[23] C. H. Chang and Y. C. Wu, "The genetic algorithm based tuning method for symmetric membership functions of fuzzy logic control systems," in *Proc. IEEE Conf. Industrial Automatic Control Emerging Technological Applications*, Taipei, R.O.C., Dec. 1995, pp. 421–428.

[24] T. Iokibe, "A method for automatic rule and membership function generation by discretionary fuzzy performance function and its application to a practical system," in *Proc. 1st Int. Joint Conf. North American Fuzzy Information Processing Society Biannual Conf., Ind. Fuzzy Control Intelligient Systems Conf. NASA Joint Technology Workshop Neural Networks Fuzzy Logic*, San Antonio, TX, Dec. 1994, pp. 363–364.

[25] K. Hayashi, H. Nomura, and N. Wakami, "Acquisition of inferential rule by neural network drive type fuzzy reasoning," *J. SOFT*, vol. 2, no. 4, pp. 579–585, 1990.

[26] R. Krishnapuram, "Generation of membership functions via possibilistic clustering," in *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, Orlando, FL, June 1994, pp. 902–908.

[27] C. J. Kim and B. D. Russell, "Automatic generation of membership function and fuzzy rule using inductive reasoning," in *Proc. 3rd IEEE Int. Conf. Industrial Fuzzy Control Intelligient Systems*, Houston, TX, Dec. 1993, pp. 93–96.

[28] L. Chen, J. Yan, and Y. He, "A new approach for the automatic generation of membership functions and rules of multi-variable fuzzy system," in *Proc. IEEE Int. Conf. Neural Networks*, Perth, Australia, Nov. 1995, pp. 1342–1380.

[29] C. J. Wu and C. C. Li, "Automatically generated rules and membership functions for a neural fuzzy-based fault classifier," in *Proc. 37th Midwest Symp. Circuits Systems*, Los Angeles, CA, Aug. 1994, pp. 1377–1380.

[30] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 28–44, Jan. 1973.

[31] W. Banks, "Mixing crisp and fuzzy logic in applications," in *WESCON'94, Idea/Microelectronics Conf. Record*, Anaheim, CA, Sept. 1994, pp. 94–97.

[32] B. Bouchon-Meunier, "Linguistic hedges and fuzzy logic," in *Proc. 1st IEEE Int. Conf. Fuzzy Systems*, San Diego, CA, Mar. 1992, pp. 247–254.

[33] V. Novák, "A horizon shifting model of linguistic hedges for approximate reasoning," in *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, New Orleans, LA, Sept. 1996, pp. 423–427.

[34] N. C. Ho and W. Wechler, "Hedge algebras: An algebraic approach to structure of sets of linguistic truth values," *Fuzzy Sets Syst.*, vol. 35, no. 3, pp. 281–293, May 1990.

[35] ——, "Extended hedge algebras and their application to fuzzy logic," *Fuzzy Sets Syst.*, vol. 52, no. 3, pp. 259–281, Dec. 1992.

[36] C. Y. Huang, C. Y. Chen, and B. D. Liu, "Current-mode linguistic hedge circuit for adaptive fuzzy logic controllers," *Electron. Lett.*, vol. 31, no. 17, pp. 1517–1518, Aug. 1995.

[37] C. Y. Chen, C. Y. Huang, and B. D. Liu, "Current-mode fuzzy linguistic hedge circuit—Contrast intensification," in *Proc. IEEE Int. Symp. Circuits Systems*, vol. 3, Atlanta, GA, May 1996, pp. 511–513.

[38] C. Y. Chen, C. Y. Huang, B. D. Liu, and T. J. Su, "A current-mode fuzzy linguistic hedge circuit—More or less," in *Proc. IEEE Int. Conf. Fuzzy Systems*, New Orleans, LA, Sept. 1996, pp. 1080–1085.

[39] C. Y. Huang, C. Y. Chen, and B. D. Liu, "Current-mode fuzzy linguistic hedge circuits," *Analog Integr. Circuits Signal Process.*, vol. 19, no. 3, pp. 255–278, June 1999.

[40] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamic systems control," *IEEE Trans. Neural Networks*, vol. 6, pp. 144–156, Jan. 1995.

[41] B. Kosko, *Neural Networks and Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[42] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, pp. 834–846, Oct. 1983.

[43] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Trans. Comput.*, vol. C-26, pp. 1182–1191, Dec. 1977.

[44] K. A. De Jong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," Ph.D. dissertation, Dept. Comput. Sci., Univ. Michigan, Ann Arbor, 1975.

[45] K. A. De Jong, *Matrix Laboratory*. Natick, MA: Mathworks, 1992.

[46] P. H. Cannon, *Dynamics of Physical Systems*. New York: McGraw-Hill, 1967.

[47] T. S. Li and C. Y. Tsai, "Parallel fuzzy sliding mode control of the cart-pole system," in *Proc. IEEE IECON*, Orlando, FL, Nov. 1995, pp. 1468–1473.

[48] J. E. Slotine, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[49] J. Y. Hung, G. Senior, and J. C. Hung, "Variable structure control: A survey," *IEEE Trans. Ind. Electron.*, vol. 40, pp. 2–22, Feb. 1993.

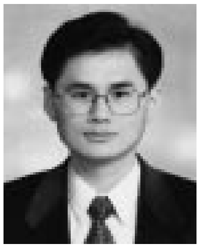
[50] W. Gao and J. C. Hung, "Variable structure control of nonlinear systems: A new approach," *IEEE Trans. Ind. Electron.*, vol. 40, pp. 45–55, Feb. 1993.

[51] J. C. Hung, *MBSP Manual*. Hsinchu, Taiwan, R.O.C.: Phimatic Enterprise Co., Ltd., 1995.

Bin-Da Liu (S'79–M'82–SM'95) received the B.S., M.S., and Ph.D. degrees all in electrical engineering from the National Cheng Kung University (NCKU), Tainan, Taiwan, R.O.C., in 1973, 1975, and 1983, respectively.

From 1975 to 1977, he served as Electrical Officer in the Combined Service Forces. Since 1977 he has been on the faculty of NCKU, where he is currently Professor and Chair of the Electrical Engineering Department. From 1983 to 1984, he was a Visiting Assistant Professor in the Department of Computer Science, University of Illinois, Urbana-Champaign. From 1988 to 1992, he was the Director of Electrical Laboratories, NCKU. From 1996 to 1999, he was Associate Chair of the Electrical Engineering Department. From 1990 to 1993, he was appointed a member of the Evaluation Committee for Junior Engineering College, Ministry of Education, R.O.C. Since 1995, he has been a consultant of the Chip Implementation Center, National Science Council, R.O.C. His current research interests include physical design and testing for VLSI circuits, high-voltage integrated circuit design, and VLSI implementation for fuzzy-neural networks and video signal processors. He has published more than 140 technical papers.

Dr. Liu is a member of Phi Tau Phi, International Union of Radio Science, Chinese Fuzzy Systems Association, and the Chinese Institute of Electrical Engineers. He received the Dragon Distinguished Paper Award from the Acer Foundation in 1991 and 1997, and the Best Paper Award from the CIEE in 1995. Since 1988 he has received the Research Award from the National Science Council every year. He served as member of the Program Committee of the 1998 and 1999 IEEE Workshop on VLSI Signal Processing Systems. He also served as member of the Technical Program Committee of the 1998 and 2000 IEEE Asia Pacific Conference on Circuits and Systems, and the First and Second IEEE Asia Pacific Conference on ASIC's. Since 1992, he has served as member of the Technical Program Committee of the R.O.C. VLSI Design/CAD Symposium and served as the Program Chair in 1994. Since 1997 he has served as member of the Technical Program Committee of the R.O.C. National Conference on Fuzzy Theory and Its Applications. He organized the R.O.C. Student VLSI Design Contest in 1998, 1999, and 2000.



Chuen-Yau Chen (S'00) received the B.S. degree in electrical engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, R.O.C., in 1995. Since September 1996, he has been pursuing the Ph.D. degree in electrical engineering at NCKU.

From 1999 to 2000, he was the System Administrator of the VLSI/CAD Laboratory at the Department of Electrical Engineering, NCKU. His current research interests include design of fuzzy logic controller as well as design of current-mode and mixed-signal circuits and systems. He has also dealt with applications of genetic algorithm and guided-evolution simulated-annealing algorithm.

Ju-Ying Tsao received the B.S. degree in electronic engineering from Chung Yuan Christian University, Taiwan, R.O.C., and the M.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1996 and 1998, respectively.

From 1998 to 1999, she was the Applications Engineer of Cadence Branch Office in Hsinchu, Taiwan. In 2000, she joined EE-Solutions, Inc., Hsinchu, where she is a member of the consulting staff of the EDA department. She focuses on the physical design portion of the Ultra Deep Submicron (UDSM) SOC design flow. Her current expertise is the analysis of UDSM effects that may cause chip failures such as IR drop, substrate noise integrity, and signal integrity.