A Fuzzy Inference Engine in Nonlinear Analog Mode and Its Application to a Fuzzy Logic Control

Takeshi Yamakawa
Invited Paper

Abstract— **This paper is a tutorial to enlighten outsiders or beginners on a utility of a fuzzy system by providing broad scope overview, especially analog mode hardware, with the author's original work. At first, the difference between deterministic words and fuzzy words is explained as well as fuzzy logic. The description of the system by using mathematical equations, linguistic rules and parameter distribution (neural networks) are discussed. The algorithm of fuzzy inference and defuzzification is presented and their hardware implementation is discussed in detail together with an advanced one. The fuzzy logic controller was applied to stabilize a glass with wine and a mouse moving around the plate on the tip of the inverted pendulum.**

I. Introduction

HUMAN beings make tools for their use and also think to control the tools as they desire. A feedback concept is a very important concept to achieve the control of the tools. The first significant work in feedback application was James Watt's flyball governor developed in 1769 for controlling the speed of a steam engine. As modern plants with many inputs and outputs become more and more complex, the description of a modern control system requires a large number of equations. Classical control theory, which deals only with single-input-single-output systems, becomes entirely powerless for multiple-input-multiple-output systems. Since about 1960, modern control theory has been developed to cope with the increased complexity of modern plants and the stringent requirements on accuracy, weight, and cost in military, space, and industrial applications [1]–[3]. This development has been accelerated by a digital computer because it facilitates a solution of simultaneous equations of many unknowns.

Control technologies described above are all based on mathematical equations such as differential equations and relational equations. We, however, occasionally face chemical plants, machines and some other systems to be controlled, the characteristics of which are very difficult to describe with mathematical equations because of their complexity. Even in these cases, human experts may achieve the control by know-hows or control rules which are squeezed out from their long experience and represented by intuitive natural language. For instance, "If the pressure in the chamber *goes high*, then the fuel supply should be *reduced significantly* and the valve should be *a little bit opened.*" Knowledges (know-hows) represented with the intuitive natural language is easily explained and understood by common sense and thus easy to remember. Further more, the number of knowledges is drastically reduced by employing intuitive natural language.

In most cases, the intuitive natural language has its ambiguous boundary of meaning, while numerical values and deterministic linguistic terms used in artificial intelligence are well-defined. This type of natural language is referred to as a *fuzzy linguistic term* and it is characterized by a so called *membership function.* This concept was presented as *fuzzy sets* (strictly speaking, it is fuzzy subsets) by L. A. Zadeh in 1965 [4].

A software or hardware system, which gives a conclusion (output) from a fact (input) and knowledges (control rules), is called an inference engine. If the knowledges include fuzzy linguistic terms, it is referred to as a *fuzzy inference engine* and can be utilized to a *fuzzy logic control.*

In this paper, a fuzzy inference (approximate reasoning) based on fuzzy sets and fuzzy logic and a defuzzification are briefly explained in comparison with other technologies and their hardware implementation in "nonlinear" analog circuits is described. The fuzzy inference engine and defuzzifier are employed to control a liquid-contained inverted pendulum, the dynamics of which is very difficult, if not impossible, to describe with mathematical equations.

II. Fuzzy Sets and Fuzzy Logic

A. Fuzzy Sets [4]–[28]

Linguistic terms and numerical values are classified to three categories in accordance with their meanings which are defined by the characteristic functions.

Deterministic words, e.g., "male" and "female," "dead," and "alive," the personal name "John," have truth values of 0 or 1 corresponding to NO or YES, respectively. In other words, if one is asked, "Are you male?," "Are you alive?," "Is your name John?," then answer can be made with "YES" or "NO." Exact numerical value, e.g., "exactly 26°C, "concentration of 0.1 Mol/1," "38g" etc., are in the same situation. These deterministic words and numerical values have neither flexibilities nor intervals. Those meanings are characterized by a characteristic function as shown in Fig. 1(a). The word "exactly 26°C" means only one single point

Manuscript received September 10, 1992.
The author is with the Department of Control Engineering and Science, Kyushu Institute of Technology, Iizuka, Fukuoka, 820 Japan.
IEEE Log Number 9207196.

1045-9227/93$03.00 © 1993 IEEE

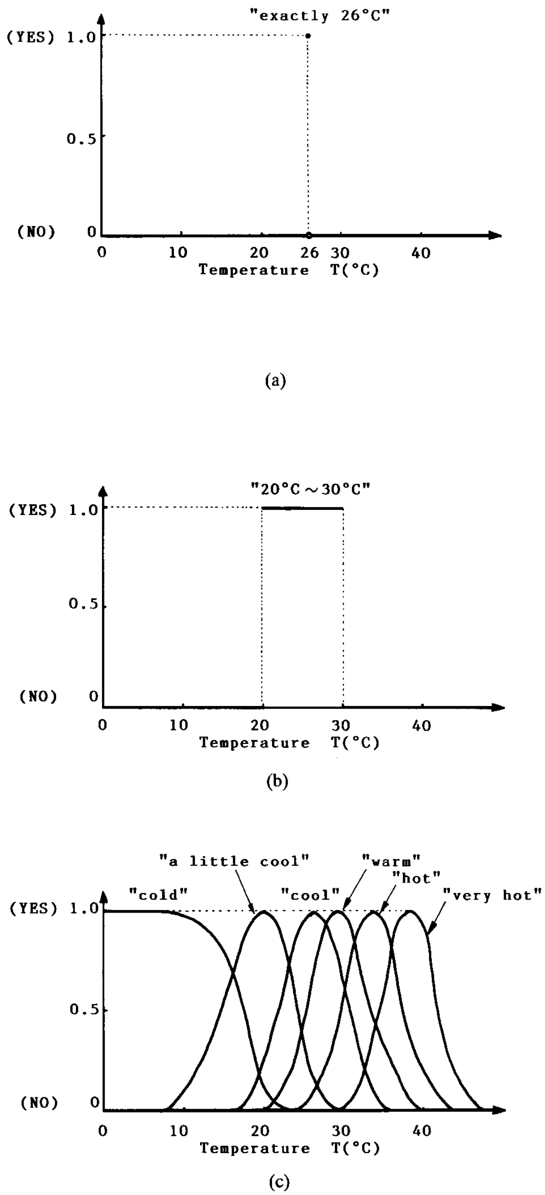


Fig. 1. (a) A singleton, (b) a crisp interval (crisp sets), and (c) fuzzy sets.

of temperature at 26°C, so that this type of term or value is called a *singleton.*

Even in the scientific analysis, where exact values are preferred, numerical intervals are sometimes used to represent flexible values. For instance, "The comfortable room temperature for human beings is 20°C–30°C." Temperatures 19.9°C, and 30.0001°C are not the case. Thus the characteristic function representing a numerical interval 20°C–30°C" is shown in Fig. 1(b). Truth values for any temperatures are given by YES or NO in this case as well as in a singleton. This interval can be regarded as a set of numerous singletons. Therefore this type of deterministic interval is called *crisp sets.* Crisp sets are also adopted to represent linguistic terms in knowledges in artificial intelligence (AI).

In our daily life, fuzzy natural languages are used for easy and efficient communication. Although these terms are usually intuitive and includes some kinds of uncertainties, they are very easy to select for practical use. For example, "Be careful when you carry *important* documents." "*High* accuracy is a measure of the technology." "Cool it *a little bit more* and a solid will be deposited." The meaning of the fuzzy linguistic term is defined by a characteristic function as shown in Fig. 1(c). This function is specifically called a *membership function,* because it indicates a grade of membership of each element (physical value in the horizontal axis) in a fuzzy linguistic term of interest. For instance, a fuzzy linguistic terms "cold," "a little cool," "cool," "warm," "hot," and "very hot" are indicated in Fig. 1(c). According to a common sense, a grade of membership of 20°C in "a little cool" is undoubtedly 1.0. In other words, you can answer "YES" to the question, "Is a temperature of 20°C included in 'a little cool'?" On the other hand, how about 0°C, 5°C, 40°C, etc.? You may answer "NO" to this question, i.e., grades of membership of 0°C, 5°C, 40°C in "a little cool" are 0. How about 14°C and 23°C? You can not give the answer "NO" nor "YES" to the same question. The grades will be answered to be 0.4 and 0.8 corresponding to 14°C and 23°C, respectively. Of course, strictly speaking, these grades are given by intuition or common sense, so that the shape of membership function changes a little from person to person. In any way, a fuzzy linguistic term can be defined by a membership function, and the membership function exhibits a continuous curve changing from 0 to 1 or vice versa. And this transition region represents a fuzzy boundary of the term. If the fuzzy linguistic term includes a numerical value, e.g., "around 20°C," "much higher than 30%," it is called a *fuzzy number.*

These fuzzy linguistic terms can be regarded as sets of singletons, the grades of which are not only 1 but also ranging from 0 to 1. Therefore, these fuzzy linguistic terms are called *fuzzy sets.* Each singleton is an *element* of the fuzzy sets. A fuzzy linguistic term, elements of which are ordered in the universe of discourse, is called a *fuzzy interval.* An interval on horizontal axis where grades of membership are not zero is called *support.*

Fuzzy sets are defined by *labels* (e.g., high pressure, around 20°C, a little, usually) and *membership functions.* While a modern AI achieves only a symbolic processing with labels, a fuzzy information processing of a future artificial intelligence achieves both of a *symbolic processing* with labels and a *meaning processing* with membership functions.

A membership function can be represented by a piecewise linear function, because important is the continuity between 0 and 1 rather than the curvature of the membership function. Parameters characterizing the shape and the label of the membership function can be also obtained by learning, if needed.

B. Fuzzy Logic [5]–[28]

Fuzzy logic functions which are most popular in engineering field are NOT (Fuzzy Logic Complement), MIN (Minimum; Fuzzy Logic Product, Fuzzy Logic Intersection) and MAX (Maximum; Fuzzy Logic Sum, Fuzzy Logic Union).

Fuzzy Logic Complement of a fuzzy linguistic term "A" is defined by $1 - \mu_{\mathrm{A}}$ and means negation of "A," where μ_{A} is a membership function of "A." Negation of "hot" is "not hot," and it is easily understood, by Fig. 1(c), that "not hot" does not necessarily mean "cold."

The most popular fuzzy logic functions which implement logical "AND" and logical "OR" are MIN and MAX, respectively, and defined by

$$\mathrm{Min}\{\mu_{\mathrm{A}}, \mu_{\mathrm{B}}, \mu_{\mathrm{C}}, \cdots, \mu_{\mathrm{Z}}\} = \mu_{\mathrm{A}} \wedge \mu_{\mathrm{B}} \wedge \mu_{\mathrm{C}} \wedge \cdots \wedge \mu_{\mathrm{Z}} = \mu_{\mathrm{K}} \quad (\mu_{\mathrm{K}} \leq \mu_{\mathrm{A}}, \mu_{\mathrm{B}}, \mu_{\mathrm{C}}, \cdots, \mu_{\mathrm{Z}}) \tag{1}$$

$$\mathrm{Max}\{\mu_{\mathrm{A}}, \mu_{\mathrm{B}}, \mu_{\mathrm{C}}, \cdots, \mu_{\mathrm{Z}}\} = \mu_{\mathrm{A}} \vee \mu_{\mathrm{B}} \vee \mu_{\mathrm{C}} \vee \cdots \vee \mu_{\mathrm{Z}} = \mu_{\mathrm{L}} \quad (\mu_{\mathrm{L}} \geq \mu_{\mathrm{A}}, \mu_{\mathrm{B}}, \mu_{\mathrm{C}}, \cdots, \mu_{\mathrm{Z}}) \tag{2}$$

where $\mu_{\mathrm{A}}, \mu_{\mathrm{B}}, \mu_{\mathrm{C}}, \cdots, \mu_{\mathrm{Z}}$ are grades of membership function ranging from 0 to 1.

III. Modeling of the System

In order to control the system of interest, we have to describe the precise behavior of the system to design the controller. In order to use the knowledge, we have to describe the knowledge effectively. The author classifies the description of the system or knowledge to three categories: mathematical equations, linguistic rules and artificial neural networks.

A. Mathematical Equations

The traditional sciences have been based on mathematical equations. Natural scientific phenomena and physical behavior of the artificial system can be modeled with relational equations or differential equations. These equations describe the dynamics or kinetics of the systems or the knowledge about the system in a very simple form. If the relation between the input x and the output $f(x)$ of the system or the relation between the cause x and the result $f(x)$ is obtained as shown in Fig. 2(a) from experiments, $f(x)$ is described as

$$f(x) = \tfrac{1}{30}(x-3)^2. \tag{3}$$

By substitution of the numerical value of x (input value, the fact or the premise) to this equation, the numerical value of $f(x)$ (the output value or the conclusion) is obtained.

A description of the system with this type of equation exhibits significant simplicity. It is, however, very difficult to identify an exact equation from the given relation, especially in the case of many variables. Furthermore, it is also very difficult to reassign this equation, when the relation between x and $f(x)$ is changed. Therefore, this description is not so suitable for complex systems such as nonlinear systems or time-varient systems. As the complexity of the system increases, the possibility to describe the system with mathematical equations diminishes.

B. Linguistic Rules

A relation between x and $f(x)$ can be described with a set of linguistic rules, typical form of which is

$$\text{Rule i:} \quad \text{If } x \text{ is } A_i, \text{ then } f(x) \text{ is } B_i, \qquad (i = 1, 2, \cdots, N) \tag{4}$$

where x and $f(x)$ are independent and dependent variables, respectively, A_i and B_i linguistic constants, and N the number of experimental data. These rules are referred to as *IF-THEN rules* because of their form. An if-clause is referred to as an *antecedent* and a then-clause as a *consequent.*

Linguistic rules can be classified to two categories with respect to linguistic constants A_i and B_i.

1) Linguistic Rules with Well-defined Languages (Crisp IF-THEN Rules): In this category, all the linguistic constants are represented by well-defined languages (crisp information or exact numerical values), e.g., male, 80°C, 60°C–80°C, 98 g, etc., as shown in Fig. 1(a) and (b). Modern AI belongs to this category. Here we have an example shown in Fig. 2(b), which gives the following crisp rules.

Rule 1	If x is -2, then $f(x)$ is $25/30$
Rule 2	If x is -1, then $f(x)$ is $16/30$
Rule 3	If x is 0, then $f(x)$ is $9/30$
Rule 4	If x is 1, then $f(x)$ is $4/30$
Rule 5	If x is 2, then $f(x)$ is $1/30$
Rule 6	If x is 3, then $f(x)$ is 0
Rule 7	If x is 4, then $f(x)$ is $1/30$
Rule 8	If x is 5, then $f(x)$ is $4/30$
Rule 9	If x is 6, then $f(x)$ is $9/30$
Rule 10	If x is 7, then $f(x)$ is $16/30$
Rule 11	If x is 8, then $f(x)$ is $25/30$. (5)

The advantage of this description is that it is very easy to change the description of the system. For instance, when the system is locally changed as indicated by squares shown in Fig. 2(b), we have only to change the corresponding values of the consequents in the IF-THEN rules (5), (e.g., Rule 7: 1/30 to 1/20, Rule 8: 4/30 to 10/57, Rule 9: 9/30 to 2/5, Rule 10: 16/30 to 5/9) because all the rules are independent of each other. This means that IF-THEN rules are suitable for learning systems, self-organizing systems, adaptive systems, and so on.

On the other hand, there are some disadvantages. When the fact $x = +1$ is given, the conclusion can be obtained to be 4/30 from the data-matching between the fact and the antecedents (if-clause) of rules. This procedure to produce the conclusion is called *inference.* However, the fact $x = +1.5$ cannot give the conclusion from the rules, because there is no antecedent which is exactly matched to the fact $x = +1.5$. This shows that the inference with crisp rules is very weak against a defect of knowledge, a defective fact, a noisy defect or a variation of a fact, and that it needs a large-scale knowledge base for significant performance. Therefore, this type of inference wastes time because of a sequential data-matching between the fact and a large data base.

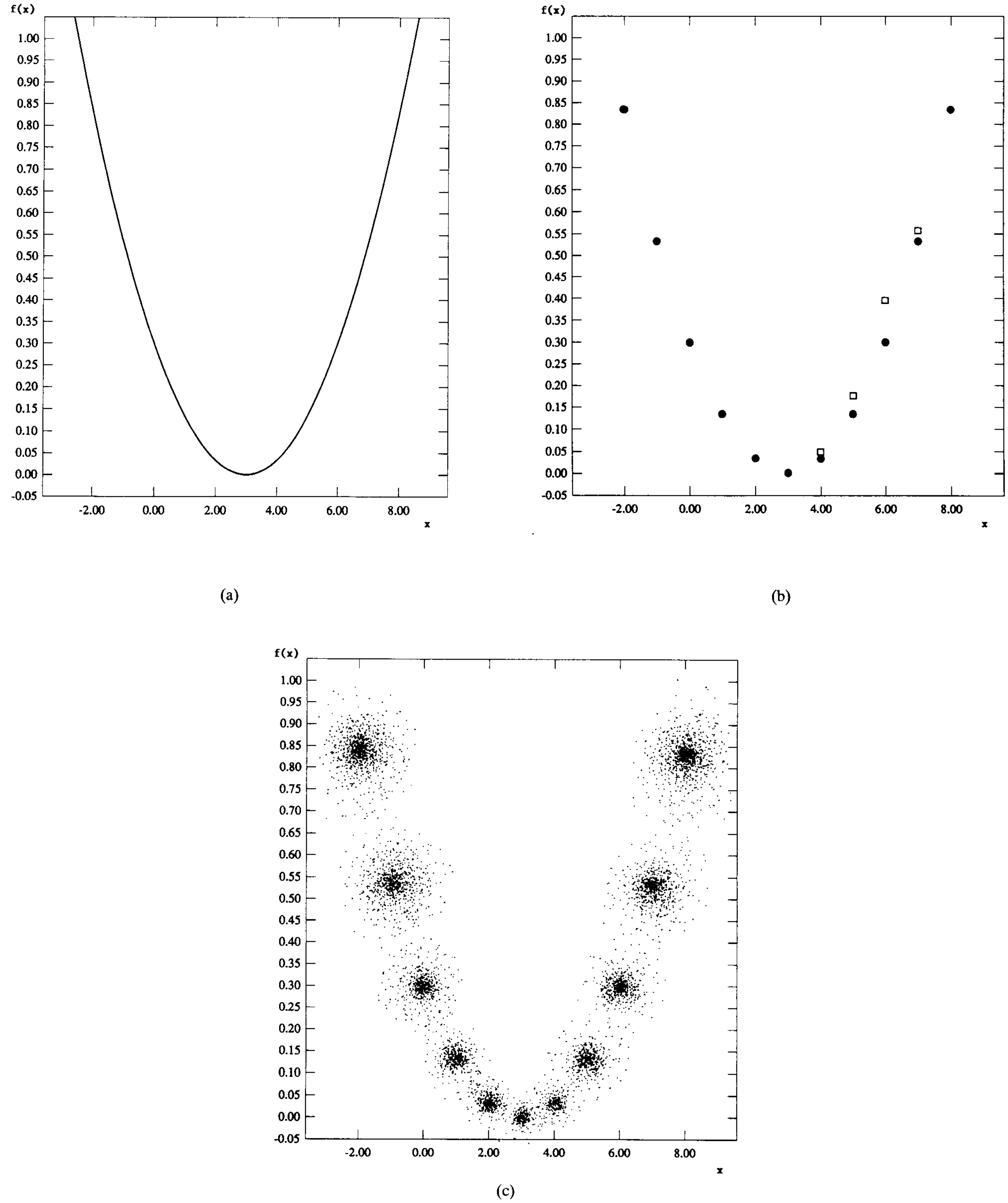


Fig. 2. (a) Example of the system description by a mathematical equation. (b) Example of the system description by a set of linguistic rules with well-defined languages (crisp rules). This description has the advantage of easy reassignment of the rules, when the system is changed as shown by squares. (c) Example of the system description by a set of linguistic rules with ill-defined languages (fuzzy IF-THEN rules, or shortly fuzzy rules). Fuzzy rules give a continuous relationship between x and $f(x)$ through fuzzy inference and defuzzification.

Furthermore, this inference system should not include contradictory rules in a knowledge base. Otherwise, it will produce two contradictory conclusions from one fact.

These disadvantages appear in the modern AI system as well. This type of inference is based only on *symbolic processing* but not on the *processing of the meaning* of linguistic terms.

2) Linguistic Rules with Ill-Defined Languages (Fuzzy IF-THEN Rules): In this category, all the linguistic constraints are represented by ill-defined languages (fuzzy information, uncertain words, approximate numerical values, etc.), e.g., beautiful lady, high temperature, heavy, low power, around 45 kg, etc. The linguistic rules (fuzzy IF-THEN rules or, shortly fuzzy rules) can describe the relation between x and $f(x)$, for example, as shown in Fig. 2(c), and are given in the following form.

$$\begin{array}{lll} \text{Rule 1} & \text{If } x \text{ is } around\ -2, \text{ then } f(x) \text{ is } around\ 25/30 \\ \text{Rule 2} & \text{If } x \text{ is } around\ -1, \text{ then } f(x) \text{ is } around\ 16/30 \\ \text{Rule 3} & \text{If } x \text{ is } around\ \ 0, \text{ then } f(x) \text{ is } around\ \ 9/30 \\ \vdots & \qquad\vdots & \qquad\qquad\vdots \\ \text{Rule 11} & \text{If } x \text{ is } around\ \ 8, \text{ then } f(x) \text{ is } around\ 25/30. \end{array} \tag{6}$$

A crisp relation between x and $f(x)$ in Fig. 2(b) is fuzzified to make it continuous as shown in Fig. 2(c). This continuous fuzzified relation (*fuzzy relation*) gives reasonable conclusions for any fact, e.g., $x = -1.5, +3.2, +4.3$, etc., through *fuzzy inference* and *defuzzification,* which are precisely described in Section IV-A. FUZZY INFERENCE (APPROXIMATE REASONING). In other words, fuzzy inference and defuzzification facilitate a reasonable interpolation with much less data. So that the fuzzy inference is called *interpolative inference* and can save the memory hardware. This fuzzy inference with fuzzy rules exhibits the similar function to that of a mathematical equation.

It is much easier to reassign the fuzzy rules rather than the mathematical equation, when the characteristics of the system are changed. In other words, only one or more rules should be added or revised independently of other rules, while the order and the numerous coefficients of the mathematical equation should be recalculated from simultaneous equations.

The distinctive features of this description are as follows.

1. It is suitable for describing a complicated system with a small amount of knowledge.
2. It is easy to select the words to be used in fuzzy rules among the categorized few words.
3. It is easy to remember the knowledge.
4. It is easy for designers to communicate with others by using fuzzy natural languages.

While ordinary inference (modern AI) is based on symbolic processing, fuzzy inference or approximate reasoning is based on both of symbolic processing and meaning processing. Linguistic rules (crisp rules and fuzzy rules) can represent the algorithm of inference explicitly, while an artificial neural network represents it implicitly. So that linguistic rules are called *structured* and an artificial neural network is *unstructured.*

C. Artificial Neural Networks

A system can be described by a distribution of parameters, the typical example of which is a well-known artificial neural network.

An architecture of a neural network is a simple iteration of simple aggregating elements. This aggregating element is a model of a physical neuron in the neural network in the living body. When w_{ij} $(i = 1, 2, \cdots, n)$ is a weight assigned to the signal input p_i to the jth cell and θ_j and q_j are a threshold level and a signal output of the jth cell, respectively, the signal output is characterized by

$$q_j = h\left(\sum_i w_{ij} \cdot p_i - \theta_j\right) \tag{7}$$

where h is a sigmoid function and it is typically described by

$$h(x) = \frac{1}{1 + \exp(-x)}. \tag{8}$$

By employing this neuron model, the relation between x and *f(x)* can be characterized by a neural network as shown in Fig. 3(a), where the parameter distribution such as threshold levels θ_j, weights w_{ij} are obtained, after 10 000 leanings, by the same amount of data (11 data) to IF-THEN rules (5) and (6). The input-output characteristics of this artificial neural net (Fig. 3(a)) is calculated and shown in Fig. 3(b), which shows the ability of interpolation similar to fuzzy inference. It is, however, very difficult to understand, by looking at the parameter distribution, how the neural net behaves. Furthermore, we can not estimate the number of neurons and layers necessary to achieve pattern recognitions or to solve the problems and there is no guarantee of convergence of learning. If the system under modeling changes its behavior, the similar learnings should be applied again to the neural net to reassign the new distribution of weights and thresholds, and sometimes we cannot reach the reasonable goal. In other words, there is less designability in neural networks than in fuzzy systems. This point is one of the reasons why fuzzy systems have been applied to practical use much more than neural nets in Japan.

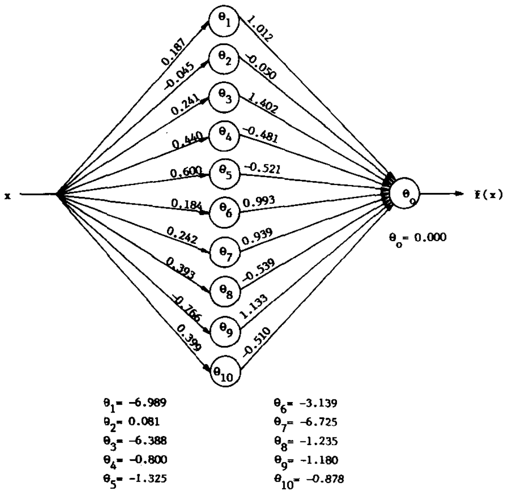
IV. A Fuzzy Logic Controller

In order to design a controller, the control algorithm should be described by some means. A classical control theory gives differential equations or transfer functions and a modern control theory gives a first-order vector matrix differential equation based on the state-space method. In these approaches, a controller designer has to possess knowledges about mathematics and the system under control.

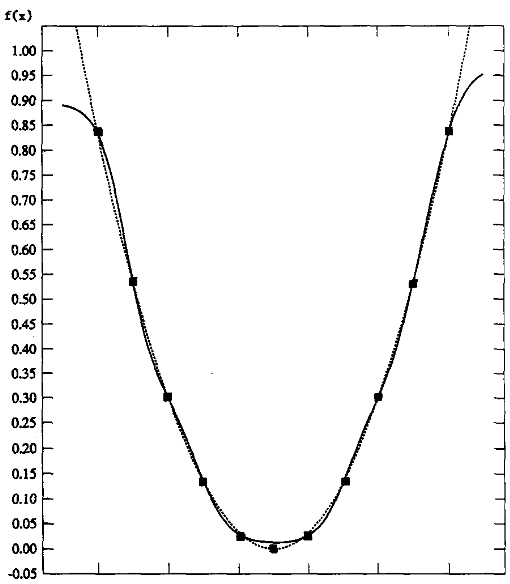
However, human experts of ripe experience can skillfully control plants, machines, vehicles, etc., even though these systems under control are very complex (time-varient and nonlinear). These human experts mostly utilize know-hows which have been summarized from a long experience including successes and faults, and are represented with IF-THEN rules including fuzzy linguistic terms. They very often succeed to control the systems reasonably with these inexact information and without any calculations such as sin ωt, cos ωt, $\exp(x)$. This suggests that there is another algorithm which facilitates to control a complicated system by a simple and inexact knowledge base. One candidate is a *fuzzy inference* which produces a conclusion from a *knowledge base* (a set of fuzzy rules) and a *fact.*

A. Fuzzy Inference (Approximate Reasoning) [5]–[28]

The algorithm of the fuzzy inference in case of a deterministic fact is illustrated in the following. Let us consider the control of air conditioner, which is very primitive and



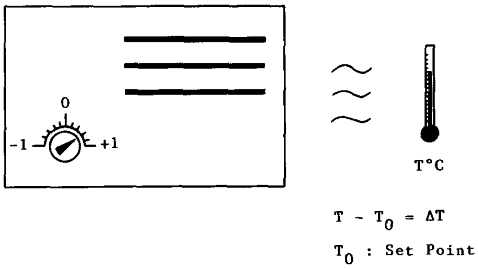
(a)



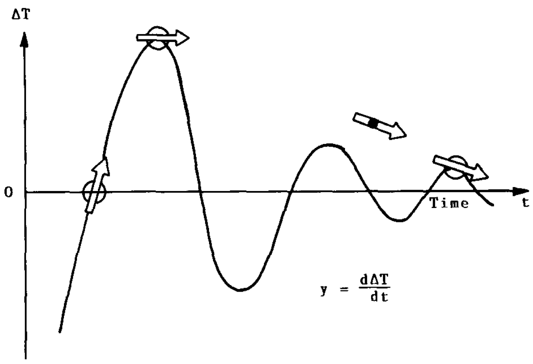
(b)

Fig. 3. (a) Example of the system description by a distribution of parameters, e.g., synaptic weights and thresholds in an artificial neural network. However, nobody can estimate, at a glance, how the output $f(x)$ responds to the input x. (b) The input-output characteristics (solid line) of the artificial neural net which was established after 10 000 times of learning by eleven data (squares) same to Fig. 2(b) and (c). The dotted line shows the system under modeling.

possesses no feedback outside. It blows off the air, the temperature and the volume of which can be changed by the dial (but possibly not proportional to the dial) as shown in Fig. 4(a). When the dial is assigned to be positive, warm or hot air is supplied from the air conditioner and when it is assigned to be negative, cool or cold air is supplied. Zero means no air supply. A room temperature T°C is measured with a thermometer as a deterministic value. A human looks



(a)



(b)

Fig. 4. (a) A primitive air conditioner and (b) a typical transient response.

at this value and notices the difference ΔT°C between the room temperature and the temperature T_0°C at which the room is desired to be kept. He or she must have a "strategy" to keep the room temperature constant even though the number of persons in the room changes, the outside temperature is changed, location of furnitures is changed and so on. By the strategy, he or she will reasonably changes the dial on the panel of the air conditioner in order to supply the warm or cool air to compensate the temperature change. A typical transient response of the room temperature is shown in Fig. 4(b), where the vertical axis represents the temperature difference ΔT°C, i.e., zero level is a set point of the room temperature.

The strategy may be written by sentences including fuzzy linguistic terms as follows. 1) When the room temperature is approximately equal to the set point and the temperature is rapidly changing higher, i.e., ΔT is *approximately zero* and the temperature change $y = d\Delta T/dt$ is *positively large* (as indicated by the left circle and arrow in Fig. 4(b)), cold air should be blown off rapidly to suppress the increasing temperature, otherwise it will positively deviate from the set point. Thus the dial should be turned to *negative large* (or approximately -1). 2) When the room temperature is high and the temperature does not change, i.e., ΔT is *positively large* and the temperature change $y = d\Delta T/dt$ is *approximately zero* (as indicated by the second circle and arrow in Fig. 4(b)), cold air should be blown off intermediately to decrease the temperature. Thus the dial should be turned to *negative medium* (or

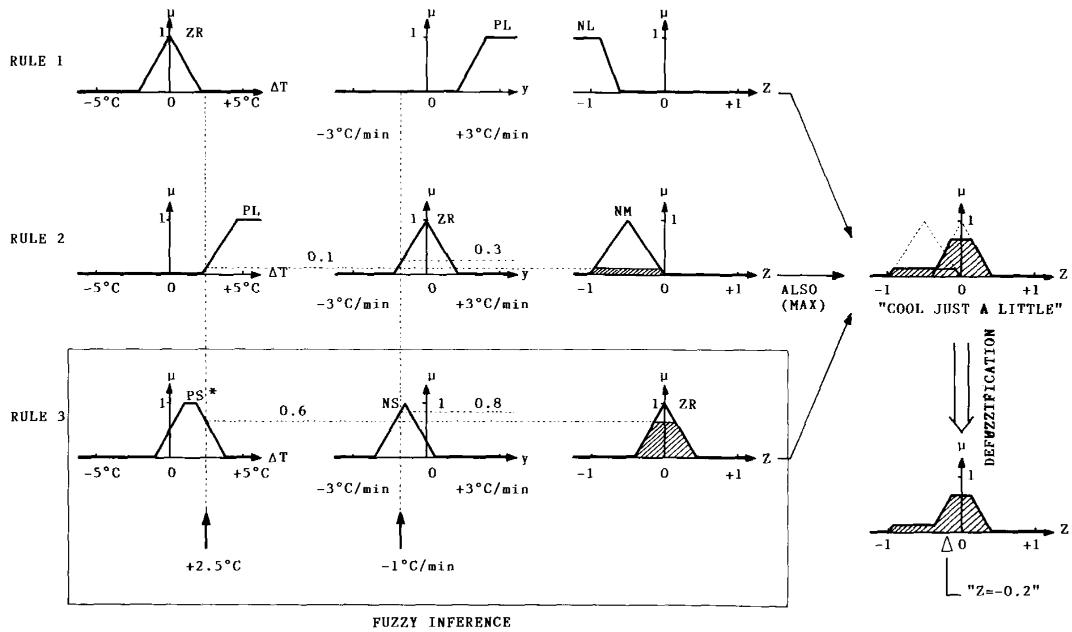


Fig. 5. An aspect of an individual fuzzy inference, aggregation (ALSO) and defuzzification.

approximately -0.5). 3) When the room temperature is a little bit higher than the set point and the temperature is gradually decreasing, i.e., ΔT is *positively small* and the temperature change $y = d\Delta T/dt$ is negatively small (as indicated by the right circle and arrow in Fig. 4(b)), there is no need to supply the cold nor hot air, because the room temperature is asymptotically close to the set point. Thus the dial should be turned to *approximately zero.* Of course, the strategy includes more than three knowledges described above. The other ones are abbreviated here for the simplicity.

The above strategy of temperature control can be rewritten by fuzzy rules as follows, together with a fact (measured values of $\Delta T = +2.5°\mathrm{C}$, $y = -1°\mathrm{C/min}$ indicated by the dot and arrow in Fig. 4(b)), as shown at the bottom of the page, where ΔT and y are variables in an if-clause (antecedent) and "the dial" is a variable in a then-clause (consequent), respectively, in a fuzzy rule, and NL, NM, NS, ZR, PS, and PL are fuzzy constans or fuzzy values and are abbreviations of fuzzy linguistic terms, *negatively large, negatively medium, negatively small* and *approximately zero, positively small* and *positively large,* respectively. When the room temperature is obtained by a thermometer to be 2.5°C higher than the set point and the temperature is decreasing at 1°C/min, what degree do we have to assign the dial to make the room temperature to be close to the set point?

Fig. 5 illustrates the algorithm of fuzzy inference to obtain the dial assignment from the strategy. A fuzzy linguistic term is identified by a label and characterized by a membership function as described in Section II-A. FUZZY SETS, so that a set of rules (9) can be rewritten as Fig. 5 where all the membership functions are defined by piecewise linear function for the simplicity of hardware design. The shapes (slopes, peak points, supports, and shoulders) of the membership functions are assigned by an intuition of a human expert after summarization of his wealth of experience, or assigned by an intuition of a designer of a fuzzy logic controller who has a knowledge about the air conditioner, heat transfer, etc.

Measured deterministic values $\Delta T = +2.5°\mathrm{C}$ and $y = -1°\mathrm{C/min}$ are referred to the antecedent (ΔT is PS* and y is NS) of Rule 3 in order to obtain the compatibility of the fact to the antecedent, where "*" in Fig. 5 represents a trapezoidal membership function, otherwise membership functions are S-

$$
\begin{array}{ll}
\text{Rule 1} & \text{If } \Delta T \text{ is } ZR \text{ and } y \text{ is } PL, \text{ then the dial should be } NL \\
\text{Rule 2} & \text{If } \Delta T \text{ is } PL \text{ and } y \text{ is } ZR, \text{ then the dial should be } NM \\
\text{Rule 3} & \text{If } \Delta T \text{ is } PS \text{ and } y \text{ is } NS, \text{ then the dial should be } ZR \\
\vdots & \quad \vdots \qquad \vdots \qquad \vdots \\
\text{Fact} & \Delta T \text{ is } +2.5°\mathrm{C} \text{ and } y \text{ is } -1°\mathrm{C/min} \\
\hline
\text{Conclusion} & \qquad\qquad \text{the dial should be ?}
\end{array} \tag{9}
$$

shaped, Z-shaped or triangular. The compatibility is one kind of similarity measure. The compatibility of +2.5°C to PS* and that of −1°C/min to NS are 0.6 and 0.8, respectively. Variables in the antecedent are combined by a conjunction "and," so that the constraint of the antecedent is more severe than the individual variable. So that it is reasonable to evaluate the severe compatibility of the fact, "$\Delta T = +2.5$°C and $y = -1$°C/min," to the antecedent, "ΔT is PS* and y is NS," to be a smaller grade of membership 0.6. This grade is a degree of soft matching between the measured value and the condition of the rule. When a conjunction "or" is used, the bigger compatibility may be adopted as a soft matching degree.

If it is 1, the condition is fully satisfied and thus the then-clause should be entirely adopted. Alternatively, if the degree of soft matching is 0, the then-clause should be rejected. In the fuzzy inference, the degree of soft matching can range from 0 to 1, so that the degree of adopting the then-clause may range from 0 to 1, while it is 0 or 1 in case of unification in AI technology. Since the soft matching degree in this case is 0.6, the then-clause is partially adopted, i.e., the fuzzy constant ZR in the consequent in Rule 3 is weighted by 0.6. There are some weighting methods such as multiplication, minimization, etc. The latter is adopted in this paper. As shown in Fig. 5, the membership function of "dial is ZR" is truncated (minimized) by 0.6 to be an individual conclusion (shaded part) from Rule 3. The procedure to obtain the individual conclusion from each fuzzy rule and the fact is called *fuzzy inference,* and the tool to achieve the fuzzy inference is called a *fuzzy inference engine.* The fuzzy inference is also referred to *approximate reasoning,* because it gives an approximate conclusion represented with a membership function.

There is another rule contributing to produce the final conclusion. By referring measured values to the antecedent of Rule 2, the another individual conclusion (shaded part) is obtained after truncating "dial is NM" by 0.1. Concerning Rule 1, the measured value $y = -1$°C/min exhibits the compatibility 0 to "y is PL," so that the degree of soft matching between the measured value and the condition of Rule 1 is 0. Therefore, we cannot obtain any contribution to the final conclusion from Rule 1. All the rules are implicitly combined by conjunctions "also." Thus the individual conclusions should be aggregated to obtain the final conclusion. The "also" can be interpreted to be a maximization of individual conclusions and the final conclusion is obtained as a trapezoidal membership function with a side lobe as shown in Fig. 5.

B. Defuzzification

A controller employing a fuzzy inference is referred to as a *fuzzy logic controller* (or shortly, *fuzzy controller.*) A fuzzy logic controller delivers a deterministic signal but not a fuzzy signal to the system under control. In case of the air conditioner, we have to assign the dial. The procedure to obtain a deterministic value, on the universe of discourse, from a fuzzy value (membership function) is called *defuzzification* and its tool a *defuzzifier.* The most popular way of defuzzification is a *center-of-gravity method,* or a *centroid method* [22], [58].

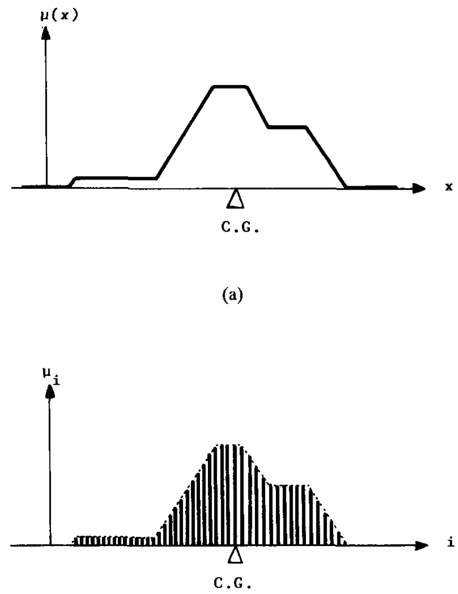


Fig. 6. Defuzzification (center-of-gravity method) of (a) a continuous membership function and (b) a discrete membership function.

A membership function is very often represented by a sampled data (a set of elements) on the universe of discourse as well as a continuous one as shown in Fig. 6. The center-of-gravity can be calculated from the following equations depending upon whether it is continuous or discrete.

$$\text{C.G.} = \frac{\int x\mu(x)\,dx}{\int \mu(x)\,dx} \quad \text{(for a continuous membership function)} \tag{10}$$

$$\text{C.G.} = \frac{\sum_{i=1}^{n} i \cdot \mu_i}{\sum_{i=1}^{n} \mu_i} \quad \text{(for a discrete membership function)} \tag{11}$$

where n represents the number of elements of the sampled membership function and μ_i the grade of ith element. The center of gravity of the final conclusion in Fig. 5 is obtained to be -0.2. Therefore, the dial should be assigned to be -0.2 in case of $\Delta T = +2.5$°C and $y = -1$°C/min.

The final conclusion is characterized by a relatively complicated membership function and the physical meaning may be represented by a fuzzy linguistic term "cool just a little," because a center of gravity is shifted a little bit to the left by the side lobe.

When the two inputs are changed independently, we can obtain the continuous (sometimes nonlinear) control surface in the 3-D space ($\Delta T, y$, and z) from small amount of fuzzy rules because of the interpolative reasoning.

Consequently we have to develop two kinds of hardware systems. The first one is a *fuzzy inference engine* which achieves an individual fuzzy inference and the second one is a *defuzzifier* which aggregates all the individual conclusions and derives a center of gravity of the membership function of the final conclusion.

V. Hardware Implementation of a Fuzzy Logic Controller in Nonlinear Analog Mode

A fuzzy logic controller, the function of which is characterized by a set of fuzzy IF-THEN rules, accepts and produces deterministic signals. It is constructed by fuzzy inference engines, the number of which is equal to that of rules employed, and a defuzzifier which converts a final conclusion of fuzzy value to a deterministic value.

Fuzzy inference and defuzzification can be accomplished by software or hardware. The software implementation with an ordinary personal computer or a work station takes a long time (for instance, several ms to several hundreds of a ms) to obtain the inference result. On the other hand, the specific hardware implementation accomplishes higher speed processing and exhibits better compactness and lower cost.

The hardware implementation is realized in either digital mode or analog mode. The digital hardware implementation of a fuzzy system originated from M. Togai and H. Watanabe [29] and comes to practical useful chips [30]–[33]. Digital fuzzy chips have distinctive features, such as good programmability, easy design, good compatibility with digital systems, etc. The analog hardware implementation originated from the author's work both in current mode [34] and voltage mode [35], and were followed by many researchers [36]–[42]. Analog fuzzy chips have distinctive features, such as high speed, good compatibility with sensors, etc. Attempts were often made to discuss merits and demerits of digital and analog fuzzy chips. It has not been concluded to choose between them. The adoption should be made according to the circumstances. The interest of this paper is focused on the analog fuzzy hardware system.

Electronic circuits employed in the fuzzy inference engine mainly handle analog signals in this paper. However, these circuits are different from the traditional analog circuits such as an operational amplifier, a multiplier/divider and their derivatives. The circuits employed in the inference engine are *nonlinear analog circuits.*

In this chapter, nonlinear analog circuits, a practical implementation of a fuzzy inference engine, and a defuzzifier are described.

A. Ordinary Analog Circuits and Fuzzy Circuits

There are two ways to implement the algorithm to the hardware system. One is a digital circuit and the other is an analog circuit. Analog circuitry was used to construct an analog computer or some other instrumentations. However, it had less extensibility, so that it is now replaced with digital circuitry.

When we consider the hardware of a fuzzy system, we have to recall the analog system. This is because fuzzy logic handles a continuous grade ranging from 0 to 1, not only 0 or 1. However, an analog circuit peculiar to a fuzzy system is expected to be different from the ordinary analog circuit in the following respects.

1. Since a grade of membership handled in the fuzzy system is 0 through 1 and a resolution of 10% (20 dB) is enough, the designer should not suffer from accuracy including linearity, low thermal drift, low offset, etc.
2. A fuzzy circuit essentially does not need the gain greater than unity, while a high gain and roll off of an amplifier causes unstability (oscillation) and diminishes the extensibility of the system.
3. A fuzzy logic system usually exhibits a narrow signal processing procedure, while an ordinary analog system achieves a deep signal processing, i.e., multistage processing with amplifiers and thus produces a significant cumulative error.
4. A fuzzy circuit should be designed in a simple architecture (with a few devices) to achieve high speed processing, while an ordinary analog circuit is constructed with many devices, some of which are dedicated to increase the linearity, compensate the thermal drift and phase shift, generate the reference voltages, obtain a stable gain, and so on.

The input-output characteristics of intrinsic fuzzy circuits are nonlinear, so that a fuzzy circuitry can be called a *nonlinear analog circuit,* while the traditional operational amplifiers and multipliers are called linear analog circuits.

B. Fuzzy Inference Engine (Rule Chip) [35], [43]–[47]

A fuzzy inference engine described here was designed for PID control, so that three input variables (error, change of error and rate of change of error) and one output were considered. Of course, these input signals are of deterministic values and output signal is a fuzzy value. The architecture of the fuzzy inference engine is shown in Fig. 7, where Rule 3 is programmed (If ΔT is PS* and y is NS, then the dial should be ZR) as an example.

The first stage of the fuzzy inference engine is a *membership function circuit* (MFC), three of which are dedicated to input signals X′, Y′, and Z′. The input-output characteristics of an MFC exhibits the shape of the membership function, which can be assigned externally. Each input terminal has the acceptable range, i.e., minimum and maximum values, which are represented by -1 and $+1$, respectively. In the practical case, -1 and +1 are represented by -5 V and $+5$ V and correspond to -5°C and $+5$°C and also to -3°C/min and $+3$°C/min, respectively. The output voltage of the MFC ranging from 0 V to 5 V corresponds to the grade of membership ranging from 0 to 1. The third membership function is not assigned, because the IF-THEN rules does not include the third variable in the antecedent. The label of the third input variable is referred to as NA (not assigned, don't care). An MFC assigned to be NA produces the grade 1 or the voltage 5 V not to affect other grades at the next MIN stage.

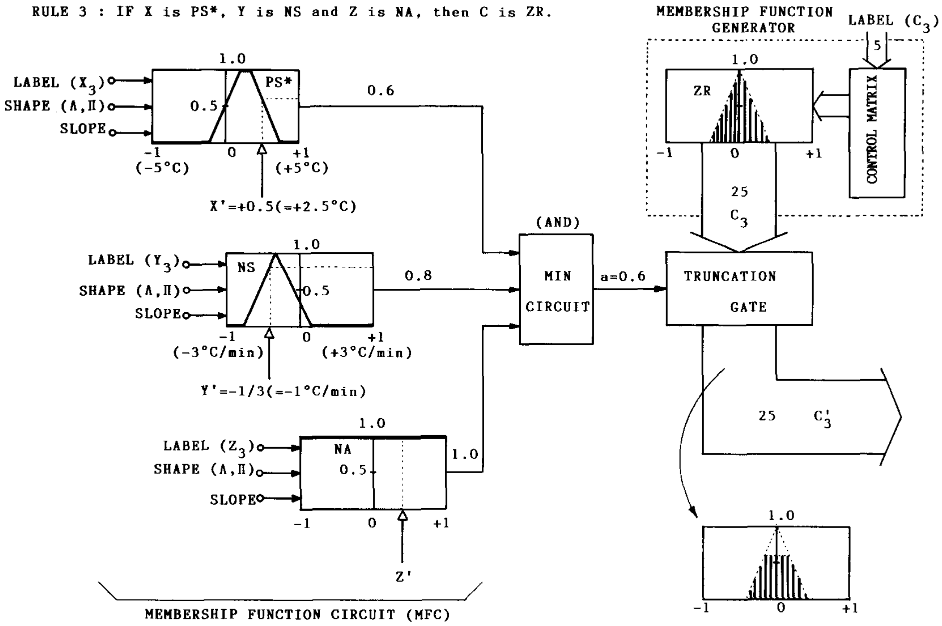


Fig. 7. An architecture of a fuzzy inference engine which accepts three deterministic input values and produces one fuzzy output value.

The second stage is a *MIN circuit* which accepts more than one input signals and produces the lowest values of them at the output terminal. This circuit achieves a conjunction "and" in the antecedent of IF-THEN rules. If variables in the antecedent are connected with "or," the MIN circuit should be replaced with a MAX (maximum) circuit. The output of these circuits represents the degree of soft matching between the antecedent and the inputs and thus feeds to the next stage to weigh the consequent membership function.

The consequent membership function is represented as voltage signal distribution on a data bus. The amplitude of the signal voltage ranges from 0–5 V corresponding to grades 0 to 1, respectively, and the number of signal lines in the data bus is 25 in this paper. We cannot call this data bus as 25 bits but as 25 elements, because each signal voltage is not binary but continuous (analog). This voltage distribution representing the consequent membership function is delivered from a *membership function generator* which has no input ports but a control port. In other words, the voltage distribution is assigned by a 5-bit binary word (LABEL).

A consequent membership function is truncated in a truncation gate by a degree of soft matching and the individual conclusion is obtained at the output data bus C_3'.

Each block is described in detail in the following.

1) MIN Circuit and MAX Circuit: A *MIN circuit* and a *MAX circuit* are the most basic fuzzy logic gates. Fig. 8(a) shows a MIN circuit employing bipolar transistors. PNP transistors and a current source I_{E1} construct a comparator and an NPN transistor and another current source I_{E2} construct a compensator. The potential of the emitter of the PNP transistor is equal to the minimum value among input voltages $X_1, X_2, X_3, \cdots, X_n$ plus 0.7 V (V_{EB} of an active input transistor), because higher input voltages makes corresponding input transistors cut-off. The following compensator (emitter follower) compensates the voltage shift of 0.7 V to produce the output voltage equal to the minimum input voltage. The compensator compensates the thermal drift as well as the voltage shift of emitter junction of the comparator. Fig. 8(b) shows an input-output characteristic and exhibits a nonlinear analog aspect. A transient response of this circuit is shown in Fig. 8(c), which illustrates that the rise and fall times are less than 10 ns. A Max circuit is shown in Fig. 9(a), where PNP transistors and an NPN transistor in a MIN circuit is replaced with NPN transistors and a PNP transistor, respectively. Comparison and compensation are achieved in the similar manner. Fig. 9(b) and (c) show an input-output characteristic and a transient response.

Since all transistors in comparators of a MIN circuit and a MAX circuit are coupled at emitters, the author named these circuits as *emitter coupled fuzzy logic* (ECFL) *gates,* the behavior of which is quite different from that of ordinary emitter coupled logic (ECL) in binary digital logic.

Two or more MIN circuits can be combined at the emitter of input transistors to extend the number of input terminals. In the similar manner, a MAX circuit can be extended.

The distinctive features of emitter coupled fuzzy logic gates are summarized in the following.

1. The output error is insensitive to the transistor size (e.g., base width, collector and emitter areas etc.), a variance in transistor characteristics (e.g., h_{FE}, I_{CBO}, etc.), while the missmatch and the variance give fatal errors in linear analog circuits.

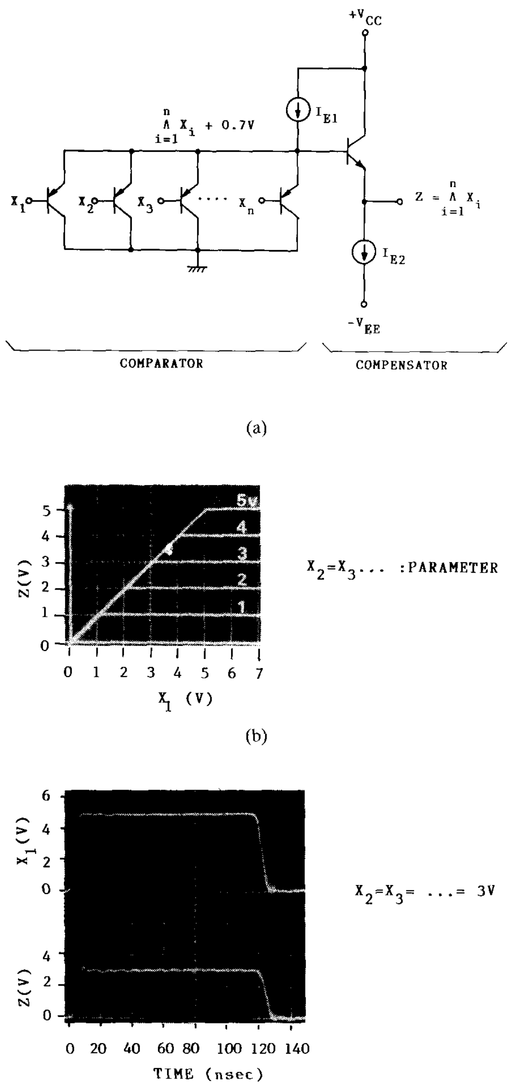


Fig. 8. An emitter coupled fuzzy logic (ECFL) gate (a MIN circuit). (a) Circuit configuration, (b) input-output characteristics, and (c) transient response.

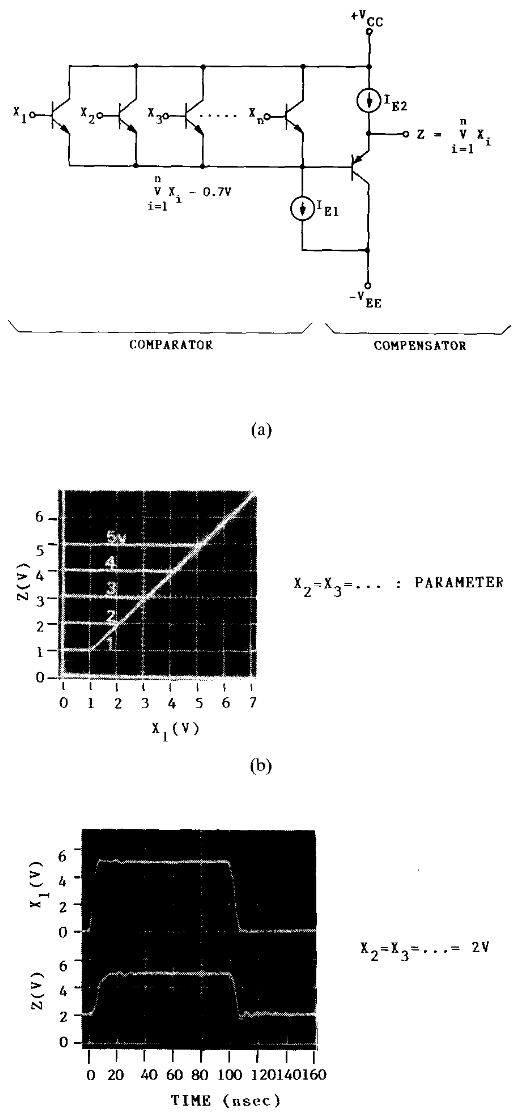


Fig. 9. An emitter coupled fuzzy logic (ECFL) gate (a MAX circuit). (a) Circuit configuration, (b) input-output characteristics and (c) transient response.

2. Good thermal stability is guaranteed. Experimental results give the output change of 0.2 %F.S. by the temperature change ranging from $-55°C$ to $+125°C$. The output fluctuation is not the problem.
3. The output error is insensitive to the operating currents. The author obtained the experimental results, for the device carefully designed, that the output change caused by the current change of one decade was less than 2 %F.S.
4. These ECFL gates are very robust against the fluctuation of supply voltages, because both a comparator and a compensator are driven by current sources. 5-V fluctuation of supply voltage caused the output fluctuation less than 0.1 %F.S.
5. High speed operation is obtained because no saturation of minority carriers in the base region occurs.
6. The ECFL gates are designed based on emitter followers, thus exhibit high input impedance and low output impedance. It guarantees a large fan-out.
7. The input terminals which are not utilized can be left opened (there is no need to terminate them), because the input transistors of open base are cut-off and does not affect other input transistors. This characteristic is very important in designing a semicustom IC.

2) MFC: Membership functions of variables in the antecedent are implemented by an MFC. In other words, a membership function circuit is one whose input-output charac-

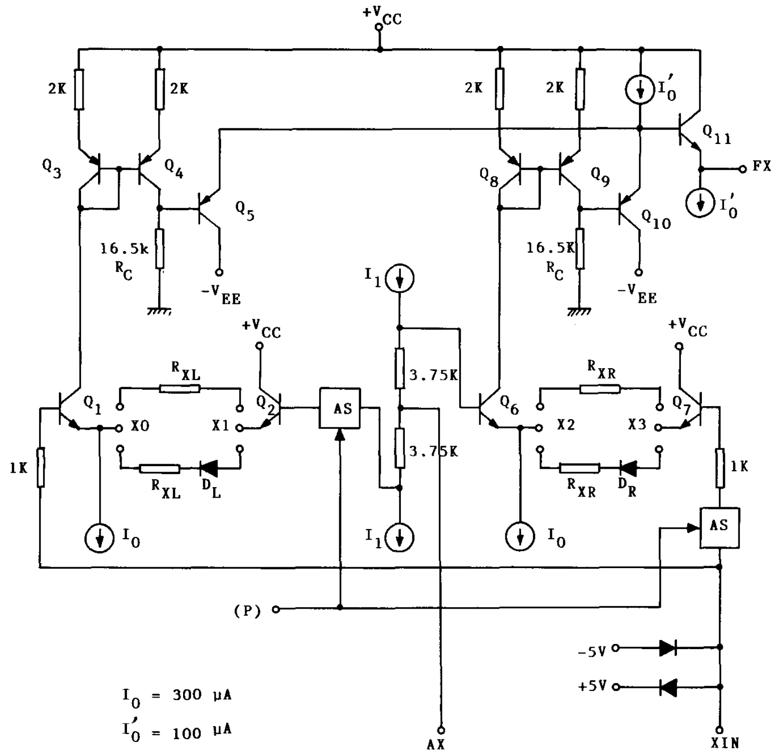


Fig. 10. Circuit configuration of a membership function circuit (MFC).

teristics can be assigned by external signals or devices. Fig. 10 shows the circuit configuration of an MFC which can realize five functions of a Z-function, an S-function, a Π-function (trapezoidal function), a Λ-function (triangular function), and NA (not assigned, don't care). XIN and FX are an input and an output terminals, respectively. AS is an analog switch controlled by an external signal (P).

If (P) is disable, two analog switches turn off and thus two current sources I_0 drive Q_1, Q_3, Q_4 and Q_6, Q_8, Q_9, and base voltages of Q_5 and Q_{10} reach $I_0R_C \approx 5$ V. Q_5, Q_{10}, Q_{11} and two current sources I_0' construct a MIN circuit, so that the output FX is 5 V independent of XIN. This means NA.

If (P) is enabled and X0-X1 are terminated by R_{XL} with X2-X3 opened, the input-output characteristics of this MFC exhibits S-shapes as shown in Fig. 11(a). If (P) is enabled and X2-X3 are terminated by R_{XR} with X0-X1 opened, the input-output characteristics exhibit Z-shapes as shown in Fig. 11(b). If (P) is enabled, and X0-X1 and X2-X3 are terminated by R_{XL} and R_{XR}, respectively, the MFC becomes a Λ-function as shown in Fig. 11(c). If (P) is enabled, and X0-X1 and X2-X3 are terminated by $R_{XL} + D_L$ and $R_{XR} + D_R$, respectively, the MFC becomes a Π-function as shown in Fig. 11(d). The shoulder voltages of S- and Z-functions and the center voltages of Λ- and Π-functions can be assigned by an external voltage AX. Thus labels of the membership functions can be assigned by AX. The positive slope and the negative slope of each function are reciprocally proportional to R_{XL} and R_{XR}, respectively. Therefore, fuzziness of the membership function can be assigned by R_{XL} and/or R_{XR}. Fig. 12(a) and (b) show the input-output characteristics of a triangular membership function for various labels assigned by AX and for various resistors R_{XR} (upper) and R_{XL} (lower), respectively.

3) Membership Function Generator: A membership function of a consequent is sampled to discrete grades, which are represented by voltages ranging 0–5 V and distributed on 25 signal lines (25-element data bus) as shown in Fig. 6(b) or Fig. 7. The shape and the label of the membership function can be changed by reassigning the voltage distribution as shown in Fig. 13. Seven labels (NL, NM, NS, ZR, PS, PM, and PL) and NA (not assigned: all voltages on the data bus are 0 V) are discriminated by a three-bit binary word and Λ-shape, S-shape, Z-shape, and Π-shape are done by a two-bit one, so that these membership functions in the consequent are assigned by a 5-bit binary word (control word). The slope of a membership function shown in Fig. 13 with 25 signal lines (25 elements) guarantees that only four dc voltage sources (1.25 V, 2.5 V, 3.75 V, 5 V) are enough to realize these various types of membership functions.

The desired voltage distribution is obtained by switching each signal line to a desired supply voltage. Fig. 14(a) shows a switch matrix and two emitter follower arrays, and Fig. 14(b)

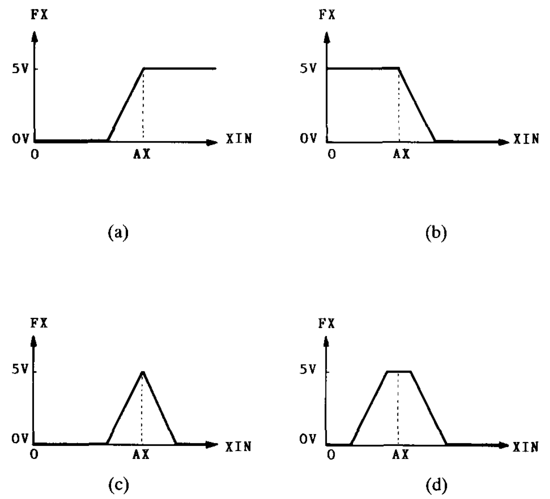


Fig. 11. Four types of membership functions realized by an MFC shown in Fig. 10. (a) S-function, (b) Z-function, (c) Λ-function (triangular function), and (d) Π- function (trapezoidal function).

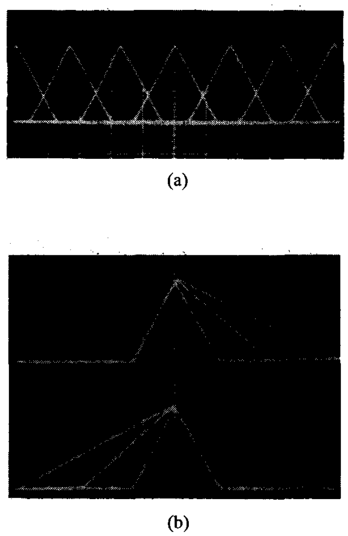


Fig. 12. Input-output characteristics of a membership function circuit (MFC). (a) Λ-functions for various labels (AX = -5 V, -3.3 V, -1.7 V, 0 V, 1.7 V, 3.3 V, 5 V). (b) Λ-functions for various slopes. (Upper: $R_{XR} = 120\,\Omega, 266\,\Omega, 444\,\Omega, R_{XL} = 120\,\Omega$. Lower: $R_{XL} = 120\,\Omega, 266\,\Omega, 444\,\Omega, R_{XR} = 120\,\Omega$).

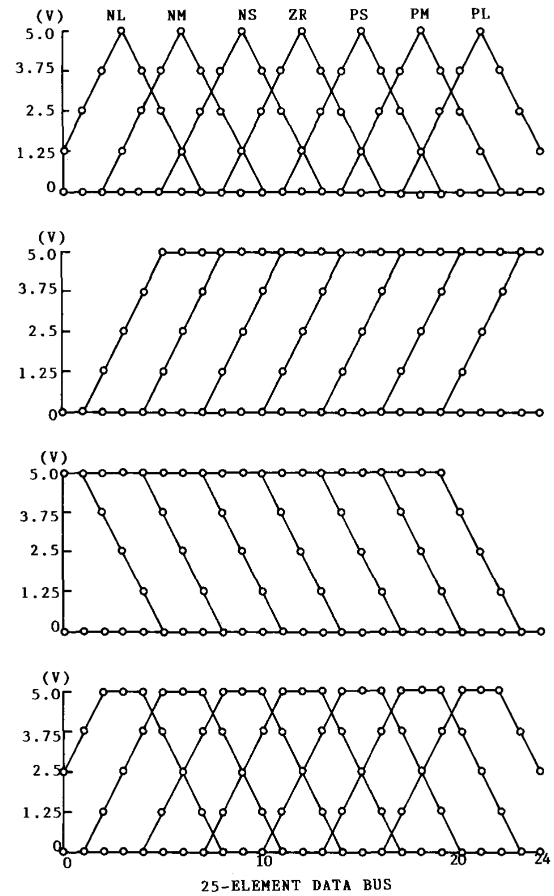


Fig. 13. Voltage distributions on 25-element data bus which represent consequent membership functions such as NL (negatively large), NM (negatively medium), NS (negatively small), ZR (approximately zero), PS (positively small), PM (positively medium), PL (positively large), and NA (all voltages on 25-element data bus are 0 V). NA in a consequent is used for eliminate the rule. Four shapes (triangular, S-shaped, Z-shaped and trapezoidal) are discriminated. Four voltage sources (1.25 V, 2.5 V, 3.75 V, and 5 V) are enough to realize a consequent membership function on 25-element data bus.

shows a control matrix which delivers control signals to analog switches in a switch matrix. In each column of the switch matrix in Fig. 14(a), only one analog switch is turned on by a control signal $G_{i,j}$. In other words, one bit in each binary word C_i is enabled by a 5-bit control word (CS1, CS0, CL2, CL1, CL0) through the 5-input 125-output control matrix. The control matrix can be designed in combinational logic or PLA. Emitter follower arrays are used for impedance transformation and level/temperature compensation. How significant is the number of the element in the consequent membership function? How important is the shape of consequent membership function? They will be discussed in detail in Section V-D-1.

4) Truncation Gate: A truncation gate accepts one fuzzy vector signal (fuzzy word signal) $\boldsymbol{B}$ and one fuzzy scalar signal (ranging from 0 V to 5 V) "a." The former one represents a membership function of consequent and is delivered from a membership function generator. The latter one represents the degree of soft matching between the fact and the antecedent, and is delivered from a membership function circuit. The truncation gate produces a fuzzy vector signal $\boldsymbol{B}'$ truncated by the fuzzy scalar input signal as shown in Fig. 15(a). In other words, the truncation gate accepts one 25-element input data and one analog input and produces one 25-element output data, the voltage distribution of which is truncated by the analog input. This can be constructed by arranging MIN circuits in

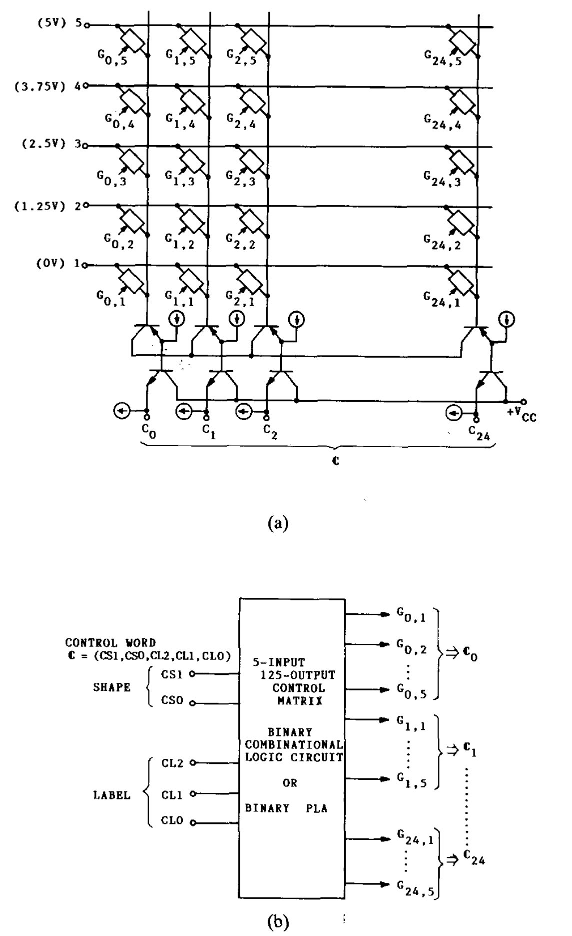


Fig. 14. Main blocks of a membership function generator. (a) A switch matrix and an emitter follower array (b) a control matrix.

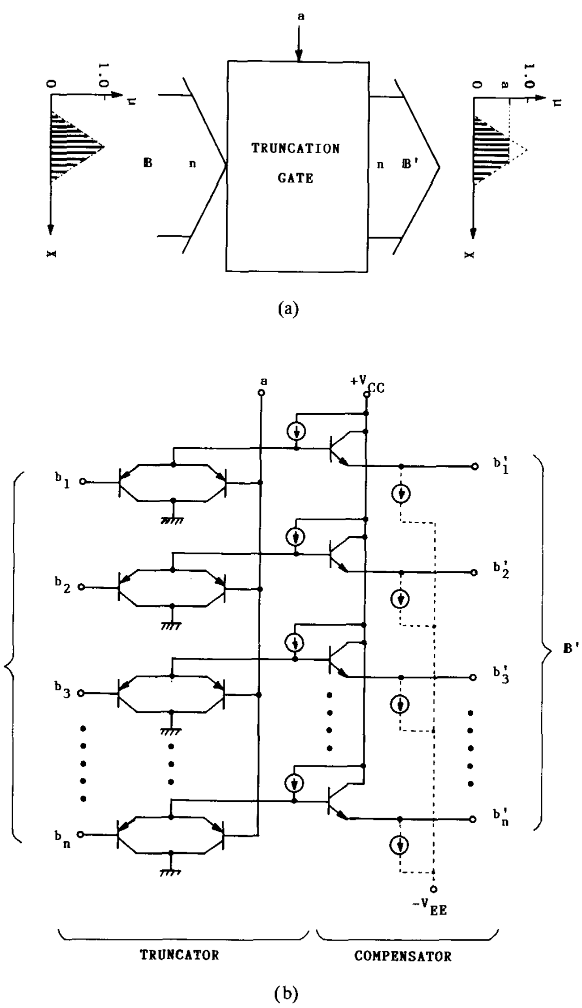


Fig. 15. A truncation gate. (a) Its function and (b) its circuit configuration.

an array. The circuit configuration is shown in Fig. 15(b), where one input terminal of each MIN circuit is connected to a common terminal "a." The other input terminal of each MIN circuit is dedicated to the element signal of an input fuzzy word $\boldsymbol{B}$. The output terminal of each MIN circuit produces the element signal of an output fuzzy word $\boldsymbol{B}'$.

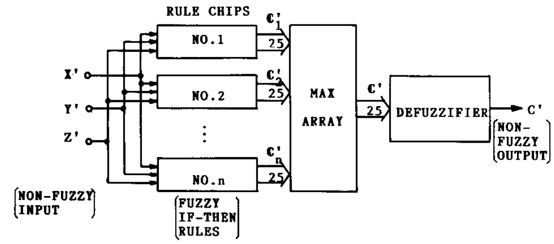
5) Open Emitter for Saving a MAX Array: The output ports of fuzzy inference engines should be usually fed to a MAX array to aggregate all the individual conclusions before defuzzification as shown in Fig. 16(a). If all current sources (indicated by a dotted line in Fig. 15(b)) in a comparator of a truncation gate are omitted, all input terminals of a defuzzifier are driven by the flow-in current sources inside the defuzzifier and each input terminal of the defuzzifier is connected to the corresponding terminals of all the fuzzy inference engine outputs as shown in Fig. 16(b), the MAX array can be omitted from this construction. In other words, a fuzzy inference engine with open emitter structure and a defuzzifier involving the current source in the input terminal facilitates *wired MAX* by themselves. Thus the open emitter structure saves a MAX array as well as current sources in compensators.

6) Silicon Implementation: The architecture of the fuzzy inference engines shown in Fig. 17, which is implemented in the monolithic form and called a Rule Chip because one fuzzy inference based on one fuzzy IF-THEN rule (or shortly, fuzzy rule, or control rule), is achieved in this chip.

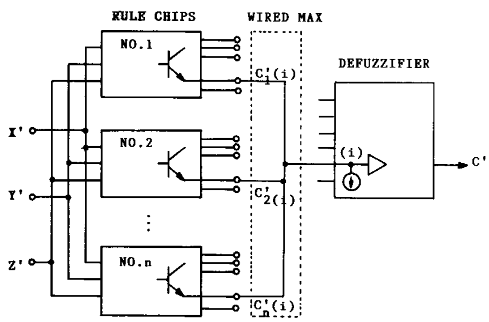
A LABEL X input for a variable x is a 4-bit word ($\overline{\text{ANAX}}$, AX2, AX1, AX0) to assign the label of the membership function of x. When $\overline{\text{ANAX}}$ is 0, a multiplexor selects the control signal, ANALOG LABEL, to a membership function circuit. Otherwise, the following 3-bit word (AX2, AX1, AX0) assigns one of eight labels as shown in Table I.

The upper and lower limitations of the input signal is assigned by the voltage REF1 and REF2, which can be supplied from a power supply (+10 V) through variable resistors, and detected at two corresponding terminals MONITOR.

REF3 is the terminal where +5 V is usually supplied or a lower voltage is supplied through an external resistor in case of lightening the rule (weight less than 1 or less important than other rules). The weight can be detected at MONITOR corresponding to REF3.



(a)



(b)

Fig. 16. (a) Aggregation of individual conclusions (individual inference results) with a MAX array. (b) Aggregation by wired MAX.

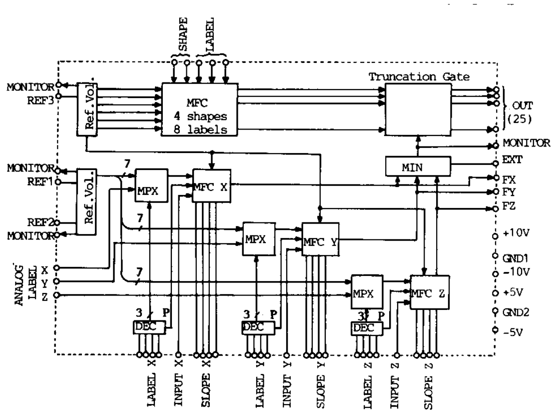


Fig. 17. Architecture of the fuzzy inference engine (rule chip).

A membership function in a consequent can be selected by 5-bit control word as shown in Tables II and III.

When a binary terminal is floating, it is pulled up to a high level equivalent to 1 just like a TTL.

FX, FY, and FZ are the terminals for monitoring the degree of soft matching. The shape of each membership function can be watched by applying a sinusoidal wave to each signal input.

An EXT terminal can be used for combining several rule chips in order to realize more than three variables in the antecedent as described in Section V-B-1. For instance, if three rule chips are connected to each other at terminals EXT, we

TABLE I

	$\overline{\text{ANAX}}$	AX2	AX1	AX0
Analog	0	DON'T CARE		
NL	1	0	0	0
NM	1	0	0	1
NS	1	0	1	0
ZR	1	0	1	1
PS	1	1	0	0
PM	1	1	0	1
PL	1	1	1	0
NA	1	1	1	1

TABLE II

Shape	CS1	CS0
Λ	0	0
S	0	1
Z	1	0
Π	1	1

TABLE III

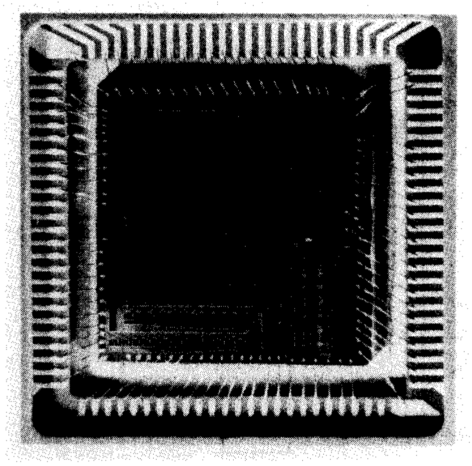
Label	CL2	CL1	CL0
NL	0	0	0
NM	0	0	1
NS	0	1	0
ZR	0	1	1
PS	1	0	0
PM	1	0	1
PL	1	1	0
NA	1	1	1

can implement a rule which has variables in the antecedent up to nine.

The circuit of the fuzzy inference engine is integrated in the monolithic form by a standard BiCMOS process. The design rule of 5 μm, two layers of aluminum and one layer of poli-silicon is adopted. About 600 transistors and about 800 resistors are embedded in a 8 mm $\times$ 8 mm silicon chip. Fig. 18(a) shows a microphotograph of the fuzzy inference engine (Rule Chip). This chip is molded in an 84-pin plastic package. The response time of fuzzy inference is 1 μs as shown in Fig. 18(b). This is equivalent to 1 000 000 fuzzy inferences per second (FIPS).

C. Defuzzifier (Defuzzifier Chip) [48]

In order to obtain a deterministic value from a membership function of the final conclusion of fuzzy inference, the defuzzification should be achieved. Fig. 19 shows the defuzzifier circuit. All the individual conclusions are aggregated by wired MAX technology as described in Section V-B-5, that is, the defuzzifier possesses the current-source-input buffer which facilitates a wired MAX with rule chips connected to the input. The output of the buffer is fed to a weighted sum circuit and an ordinary sum circuit, the outputs of which are fed to an analog divider to calculate (15). The weights i in (15) are reciprocally proportional to resistances of a resistor array and thus positive.



(a)

(b)

Fig. 18. (a) A microphotograph of the fuzzy inference engine (rule chip) and (b) its transient response.

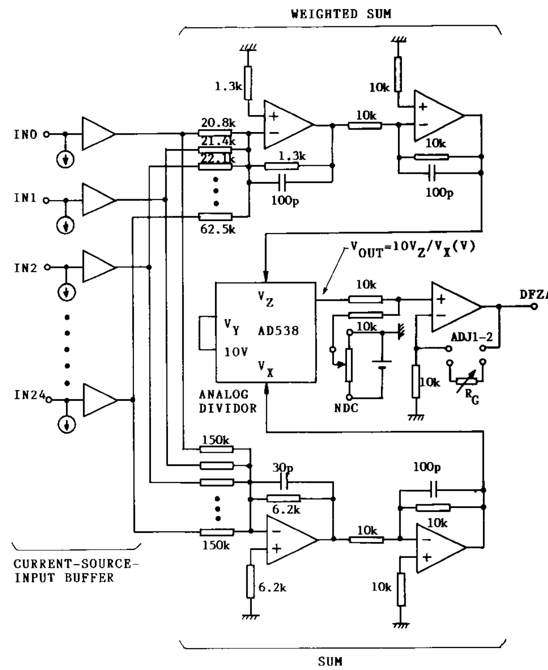


Fig. 19. A circuit configuration of the defuzzifier.

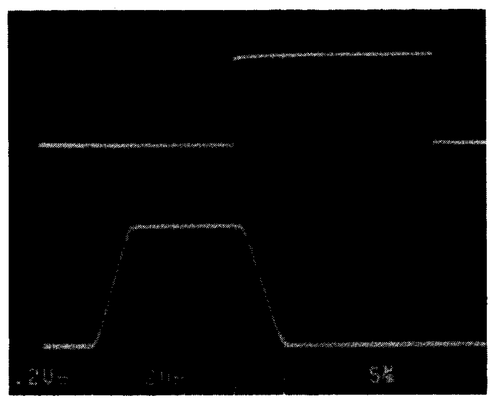


Fig. 20. A step response of the defuzzifier.

Therefore, the calculated value is always positive. In order to obtain the bipolar output signal ranging from -5 V to 5 V, an offset adjustment and an amplification adjustment are achieved by NDC and R_G, respectively. The defuzzifier is implemented on a ceramic base in the hybrid form, and molded in a 44-pin plastic package. The response time of defuzzification is about 5 μs as shown in Fig. 20, which is almost determined by that of an analog divider used in this defuzzifier.

D. Advanced Implementation of a Fuzzy Logic Controller [49],[50]

A direct hardware implementation of the fuzzy inference algorithm shown in Fig. 5, both of inference engines and a defuzzifier, needs too many devices, dissipates high power, and thus the costs are high. So that it is necessary to consider to design another one for each of fuzzy inference and defuzzification.

1) Fuzzy Inference with a Singleton Consequent: The algorithm of fuzzy inference shown in Fig. 5 is based on the compositional rule of fuzzy inference [27], which is the extension of a classical logic inference to a fuzzy logic and allows us to understand the aspects and the meaning of the fuzzy inference easily. Once the concept is understood, we have to consider other ways equivalent to it, which may be suitable for hardware implementation.

One of the main problems in the fuzzy inference engine is a parallelism in truncation of the consequent. As the number of elements of the consequent membership function increases, the number of devices for switching, buffering, minimizing, and

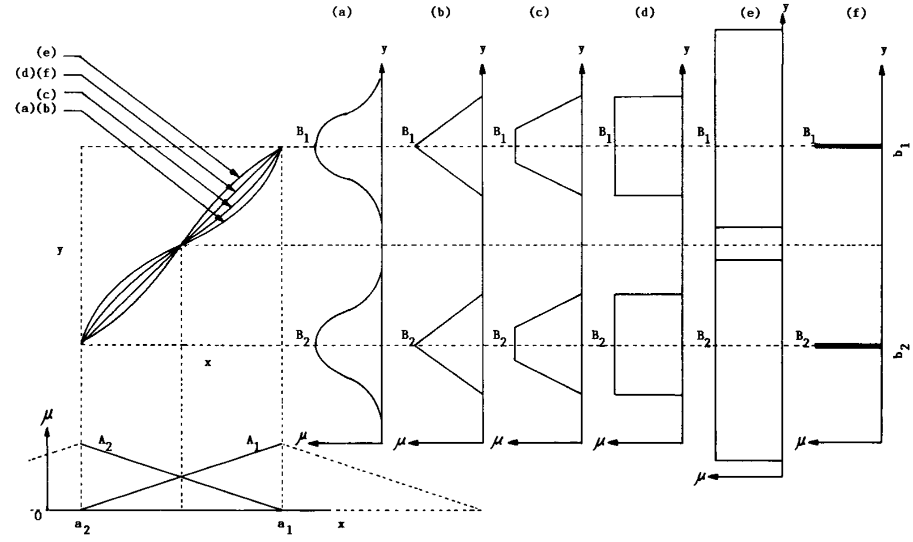


Fig. 21. Comparison of interpolation between some fuzzy inferences.

binary combinational logic circuits increases proportionally. How many elements are needed in the consequent membership function for reasonable inference? Which shape of the membership function in the consequent is the best, though the shape is not essential for the fuzzy inference?

The significant ability of fuzzy inference is the reasonable interpolation between a small number of fuzzy data. Therefore, the attempt is made to compare the abilities of interpolation. Let us consider the case of interpolation between two fuzzy rules as

Rule 1 If x is A_1, then y is B_1.

Rule 2 If x is A_2, then y is B_2.

A_1 and A_2 are assumed to be the fuzzy linguistic terms defined by triangular membership functions as shown in Fig. 21, though the membership functions are meaningless outside the range a_1–a_2. B_1 and B_2 are assigned to be of (a) bell-shape, (b) triangular shape, (c) trapezoidal shape, (d) square shape (crisp interval), (e) overlapped square, and (f) singleton (deterministic value). An inference results of interpolation between these two data is shown in a square formed by dotted lines corresponding to typical values a_1, a_2, b_1, and b_2 in Fig. 21, which is exaggerated to make it easy to understand the differences.

This figure shows that the unoverlapped crisp intervals and the singletons in the consequent facilitate the *linear interpolation,* while all the other cases produce the nonlinear interpolation, the deviation of which depends upon the shapes and overlapping rate. This aspect of singletons is entirely favorable to hardware design and programming. Because we do not need to design the massively parallel processing architecture in a truncation gate and a membership function generator, we do not need to search for the best shape of the membership function in the consequent by computer simulation.

2) Division Reduction by Feedback: In a practical use of fuzzy inference, e.g., a fuzzy logic controller, an output signal of deterministic value, but not fuzzy value, is desired. Thus a defuzzifier is usually attached to the inference engines to construct a fuzzy logic controller.

A significant portion, which determines the operating speed and causes the difficulty of design and the troublesome adjustment, is an analog divider in a defuzzifier. In order to eliminate this analog divider from a defuzzifier, the new architecture should be developed. A center of gravity is calculated by (10) or (11), where a weighted sum of membership grades is divided by a normal sum of them. If the normal sum (denominator) is kept equal to the unity or equal to the constant at any time, no divider is needed. The normal sum is calculated and should be fed back to all the inference engines to reproduce the individual conclusion, the total of which is equal to unity or the constant. It can be realized by a membership function circuit, whose maximum value (or full grade) of membership can be changed by an external signal. The author named this type of membership function circuit as a *grade-controllable membership function circuit (GC-MFC).*

The grade-controllable membership function circuit can be obtained by modifying the ordinary membership function circuit (Fig. 10) as shown in Fig. 22, where two current sources I_0 can be changed by the output current of the V-I converter. The input-output relation of this V-I converter is obtained as

$$I_0 = \frac{1}{R_i} W \tag{12}$$

where W and R_i are a weight input to assign the peak value of the membership function and a resistance shown in Fig. 22, respectively, and the peak value of MFC output is

$$FX_{\max} = R_C \cdot I_0 \tag{13}$$

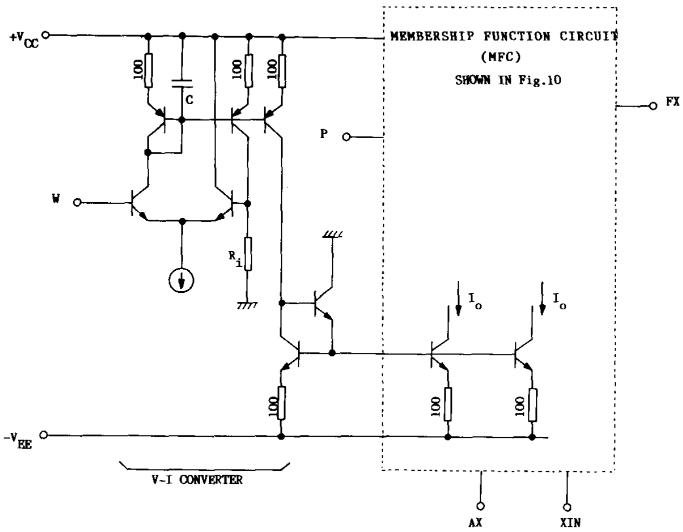


Fig. 22. A grade-controllable membership function circuit (GC-MFC). It is obtained by modifying the ordinary membership function circuit.

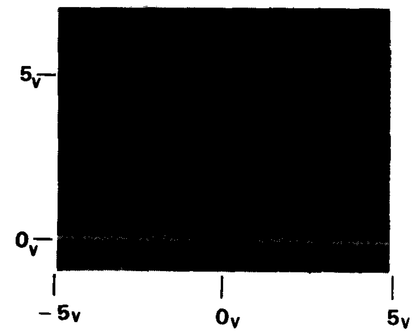


Fig. 23. Input-output characteristics of a grade-controllable membership function circuit. Attached numbers are the control voltages W in volts.

where R_C is an I-V conversion resistance of the grade-controllable membership function circuit (Fig. 22) and is practically shown in Fig. 10.

Thus the peak value of GC-MFC output is obtained as

$$FX_{\max} = \frac{R_C}{R_i} W. \tag{14}$$

Fig. 23 shows the input-output characteristics of this GC-MFC, where $R_C = R_i$ and attached numbers are the control voltages W in volts. This figure illustrates that the input-output characteristics are not multiplied by the control signal, but shifted down and up by it.

A fuzzy logic controller, which achieves fuzzy inferences with fuzzy linguistic terms in IF-clause and singletons in THEN-clause and defuzzifies the final conclusion, is referred to as a *singleton controller.* The example of the rule is "If TEMPERATURE OF CHAMBER grows up *higher* and PRESSURE becomes *a little bit higher,* then FUEL should be *reduced to 3 liters/minute.*"

The GC-MFC's are employed to construct the singleton controller which accomplishes defuzzification without an analog divider as shown in Fig. 24. The ordinary sum of soft matching degrees $a_1 + a_2 + a_3 + \cdots + a_n$ are compared with the constant E_1. When the sum exceeds E_1, all the control inputs W are pulled down by the operational amplifier to make the output of each GC-MFC decrease and thus the ordinary sum decrease. When the ordinary sum becomes below E_1, all the control inputs W are pulled up to increase the ordinary sum. Consequently, the sum $a_1 + a_2 + a_3 + \cdots + a_n$ is equal to the constant E_1 at any time. All of these soft matching degrees $a_1, a_2, a_3, \cdots, a_n$ are weighted by resistors R_j in consequent blocks, respectively, to be summed to produce the weighted sum equal to the center of gravity. The offset E_0 is subtracted from the weighted sum to shift the output level. Precise description on E_0 is presented in the following. Three GC-MFC's in each antecedent block and a resistor R_j in each consequent block are programmed in accordance with the fuzzy rules to be implemented.

The estimation of the resistance R_j depends upon whether the consequent space is unipolar (there exist only the positive values) or bipolar (positive and negative values can be defined). When the output of the singleton controller is fed to an electric heater, the consequent should be unipolar. When it is fed to a dc servo motor, the consequent should be bipolar. A reference diagram shown in Fig. 25 is helpful to estimate the resistance R_j. $R_{\max}$, and $R_{\min}$ should be also estimated so that the resistance ration $R_{\max}/R_{\min}$ should be implemented in the integrated circuit technology with reasonable accuracy. For instance, if E_0 is assigned to be 0 in case of unipolar consequent space in Fig. 25(a), $R_{\max}$ is infinity (open circuit) and R_j should be assigned ranging from $R_{\min}$ to $\infty\,\Omega$ which is impossible to realize. When the consequent space is bipolar and the weighting is achieved by a resistor array, negative weighting cannot be implemented.

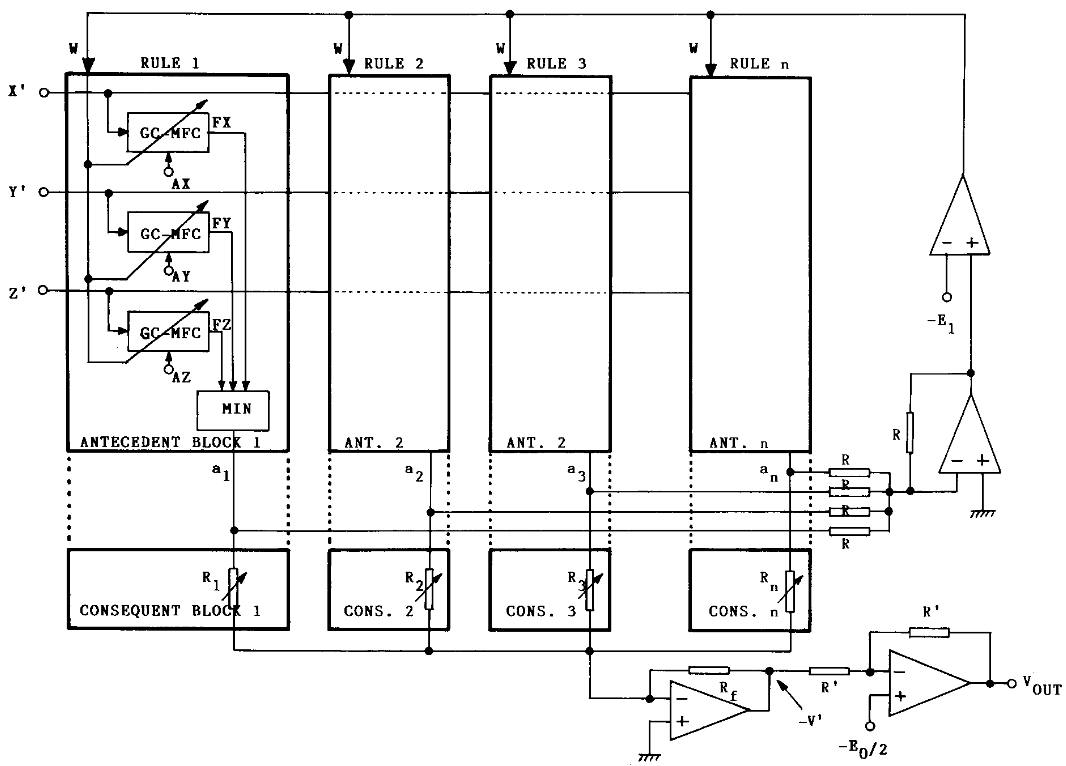


Fig. 24. A singleton controller which achieves fuzzy inferences with singletons in consequents and defuzzify the final conclusion. The defuzzification is accomplished without an analog divider but by employing the grade-controllable membership function circuits and the feedback loops including them. Three GC-MFC's in each antecedent block and a resistor R_j in each consequent block are programmable in accordance with the given fuzzy rules.

Therefore, the voltage space is shifted to the negative in order to calculate the center of gravity in the positive region and shifted back to the positive. The aspect is illustrated in Fig. 25(b), where E_0 is the offset voltage.

As a result, we can get the following equations for the unipolar space of consequent from Fig. 25(a), providing that $E_S = 10$ V, $E_0 = 2$ V, $R_f = 10\,\mathrm{k\Omega}$:

$$R_j = \frac{25}{1 + \dfrac{5C_j}{C_{\max}}}\ (\mathrm{k\Omega}) \tag{15}$$

and $R_{\max} = 25\,\mathrm{k\Omega}$ and $R_{\min} = 4.17\,\mathrm{k\Omega}$. Therefore, the resistance ratio is

$$\frac{R_{\max}}{R_{\min}} = 6 \tag{16}$$

and the value is reasonable to realize.

On the other hand, we can get the following equation for the bipolar space of consequent from Fig. 25(b), providing that $E_S = 10\,\mathrm{V}, E_0 = 7\,\mathrm{V}, R_f = 10\,\mathrm{k\Omega}$:

$$R_j = \frac{50}{5\dfrac{C_j}{C_m} + 7}\ (\mathrm{k\Omega}) \tag{17}$$

and $R_{\max} = 25\,\mathrm{k\Omega}$ and $R_{\min} = 4.17\,\mathrm{k\Omega}$.

A very simple example of three fuzzy rules is presented as

$$\begin{array}{ll} \text{Rule 1} & \text{If } x \text{ is } NL, \text{ then } c \text{ is } +5\,\mathrm{V} \\ \text{Rule 2} & \text{If } x \text{ is } PS, \text{ then } c \text{ is } \ 0\,\mathrm{V} \\ \text{Rule 3} & \text{If } x \text{ is } PL, \text{ then } c \text{ is } -5\,\mathrm{V}. \end{array} \tag{18}$$

The membership functions in antecedents are assigned as shown in Fig. 26. Assuming that C_m is 5 V, three resistors R_1, R_2, and R_3 in consequent blocks for these three rules are calculated by (17) to be $R_1 = 4.17\,\mathrm{k\Omega}, R_2 = 7.14\,\mathrm{k\Omega}$, and $R_3 = 25\,\mathrm{k\Omega}$, respectively. Fig. 27 shows the input-output characteristic of this singleton controller which verifies good linear interpolation of a singleton controller.

VI. Stabilization of a Glass with Wine

The best way to evaluate a tool is to use it for a typical problem. An inverted pendulum is the most popular example to evaluate the control technology [51]. It is a model of stabilizing a pole on a palm. The author presents a more interesting example to evaluate the control technology. It is stabilization of a glass with wine on a finger as shown in Fig. 28(a). This is very difficult, if not impossible, to control the finger to stabilize the glass, because the glass includes wine which swings back and forth by the external disturbance to change the center of gravity of the glass. How it can be stabilized?

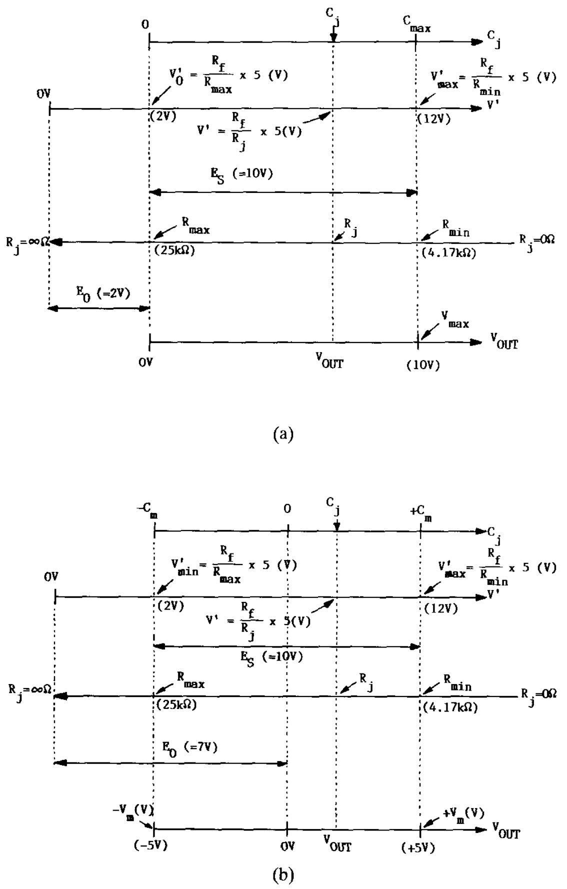


Fig. 25. A reference diagram of the consequent space which gives the resistance R_j of the jth consequent block from the deterministic value Cj in the consequent of the jth rule. The way of estimating the resistance R_j depends upon whether the consequent is (a) a unipolar space or (b) a bipolar space.

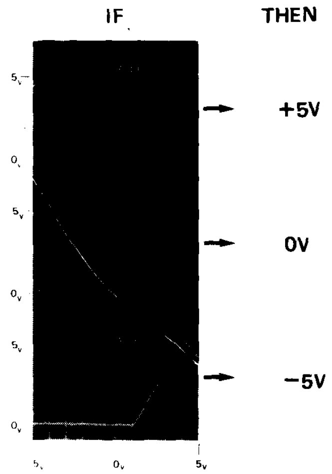


Fig. 26. Membership functions of NL (top), PS (middle), and PL (bottom) which are used in the antecedents of three rules for testing a singleton controller.

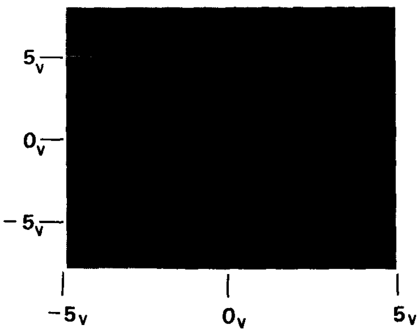


Fig. 27. Input-output characteristic of the singleton controller, the fuzzy rules of which are; Rule 1: If x is NL, then c is +5 V, Rule 2: If x is PS, then c is 0 V, Rule 3: If x is PL, then c is −5 V. This characteristic verifies good linear interpolation of a singleton controller.

A. Mathematical Approach

An approach to solve this problem, which is very familiar to us, is a mathematical one. At first, we make a simplified model to obtain the mathematical equations which describe the dynamics of this problem. Fig. 28(b) is one example of modeling. The vehicle corresponds to the finger and is driven by the controller not to make the sphere on the pole fall down. The sphere corresponds to the cup, the pole to the support, and the pivot to the tip of the finger. x is a distance from a set point (the place where it is desired to be stabilized). The mathematical equations describing this dynamics are obtained as in (19)–(21), shown at the bottom of the page.

$$A = \tfrac{1}{2}\rho\pi(-\tfrac{1}{5}h^5 + rh^4 - \tfrac{5}{3}r^2h^3 + r^3h^2) \tag{22}$$

$$B = \tfrac{1}{2}\rho\pi(\tfrac{1}{2}h^4 - 2rh^3 + 2r^2h^2) \tag{23}$$

$$C = \tfrac{1}{2}\rho\pi(rh^2 - \tfrac{1}{3}h^3) \tag{24}$$

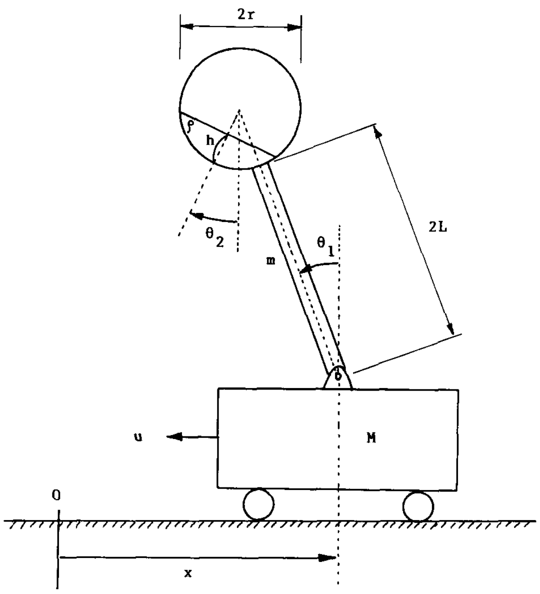
where M and m are the masses of the vehicle and the pole, respectively, $2L$ the length of the pole, r the radius of the sphere, h the depth of wine, ρ the specific gravity, θ_1 and

$$B\ddot{x}\cos\theta_2 + B(r+2L)\{(\ddot{\theta}_1 - \ddot{\theta}_2)\cos(\theta_1-\theta_2) - (\dot{\theta}_1^2 + \theta_2^2)\sin(\theta_1-\theta_2)\} = 0 \tag{19}$$

$$\begin{aligned}(C+Lm)\ddot{x}\cos\theta_1 &+ \{\tfrac{4}{3}mL^2 - 2C(r+2L)\}\ddot{\theta}_1 - Lm + 2C(r+2L)\,g\sin\theta_1\\ &- B(r+2L)\{\ddot{\theta}_2\cos(\theta_1-\theta_2) + \ddot{\theta}_2^2\sin(\theta_1-\theta_2)\} + D\dot{\theta}_1 = 0\end{aligned} \tag{20}$$

$$(2A+M+m)\ddot{x} + (C+Lm)(\ddot{\theta}_1\cos\theta_1 - \dot{\theta}_1^2\sin\theta_1) - B(\ddot{\theta}_2\cos\theta_2 - \dot{\theta}_2^2\sin\theta_2) = u \tag{21}$$

(a)



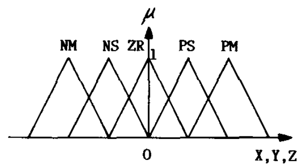
(b)

Fig. 28. (a) Stabilization of a glass with wine on a finger tip. (b) A simplified model of the glass stabilization.

$X = A\theta_1 + B\dot{\theta}_1$

$Y = Cx + D\dot{x}$	NM	NS	ZR	PS	PM
PM	NS		PS		(PL)
PS		NS		PM	
ZR	NM		ZR		PM
NS		NM		PS	
NM	(NL)		NS		PS

(a)



(b)

Fig. 29. (a) A rule map including fuzzy rules for stabilizing a glass with wine. (b) Membership functions which characterize the fuzzy linguistic terms used in the rule map.

θ_2 the angle of the pole, and the wine, respectively. When $\theta_1 = \theta_2$, this system reduces to a problem of the ordinary inverted pendulum.

Although this model is a simplified one, it is very difficult to derive (19)–(24). It needs high level knowledges about mathematics and physics. Even though these equations are derived, the analytical results on the criteria for stability, the dynamic response, the accuracy of position, and so on may not be necessarily reliable because the model shown in Fig. 28(b) is not the exact model. Therefore it is easy to come to a deadlock at this stage.

B. Fuzzy Linguistic Approach

The attempt is made to solve this problem without advanced mathematics but with knowledge like techniques possessed by artists, musicians, sport players, etc. Their knowledge may be represented by the relation between causes and results, or actions and reactions in the form of IF-THEN rules including fuzzy linguistic terms. Let us consider how the glass with wine is stabilized by the fuzzy linguistic approach.

1) Design of a Fuzzy Logic Controller: When we design a fuzzy logic controller, we have to think about the following items.

1. *What are the actuators (motor, heater, speaker, etc.)?* What are the variables in the consequent of fuzzy rules?
2. *What should be measured for controlling the system?* What are the variables in the antecedent of fuzzy rules?
3. *How wide are the ranges of sensors and actuators?* What are the minimum and maximum values of the above variables? All the labels should be assigned within this range.
4. *How many labels are needed to describe the strategy of control?* How precisely the system should be described? If the system should be described precisely, the support of each membership function should be narrow.
5. *What shape of membership functions are suitable for describing the expert's intuition?* Should it be bell-shaped, S-shaped, Z-shaped, triangular, trapezoidal, or of some other shape? The shape should not be necessarily symmetrical.
6. *What is the strategy (a set of fuzzy rules)?* What are the useful fuzzy rules? Which conjunction should be used to combine variables in the antecedent and/or the consequent? Of course, there is no problem, even though the strategy includes contradictory rules or exceptional rules. There is no need to adopt all the combinationally possible rules.

The velocity of the vehicle was adopted as a variable in the consequent of fuzzy rules. Of course the velocity is proportionally related to that of the driving motor.

It is obvious that the pole angle θ_1 and x can be the variable in the antecedent. We have to assign the different rules for positive $\dot{\theta}_1$ and negative $\dot{\theta}_1$ even in case of the same θ_1. So that $\dot{\theta}_1$ is also the variable to be considered in the antecedent. In the similar manner, $\dot{x}$ is also the variable in the antecedent.

The variable gain in the path of each signal flow is very useful to adjust the sensitivity of the fuzzy linguistic term (bandwidth, fuzziness, and label of the word). By extension or compression of the membership function with respect to the universe of discourse (physical quantity such as length, weight, speed, amount of fuel, brightness, etc.), the meaning of the term can be changed.

2) Reduction of Fuzzy IF-THEN Rules: When four variables $\theta_1, \dot{\theta}_1, x, \dot{x}$ are used in the antecedent and 5 labels are adopted for each variable, $5^4 = 625$ rules should be examined. It is not so easy for a designer to do it. To cope with this problem, the number of variables should be reduced. The rule chip does not accept more than three inputs, so that four variables should be reduced to two or three, preferably two, because of the number of possible rules.

The simplest way of variable reduction is a linear combination of two variables to one variable. There are some possibilities.

The first possibility is

$$X = x + 2L\theta_1 \qquad \text{(position of the pole tip)} \tag{25}$$

$$Y = \dot{x} + 2L\dot{\theta}_1 \qquad \text{(velocity of the pole tip)} \tag{26}$$

where X and Y are the new variables.

Equations (25) and (26) represent the position and the velocity of the tip of the pole, respectively. In order to judge whether the system is stable or not, we need the position and the velocity of the vehicle as well as (25) and (26). Because we cannot judge the stability of the pole with information of the top tip of the pole. We need information on the bottom tip of the pole related to the top tip. Thus (25) and (26) are incomplete.

The second possibility is

$$X = a\theta_1 + b\dot{x} \tag{27}$$

$$Y = c\dot{\theta}_1 + dx \tag{28}$$

where a, b, c, and d are constants. It is impossible to identify the physical meaning of X and Y in (27) and (28), respectively. Since the physical meaning of these new variables are not clear, it is impossible to represent the knowledge (fuzzy rules) by using intuitive fuzzy linguistic terms.

The third possibility is

$$X = A\theta_1 + B\dot{\theta}_1 \tag{29}$$

$$Y = Cx + D\dot{x} \tag{30}$$

where A, B, C, and D are positive constants. When θ_1 is positive and $\dot{\theta}_1$ is also positive, the pole is now falling down to the clockwise and X is big. When θ_1 is positive and $\dot{\theta}_1$ is negative, the pole is now returning to the equilibrium point from the positive angle (we do not feel the *emergency* so much) and X is smaller than the former case. Thus X seems to be a measure of emergency in the angle. In the similar manner, Y seems to be a measure of emergency in the position. Therefore, we can assign the label to these physical measures by intuition. For example, "X is negatively medium (NM) and Y is negatively medium (NM)" means the situation that the pole is falling down to counterclockwise (left hand side) and that the vehicle is now moving to the left from the negative position (i.e., the vehicle is going away from the set point $(x = 0)$)." In this situation, anybody will respond to move the vehicle to the left much faster at the sacrifice of accuracy of position to recover the pole. This strategy gives a fuzzy rule "If X is NM and Y is NM, then $\dot{x}$ should be NL," where negative velocity of the vehicle means the left direction.

In this manner, the fuzzy rules can be assigned in the rule map shown in Fig. 29(a). There is no need to assign all the possible rules. If each membership function is defined as Fig. 29(b), neighboring membership functions penetrate each other. Therefore, a defect of one rule can be compensated (interpolated) by surrounding four rules. Finally, the author obtained eleven rules to stabilize the glass with wine. Two bracketed rules are preferably added against the strong disturbance.

3) Control Systems and Experimental Results: In order to verify the utility of fuzzy inference, it is applied to a controller. The author presents two examples as systems under control. The one is a glass with wine on the plate attached to the inverted pendulum. The other is a mouse moving around on the plate attached to the inverted pendulum.

a) Glass with Wine: The block diagram of this equipment is shown in Fig. 30. A vehicle is driven by a dc servo motor through a flexible steelwire. The position x of the vehicle is detected by a potentiometer connected to the wheel which is driven by the flexible wire. x is differentiated by an analog differentiator and multiplied by a constant D to be fed to the adder. x is also multiplied by a constant C to be fed to the same adder. The output $Y = Cx + D\dot{x}$ is fed to one input of the fuzzy logic controller. On the other hand, the vehicle carries an brushless angle sensor employing hole effect, on the rotating rod of which bottom tip of the pole (inverted pendulum) is fixed. A plate is attached to the tip of this pole and a glass with wine is put on this plate. The angle of the pole θ_1 is detected by the angle sensor to be fed to the differentiator and multiplied by a constant $B. \theta_1$ is also multiplied by a constant $A. A \cdot \theta_1$ and $B \cdot \dot{\theta}_1$ are summed by the adder to be fed to the fuzzy logic controller. A fuzzy logic controller possesses 11 rule chips and 1 defuzzifier chip. A defuzzified inference result $\dot{x}$ is produced from the output of the controller to a servo driver to amplify the electric power which is applied to a dc servo motor to drive the vehicle as it should be.

The glass is successfully stabilized independent of the amount of wine and also independent of the length of support of the glass. This aspect is interesting. In case of mathematical approach, when a glass is changed, or wine is added or reduced, parameters m, r, L, and h should be changed and the stability condition will be changed. However, in case of fuzzy logic control, there is no need to change the rules nor the shapes of membership functions. Fuzzy boundary of a fuzzy linguistic term may absorb the deviation of parameters.

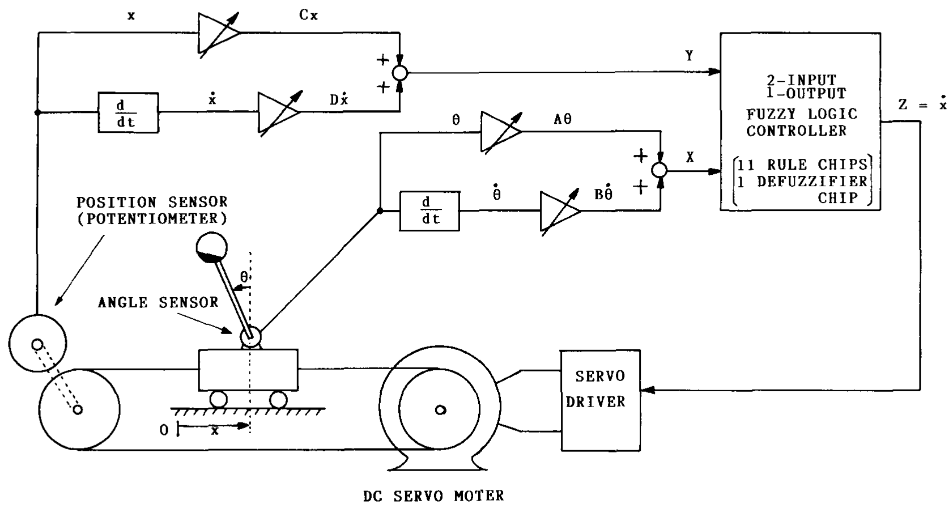


Fig. 30. A block diagram of a wine glass stabilization.

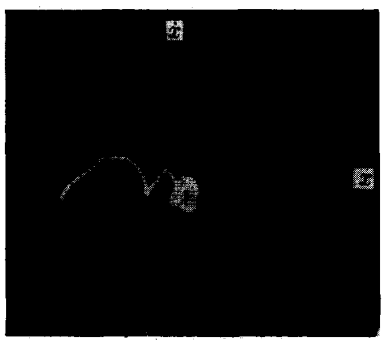


Fig. 31. Phase-plane portrait after applying a negative displacement (left) to the glass.

In order to examine the stability, a disturbance in the position was applied. Fig. 31 shows the phase-plane portrait after applying a negative displacement (left) to the glass. This portrait shows us that the vehicle comes up to the set point, swinging and balancing.

b) Mouse Moving Around: A glass with wine is not living. Therefore the movement or dynamics of this system can be described by mathematics, although it is very difficult. In other words, it may be modeled. On the other hand, let think about a living mouse moving around on the plate attached to the inverted pendulum. We cannot expect the movement nor describe the dynamics, because the center of gravity changes in accordance with the movement of the mouse. Thus we come to a deadlock again by the mathematical approach. However, the fuzzy linguistic approach can stabilize it. Fig. 32 shows the balancing motion. When a mouse moved to the right a little, the vehicle was driven by the fuzzy logic controller to the right to compensate the shift of the center of gravity at the sacrifice of the accuracy of the position.

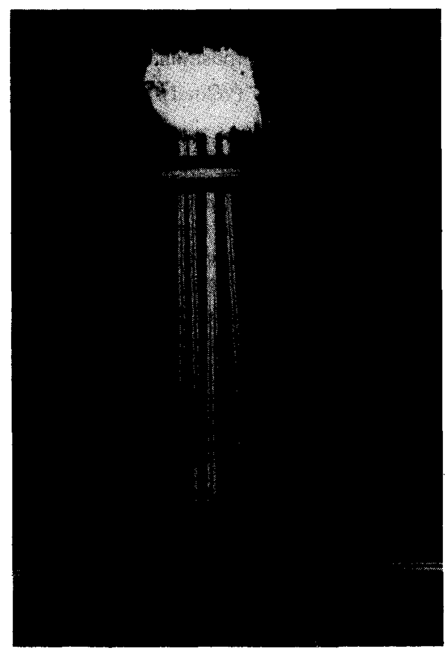


Fig. 32. Stabilization of a moving mouse which changes its center of gravity. The fuzzy logic controller can compensate the movement of the mouse at the sacrifice of the accuracy of the position.

C. Comparison with a Traditional PID Control

It is necessary for a designer to know the differences between a traditional PID control and a fuzzy logic control.

A typical control system (1-input and 1-output) employing a PID controller is shown in Fig. 33. The output state of the system under control is detected by a sensor. It is compared with the input signal (reference input) to derive an error signal

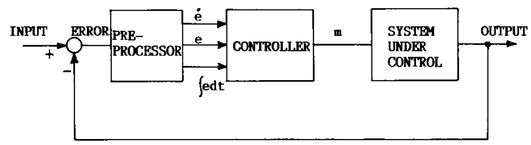


Fig. 33. A typical control system (1-input and 1-output) employing a PID controller.

$e(t)$, where the input signal is externally given and represents a desired state of the system. The error signal is delivered to a controller to produce an appropriate manipulating signal $m(t)$, which change the state of the system under control.

In a control system, there are two main objectives. The first one is to make the state (or output) of the system to be very close or equal, if possible, to the set point (or reference input). In other words, a small steady-state error $e(t)$ or a high steady-state accuracy is desired. The second one is to maintain the transient performance of the system within reasonable limits.

In order to design a controller of high steady-state accuracy and high speed settling, we need a linear combination of three control actions, i.e., proportional control action, integral control action and derivation (PID) control action. It is called a PID control and characterized by the following equation.

$$m(t) = K_P \cdot e(t) + K_I \int e(t)\,dt + K_D \frac{de(t)}{dt}. \tag{31}$$

This controller has three inputs and one output. Let us consider a controller of two inputs and one output to visualize the relationship between input and output. The first and the second terms in (31) is considered and differentiated to

$$\dot{m}(t) = K_P \dot{e}(t) + K_I e(t) \tag{32}$$

which is the linear description of a traditional PI controller.

Three-dimensional control surface in Fig. 34 shows the relation between the change of manipulator output and the linear combination of change of error input and error input, where the constant gains K_P and K_I are $\frac{2}{3}$ and $\frac{1}{3}$, respectively. Input-output characteristics of a fuzzy logic controller can be described by a set of fuzzy rules, antecedents and consequents of which corresponds to inputs and outputs of the controller, respectively. Input signal(s) applied to the controller and the fuzzy rules designed by the system designer produce the output signal(s) through fuzzy inferences and defuzzification(s) as described in Section IV-A and Section IV-B.

Here is presented an example of a fuzzy logic controller which has two inputs and provides one output. These two inputs are assigned to be change of error $\dot{e}(t)$ and error $e(t)$ and one output to the change of manipulating signal (control output) $\dot{m}(t)$ for comparison with the traditional PI controller. A set of fuzzy rules are indicated on the rule map in Table IV and membership functions used in these rules are shown in Fig. 35(a). Fig. 35(b) shows three-dimensional control surface of the *fuzzy PI controller,* the rules of which are given in Table IV.

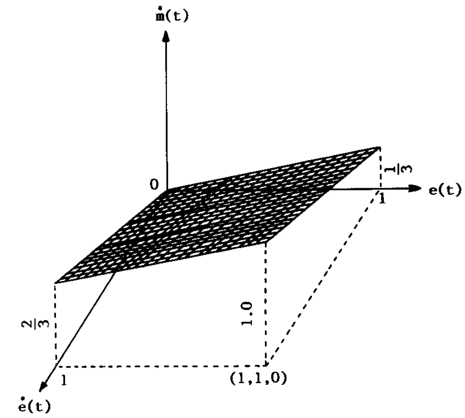


Fig. 34. The relation between the change of manipulator output and the linear combination of change of error input and error input. $K_P = \frac{2}{3}, K_I = \frac{1}{3}$. This can describe only a simple plane.

TABLE IV

$e(t)$ $e(t)$				
$\dot{e}(t)$	ZR	S	M	L
ZR	L	M	S	S
S	M	M	ZR	M
M	ZR	S	[S]	M
L	S	ZR	M	L

ZR: approximately zero, S: small, M: medium, L: large

The solid mesh in Fig. 35(b) is the inference result from fuzzy rules employing the solid membership function for "medium (M)" as shown in Fig. 35(a). When the left slope of the membership function of "medium" is reassigned as a dotted line illustrated in Fig. 35(a), then the circumference of the mesh is changed as illustrated by dotted lines in Fig. 35(b). Let us consider the case of $\dot{e}(t) = 1$ and $e(t) = 0.25$. When the membership function of "medium" is defined as the dotted line in Fig. 35(a), then $(\dot{e}(t), e(t)) = (1, 0.25)$ fires only one rule of 16 rules above. That is "If $\dot{e}(t)$ is L and $e(t)$ is S, then $\dot{m}(t)$ is ZR." So that after the inference followed by defuzzification, $\dot{m}(\dot{e}(t), e(t)) = \dot{m}(1, 0.25) = 0$ is obtained, i.e., (1, 0.25, 0) in Fig. 35(b). On the other hand, when the membership function "medium" is defined as a solid line in Fig. 39(a), then $(\dot{e}(t), e(t)) = 1, 0.25)$ fires two rules: Rule A "If $\dot{e}(t)$ is L and $e(t)$ is S, then $\dot{m}(t)$ is ZR" and Rule B "If $\dot{e}(t)$ is L and $e(t)$ is M, then $\dot{m}(t)$ is M." The degree of soft matching between (1, 0.25) and the antecedents of Rule A and Rule B are 1 and 0.5, respectively. Therefore, the conclusion $\dot{m}(1, 0.25)$ is not exclusively obtained from Rule A, but is affected by Rule B to produce a nonzero value. This means that at the typical point (for example, $e(t) = 0.25$ in "$e(t)$ is small"), grades of membership of other fuzzy linguistic terms should be zero for providing a typical conclusion.

In any way, Fig. 35(b) exhibits a very complicated and curved surface which can never be described by a linear combination of input variables such as (31) or (34). Furthermore, it is very easy to change the control surface (input-output

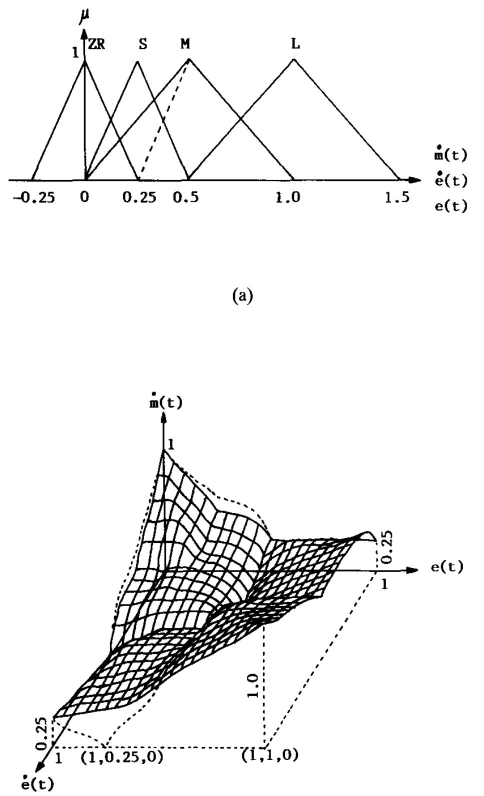


Fig. 35. (a) Membership functions used for getting the control surface. (b) The control surface of a fuzzy PI controller which achieves fuzzy inference with fuzzy rules given in Table IV. A set of fuzzy rules can describe a complicated control surface. A local change of the surface can be easily accomplished by changing the corresponding rule(s). So that a fuzzy system is suitable for learning systems, self-organizing systems, adaptive systems, etc.

characteristics) of the fuzzy logic controller in accordance with the change of the system under control. If you want to pull up the center of the control surface, you have only to change the label of the consequent in the corresponding rule, i.e., the label S (small) in the bracket in Table IV should be increased to be M (medium). If you want to change exclusively a small part of the control surface, you have to narrow the support of the corresponding membership function and increase the number of labels (fuzzy terms) to achieve the effective assignment of rules. This means the exceptional or irregular points on the control surface can be easily described in a fuzzy logic controller. Thus fuzzy rules are very suitable for describing a sophisticated system by fuzzy linguistic terms (intuitive terms). Of course, the fuzzy inference can be easily modified by the concepts of learning, self-organization, adaptation, etc.

VII. Conclusions

A brief explanation on the fuzzy set and fuzzy logic was made and followed by the comparison between mathematical description, linguistic description, and neural networks. The hardware to implement the fuzzy inference and the defuzzification is also described in detail. Lastly, a fuzzy logic controller was applied to stabilize a glass with wine put on the inverted pendulum and also a mouse moving around the plate on the inverted pendulum.

Distinctive features of a fuzzy logic controller can be summarized as follows.

1. How it should work is described with fuzzy natural languages and the structure of the knowledge is very clear. Therefore, the control strategy, know-hows, knowledge, or data are very easy to represent, very easy to understand, very easy to remember, very easy to debug.
2. Since all the rules are independent of each other, it is easy to update the rule to follow the change of the system under control. Thus a fuzzy system is suitable for a learning system, a self-organizing system, an adaptive system, etc.
3. It can accept the exceptional data (knowledge) and the contradictory data, so that the fuzzy logic controller is suitable for nonlinear and time-variant complicated systems.
4. Compound sensory signals such as outputs from a chemical sensor, an odor sensor and other contaminated cheap sensor can be accepted, as suggested in Section VI-B-2.
5. Since a fuzzy inference is one kind of interpolation with very few data, drastic reduction of data and software/hardware systems can be achieved.

The development of a fuzzy logic hardware system will be changed by another concepts. Several years also, we could find the tendency toward the fusion of fuzzy logic and neural networks [52]–[56]. Fusion of these two has lost much of its novelty now [57]–[60]. Future trends will be a fusion of fuzzy logic and chaos as well as neural networks. A fuzzy system is a modeling of a human brain summarized from the human expert's behavior and chaos is a nonlinear dynamical behavior generated by massive neural networks of the human brain.

Acknowledgment

The author would like to express his sincere thanks to Prof. Edgar Sáchez-Sinencio at Texas A&M University for giving an opportunity to the author to present this tutorial paper.

References

[1] C. L. Phillips and H. Troy Nagle, Ir., *Digital Control System Analysis and Design.* Englewood Cliffs, NJ: Prentice-Hall, 1984.
[2] R. C. Dorf, *Modern Control Systems.* New York: Addison-Wesley, 1980.
[3] K. Ogata, *Modern Control Engineering.* Englewood Cliffs, NJ: Prentice-Hall, 1970.
[4] L. A. Zadeh, "Fuzzy Sets," in *Information and Control.* New York: Academic Press, 1965, Vol. 8, pp. 338–353.
[5] V. Novak, *Fuzzy Sets and Their Applications.* New York: Adam Hilger, 1989.
[6] H.-J. Zimmermann, *Fuzzy Sets, Decision Making, and Expert Systems.* New York: Kluwer Academic, 1987.
[7] A. Kandel, *Fuzzy Expert Systems.* CRC Press, 1992.
[8] A. Kandel, *Fuzzy Mathematical Techniques with Applications.* New York: Addison-Wesley, 1986.
[9] H. J. Zimmermann, *Fuzzy Sets Theory—and Its Applications.* Kluwer Nijhoff, 1985.
[10] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information.* Englewood Cliffs, NJ: Prentice Hall, 1988.
[11] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applica-*

tions. New York: Academic Press, 1980.

[12] W. J. M. Kickert, *Fuzzy Theories on Decision-Making.* Boston, MA: Kluwer, 1978.

[13] C. V. Negoita and D. Ralescu, *Simulation, Knowledge-Based Computing, and Fuzzy Statistics.* New York: Van Nostrand Reinhold, 1987.

[14] A. Kandel, *Fuzzy Techniques in Pattern Recognition.* New York: Wiley, 1982.

[15] C. V. Negoita, *Fuzzy Systems.* Abacus Press, 1981.

[16] A. Kandel and S. C. Lee, *Fuzzy Switching and Automata.* Crane, Russak & Company, Inc., 1979.

[17] M. Zemankova-Leech and A. Kandel, *Fuzzy Relational Data Bases—A Key to Expert Systems.* Verlag TUV Rheinland, 1984.

[18] J. Kacprzyk, *Multistage Decision-Making Under Fuzziness.* Verlag TUV Rheinland, 1983.

[19] R. de Mori, *Computer Models of Speech Using Fuzzy Algorithm.* New York: Plenum Press, 1983.

[20] M. Smithson, *Fuzzy Set Analysis for Behavioral and Social Sciences.* New York: Springer-Verlag, 1987.

[21] S. Miyamoto, *Fuzzy Sets in Information Retrieval and Cluster Analysis.* New York: Kluwer Academic, 1990.

[22] W. Pedrycz, *Fuzzy Control and Fuzzy Systems.* New York: Wiley, 1989.

[23] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum Press, 1981.

[24] S. K. Pal and D. K. Dutta Majumdar, *Fuzzy Mathematical Approach to Pattern Recognition.* New York: Wiley, 1986.

[25] A. Kaufmann, *Introduction to the Theory of Fuzzy Subsets - Vol. 1-.* New York: Academic, 1975.

[26] G. E. Lasker, *Applied Systems and Cybernetics: Vol. VI - Fuzzy Sets and Fuzzy Systems, Possibility Theory and Special Topics in Systems Research.* New York: Pergamon Press, 1981.

[27] R. R. Yager, S. Ovchinnikov, R. M. Tong, and H. T. Nguyen, *Fuzzy Sets and Applications: Selected Papers by L. A. Zadeh.* New York: Wiley 1987.

[28] A. Kandel, *Fuzzy Mathematical Techniques with Applications.* New York: Addison-Wesley, 1986.

[29] M. Togai and H. Watanabe, "A VLSI implementation of a fuzzy-inference engine: Toward an expert system on a chip," *Information Sciences,* vol. 38, pp. 147–163, 1986.

[30] H. Watanabe, "RISC approach to design of fuzzy processor architecture," in *Proc. IEEE Int. Conf. Fuzzy Systems,* San Diego, CA, Mar. 8–12, 1992, pp. 431–440.

[31] Product Manual, "AT accelerator board," Togai InfraLogic, Inc., 1989.

[32] M. A. Manzoul and D. Jayabharathi, "Fuzzy controller on FPGA chip," in *Proc. IEEE Int. Conf. Fuzzy Systems,* San Diego, CA, Mar. 8–12, 1992, pp. 1309–1316.

[33] H. Eichfeld, M. Löhner, and M. Müller, "Architecture of a CMOS fuzzy logic controller with optimized memory organization and operator design," in *Proc. IEEE Int. Conf. Fuzzy Systems,* San Diego, CA, Mar. 8–12, 1992, pp. 1317–1323.

[34] T. Yamakawa, U.S. Patents: Fuzzy Logic Circuit, Pat. No. 4,694,418, Sept. 15, 1987, Appl. No. 714809, March 22, 1985; Multi-Functional Fuzzy Logic Circuit, Pat. No. 4,716.540, Dec. 29, 1987, Appl. No. 751447, July 3, 1985, Fuzzy Membership Function Circuit, Pat. No. 4,837,725, June 6, 1989, Appl. No. 917952, Oct. 14, 1986. European Patent: Fuzzy Logic Circuit, Pat. No. 0162225, Oct. 26, 1988, Appl. No. 0162225A1, March 22, 1985, Multi-Functional Fuzzy Logic Circuit, Appl. No. 0168004A2, July 5, 1985. Japanese Patent Applications, Tokuganshou No. 59-57121, March 23, 1984, No. 59-57122, Mar. 23, 1984, No. 59-57123, March 23, 1984, No. 59-57124, Mar. 23, 1984, No. 59-57125, Mar. 23, 1984, No. 59-141250, July 6, 1984, No. 59-141251, July 6, 1984, No. 59-141252, July 6, 1984, No. 59-187656, Sept. 6, 1984, No. 59-187657, Sept. 6, 1984, No. 59-263385, Dec. 13, 1984, No. 59-263386, Dec. 13, 1984, No. 60-234640, Oct. 22, 1985, No. 60-234641, Oct. 22, 1985, No. 60-234642, Oct. 22, 1985, No. 60-234643, Oct. 22, 1985, No. 60-234644, Oct. 22, 1985, No. 60-234645, Oct. 22, 1985, No. 60-234646, Oct. 22, 1985, No. 60-234647, Oct. 22, 1985, No. 60-234648, Oct. 22, 1985. T. Yamakawa and T. Miki, "The current mode fuzzy logic integrated circuits fabricated by the standard CMOS process," *IEEE Trans. Computers,* vol. C-35, pp. 161–167, 1986.

[35] T. Yamakawa, U.S. Patents: Fuzzy Logic Computers and Circuits, Pat. No. 4,875,184, Oct. 17, 1989, Appl. No. 116,777, Nov. 5, 1987; Fuzzy Membership Function Circuit, Pat. No. 5,113,366, May 12, 1992, Appl. No. 313,722, Mar. 14, 1989. European Patent: Multi-Functional Fuzzy Logic Circuit, Appl. No. 0168004, July 5, 1985; Australian Patent: MIN/MAX Circuits: Pat. No. 602817, Appl. No. 37139/89, June 28, 1989. Japanese Patent Applications, Tokuganshou, No. 61-268564, Nov. 13, 1986, No. 61-268565, Nov. 13, 1986, No. 61-268566, Nov. 13, 1986, No. 61-268567, Nov. 13, 1986, No. 61-268568, Nov. 13, 1986, No. 61-268569, Nov. 13, 1986, No. 63-49830, March 4, 1988, No. 63-203912, August 18, 1988, No. 63-206008, August 19, 1988, No. 63-206009, August 19, 1988, No. 63-282518, Nov. 10, 1988, No. 63-282519, Nov. 10, 1988. Japanese Patent Applications, Tokuganhei, No. 1-250624, Sept. 28, 1989, No. 2-114039, April 27, 1990, No. 3-322201, Dec. 6, 1991.

[36] T. Yamakawa and H. Kabuo, "A programmable fuzzifier integrated circuit—Synthesis, design, and fabrication," *Information Sciences,* vol. 45, pp. 75–112, 1988.

[37] F. Ueno, T. Inoue, Y. Shirai, and M. Sasaki, "A maximum and minimum circuit with multiple inputs in current mode," *The Transactions of the IEICE,* vol. E-70, no. 40, pp. 392–395, 1987.

[38] M. Sasaki, T. Inoue, Y. Shirai, and F. Ueno, "Fuzzy multiple-input maximum and minimum circuits in current mode and their analysis using bounded-difference equations," *IEEE Trans. Comput.,* vol. C-39, pp. 768–774, 1990.

[39] M. Sasaki, N. Ishikawa, F. Ueno, and T. Inoue, "Current-mode analog fuzzy hardware with voltage input interface and normalization locked loop," *IEICE Trans. Fundamentals,* vol. E75-A, no. 6, pp. 650–654, 1992.

[40] O. Ishizuka, K. Tanno, Z. Tang, and H. Matsumoto, "Design of a fuzzy controller with normalization circuits," in *Proc. IEEE Int. Conf. Fuzzy Systems,* San Diego, CA, Mar. 8–12, 1992, pp. 1303–1308.

[41] M. Morisue and Y. Kogure, "A super conducting fuzzy processor," in *Proc. IEEE Int. Conf. Fuzzy Systems,* San Diego, CA, Mar. 8–12, 1992, pp. 443–450.

[42] J. L. Huertas, S. Sánchez-Solano, A. Barriga, and I. Baturone, "Serial architecture for fuzzy controllers: Hardware inplementation using analog digital VLSI techniques," in *Proc. 2nd Int. Conf. Fuzzy Logic and Neural Networks,* Iizuka, Japan, July 17–22, 1992, pp. 535–538.

[43] T. Yamakawa and K. Sasaki, "Fuzzy memory device," in *Proc. Second IFSA Congress,* Tokyo, July 20–25, 1987, pp. 551–555.

[44] T. Yamakawa, "A simple fuzzy computer hardware system employing MIN & MAX operations—A challenge to 6th generation computer," in *Proc. Second IFSA Congress,* Tokyo, July 20–25, 1987, pp. 827–830.

[45] T. Yamakawa, "High-speed fuzzy controller hardware system: The mega FIPS machine," *Information Sciences,* vol. 45, pp. 113–128, 1988.

[46] T. Yamakawa, "Intrinsic fuzzy electronic circuits for sixth generation computer," in *Fuzzy Computing,* M. M. Gupta and T. Yamakawa, Eds. Amsterdam, The Netherlands: North Holland, 1988, pp. 157–171.

[47] T. Yamakawa, "Fuzzy microprocessors - Rule chip and defuzzifier chip," in *Proc. Int. Workshop on Fuzzy System Applications,* Iizuka, Japan, Aug. 20–24, 1988, pp. 51–52.

[48] T. Yamakawa, Japanese Patent Application, Tokuganshou, No. 63-205007, Aug. 19, 1988.

[49] T. Yamakawa, "An application of a grade-controllable membership function circuit to a singleton-consequent fuzzy logic controller," in *Proc. Third IFSA Congress,* Seattle, WA, Aug. 6–11, 1989, pp. 296–302.

[50] T. Yamakawa, Japanese Patent Application, Tokuganshou, No. 63-206008, Aug. 19, 1988.

[51] T. Yamakawa, "Stabilization of an inverted pendulum by a high-speed fuzzy logic controller hardware system," *Fuzzy Sets and Systems,* vol. 32, pp. 161–180, 1989.

[52] T. Yamakawa, Japanese Patent Applications, Tokuganhei, No. 1-043600, Feb. 23, 1989; No. 1-133690, May 26, 1989.

[53] T. Yamakawa, "A fuzzy neuron and its application to pattern recognition," in *Proc. Third IFSA Congress,* Seattle, WA, Aug. 6–11, 1989, pp. 30–38.

[54] T. Yamakawa, "Pattern recognition hardware system employing a fuzzy neuron," in *Proc. Int. Conf. Fuzzy Logic & Neural Networks,* Iizuka, Japan, July 20–24, 1990, pp. 943–948.

[55] T. Yamakawa, "A fuzzy neuron and its application to a hand-written character recognition system," in *Proc. IEEE Int. Symp. Circuits and Systems,* Singapore, June 11–14, 1991, pp. 1369–1372.

[56] *Proc. Int. Conf. Fuzzy Logic and Neural Networks,* Iizuka, Japan, July 20–24, 1990.

[57] T. Yamakawa, "A design algorithm of membership functions for a fuzzy neuron using example-based learning," in *Proc. IEEE Int. Conf. Fuzzy Systems,* San Diego, CA, Mar. 8–12, 1992, pp. 75–82.

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

[59] *Proc. A Second Int. Conf. Fuzzy Logic and Neural Networks,* Iizuka, Japan, July 17–22, 1992.

[60] T. Yamakawa, E. Uchino, T. Miki, and H. Kusanagi, "A neo fuzzy neuron and its applications to system identification and prediction of the system behavior," in *Proc. Second Int. Conf. Fuzzy Logic & Neural Networks,* Iizuka, Japan, July 17–22, 1992, pp. 477–484.

Takeshi Yamakawa received the B.Eng. degree in electronics engineering in 1969 from the Kyushu Institute of Technology, Tobata and the M.Eng. degree in electronics engineering in 1971 from Tohoku University, both in Japan. He received the Ph.D. degree for his studies on electrochemical devices in 1974 from Tohoku University, Japan.

From 1974 to 1977, he engaged in the development of new electrochemical devices as a Research Assistant at Tohoku University. From 1977 to 1981 he served as a Research Assistant in electrical engineering and computer science at Kumamoto University, Japan. From 1981 to 1989 he was an Associate Professor at Kumamoto University. He joined the faculty of Kyushu Institute of Technology, Iizuka, Japan and received a full professorship in April 1989. He established a foundation, Fuzzy Logic Systems Institute (FLSI), in Japan in 1990 to promote the international collaboration on fuzzy logic, neural networks and soft computing, and to prompt the spread of the research results. He is now a chairman of FLSI. His main research interest lies on hardware implementation of fuzzy systems, fuzzy neural networks, and chaotic systems. He developed fuzzy logic integrated circuits both in PMOS and CMOS processes, a fuzzy memory device, fuzzy microprocessors in analog mode, a fuzzy logic computer, a fuzzy neuron chip and its application to hand written character recognition, a chaos chip, and so on. He has applied for more than 74 patents in Japan, 8 patents in U.S.A., 3 patents in Europe and 1 patent in Taiwan. Dr. Yamakawa is a member of International Fuzzy Systems Association (IFSA), Japan Society of Fuzzy Theory and Systems (SOFT), the Institute of Electronics, Information and Communication Engineers (IEICE), the Society of Instrument and Control Engineers (SICE) and Japan Orthodontic Society. He acts as a member of editorial board and a regional editor of four international professional journal. He contributed more than 30 international conferences as an organizer or a member of organizing/programming committee. He organizes the International Conference on Fuzzy Logic, Neural Nets and Soft Computing (IIZUKA Conference) every two years in Iizuka, Japan.