

Fuzzy Logic Principles and Applications

11.1 INTRODUCTION

What is fuzzy logic (FL)? FL is another class of AI, but its history and applications are more recent than those of the expert system (ES). In Chapter 10, it was mentioned that according to George Boole, human thinking and decisions are based on “yes”/“no” reasoning, or “1”/“0” logic. Accordingly, Boolean logic was developed, and ES principles were formulated based on Boolean logic. It has been argued that human thinking does not always follow crisp “yes”/“no” logic, but is often vague, qualitative, uncertain, imprecise, or fuzzy in nature. For example, in terms of “yes”/“no” logic, a thinking rule may be

“IF it is not raining AND outside temperature is less than 80°F
THEN take a sightseeing trip for more than 100 miles”

In actual thinking, it might be

“IF weather is good AND outside temperature is mild
THEN take a long sightseeing trip”

Based on the nature of fuzzy human thinking, Lotfi Zadeh, a computer scientist at the University of California, Berkeley, originated the “fuzzy logic,” or fuzzy set theory, in 1965. In the beginning, he was highly criticized by the professional community, but gradually, FL captured the imagination of the professional community and eventually emerged as an entirely new discipline of AI. The general methodology of reasoning in FL and ES by “IF... THEN...”

statements or rules is the same; therefore, it is often called “fuzzy expert system.” For example, an ES rule for speed control in a variable-speed drive may be

IF speed of the motor is greater than 1500 rpm AND the machine stator temperature is between 60°F and 100°F
 THEN set the stator current i_{qs} less than 10 amps

The same rule in FL may read as

IF speed of the motor is high and stator temperature is medium
 THEN set the stator current i_{qs} low

FL can help to supplement an ES, and it is sometimes hybridized with the latter to solve complex problems. FL has been successfully applied in process control, modeling, estimation, identification, diagnostics, military science, stock market prediction, etc.

In this chapter, we will discuss the principles of FL and some of its applications in power electronic systems. The Fuzzy Logic Toolbox in the MATLAB environment, which includes an example program development, will then be introduced. Neuro-fuzzy systems will be discussed in Chapter 12.

11.2 FUZZY SETS

FL, unlike Boolean logic, deals with problems that have fuzziness or vagueness, as mentioned before. The classical set theory is based on Boolean logic, where a particular object or variable is either a member of a given set (logic 1), or it is not (logic 0). On the other hand, in fuzzy set theory based on FL, a particular object has a degree of membership in a given set that may be anywhere in the range of 0 (completely not in the set) to 1 (completely in the set). For this reason, FL is often defined as multi-valued logic (0 to 1), compared to bi-valued Boolean logic. It may be mentioned that although FL deals with imprecise information, the information is processed in sound mathematical theory, which has been advanced in recent years.

Before discussing the FL theory, it should be emphasized here that basically, a FL problem can be defined as an input/output, static, nonlinear mapping problem through a “black box,” as shown in Figure 11.1. All the input information is defined in the input space, it is processed in the black box, and the solution appears in the output space. In general, mapping can be static or dynamic, and the mapping characteristics are determined by the black box’s characteristics. The black box cannot only be a fuzzy system, but also an ES, neural network, general mathematical system, such as differential equations, algebraic equations, etc., or anything else.

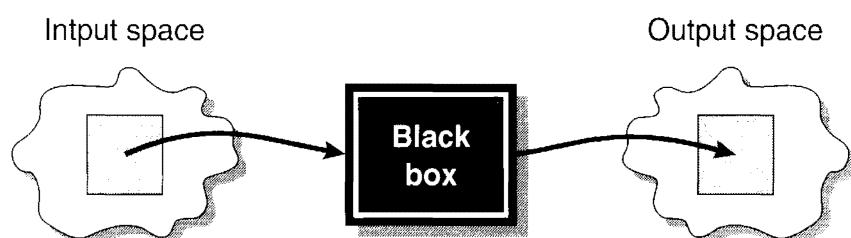
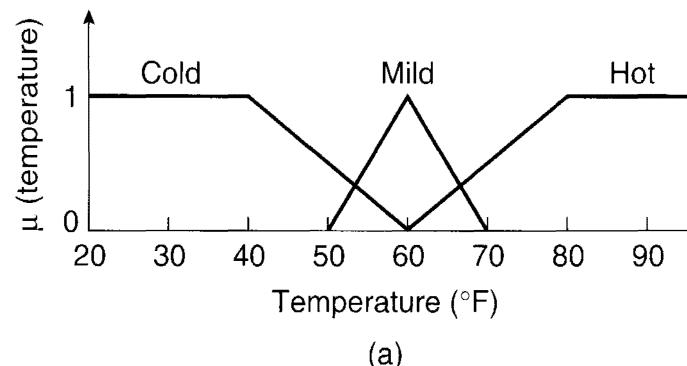


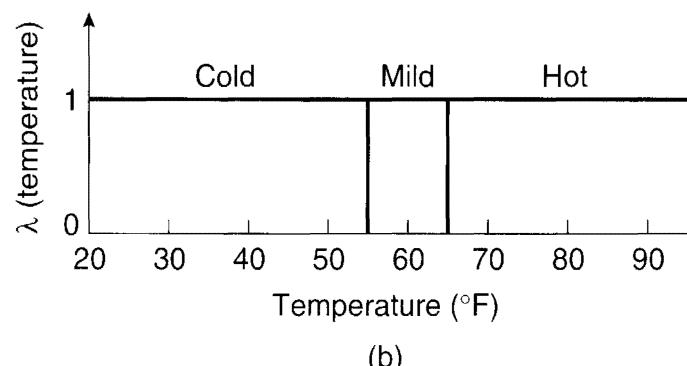
Figure 11.1 Input/output mapping problem

11.2.1 Membership Functions

A fuzzy variable has values that are expressed by the natural English language. For example, as shown in Figure 11.2(a), the stator temperature of a motor as a fuzzy variable can be defined by the qualifying linguistic variables Cold, Mild, or Hot, where each is represented by a triangular or straight-line segment membership function (MF). These linguistic variables are defined as fuzzy sets or fuzzy subsets. An MF is a curve that defines how the values of a fuzzy variable in a certain region are mapped to a membership value μ (or degree of membership) between 0 and 1. The fuzzy sets can have more subdivisions such as Zero, Very Cold, Medium Cold, Medium Hot, Very Hot, etc. for a more precise description of the fuzzy variable. In Figure 11.2(a), if the temperature is below 40° F, it belongs completely to the set Cold, that is, the MF value is 1; whereas for 55° F, it is in the set Cold by 30 percent ($\mu = 0.3$) and to the set Mild by 50 percent ($\mu = 0.5$). At temperature 60° F, it belongs completely to the set Mild ($\mu = 1$) and not in the set Cold and Hot ($\mu = 0$). If the temperature is above 80° F, it belongs completely to the set Hot ($\mu = 1$), where $\mu = 0$ for Cold and Mild. In Figure 11.2(b), the corresponding crisp or Boolean classification of the variable is given for comparison. For the temperature range below 55° F, it belongs to the set Cold ($\mu = 1$); between 55° F to 65° F, it belongs to the set Mild ($\mu = 1$); and above 65° F, it belongs to the set Hot only ($\mu = 1$). The sets are not members ($\mu = 0$) beyond the defined ranges. The numerical interval (20° F to 90° F) that is relevant for the description of a fuzzy variable is defined as the universe of discourse in Figure 11.2(a).



(a)



(b)

Figure 11.2 Representation of temperature using (a) Fuzzy sets, (b) Crisp set

An MF can have different shapes, as shown in Figure 11.3. The simplest and most commonly used MF is the triangular-type, which can be symmetrical or asymmetrical in shape. A trapezoidal MF (symmetrical or unsymmetrical) has the shape of a truncated triangle. Two MFs are built on the Gaussian distribution curve: a simple Gaussian curve and a two-sided composite of two different Gaussian curves. The bell MF with a flat top is somewhat different from a Gaussian function. Both the Gaussian and bell functions are smooth and non-zero at all points. A sigmoidal-type MF can be open to the right or left. Asymmetrical and closed (not open to the right or left) MFs can be synthesized using two sigmoidal functions, such as difference sigmoidal (difference between two sigmoidal functions) and product sigmoidal (product of two sigmoidals). Polynomial-based curves can have several functions, including asymmetrical polynomial curve open to the left (Polynomial-Z) and its mirror image, open to the right (Polynomial-S), and one that is zero at both ends but has a rise in the middle (Polynomial-Pi).

In addition to these types, any arbitrary MF can be generated by the user. In practice, one or two types of MFs (such as triangular and Gaussian) are more than enough to solve most problems. A singleton is a special type of MF that has a value of 1 at one point on the universe of discourse and zero elsewhere (a vertical spike). MFs can be represented by mathematical functions, segmented straight lines (for triangular and trapezoidal shapes), and look-up tables.

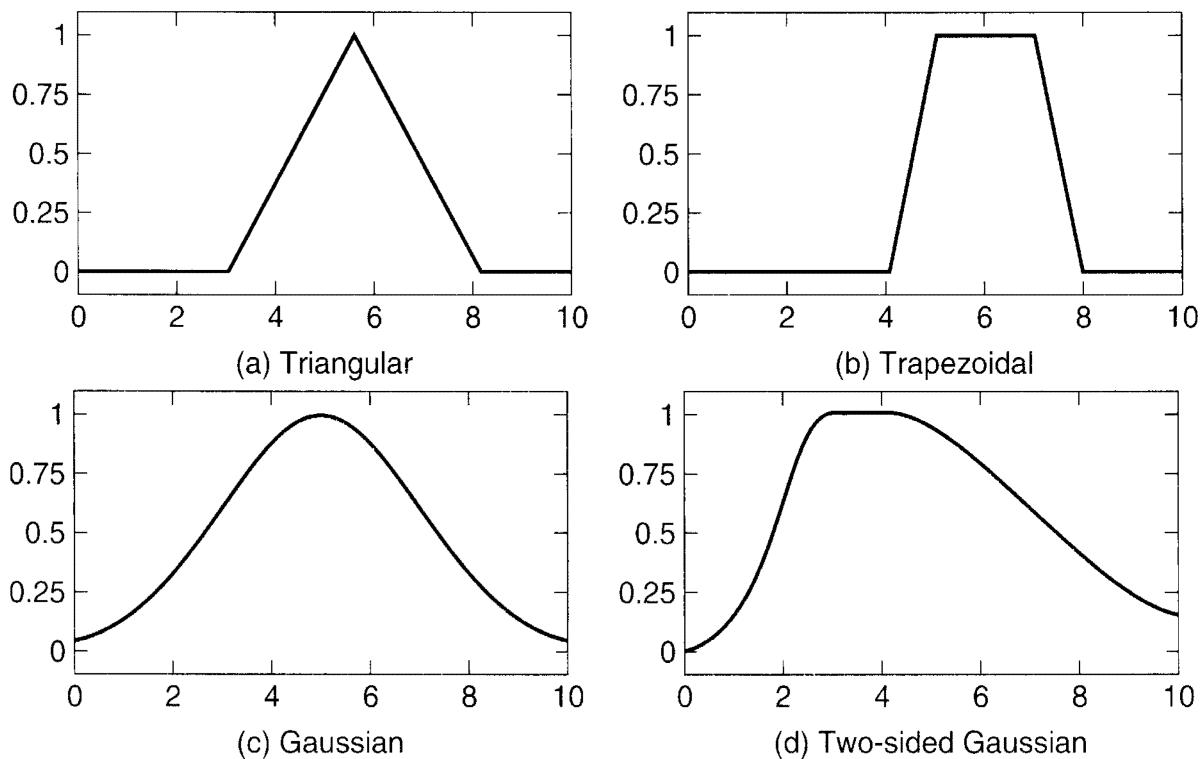


Figure 11.3 Different types of membership functions; (a) Triangular, (b) Trapezoidal, (c) Gaussian, (d) Two-sided Gaussian

(Continued)

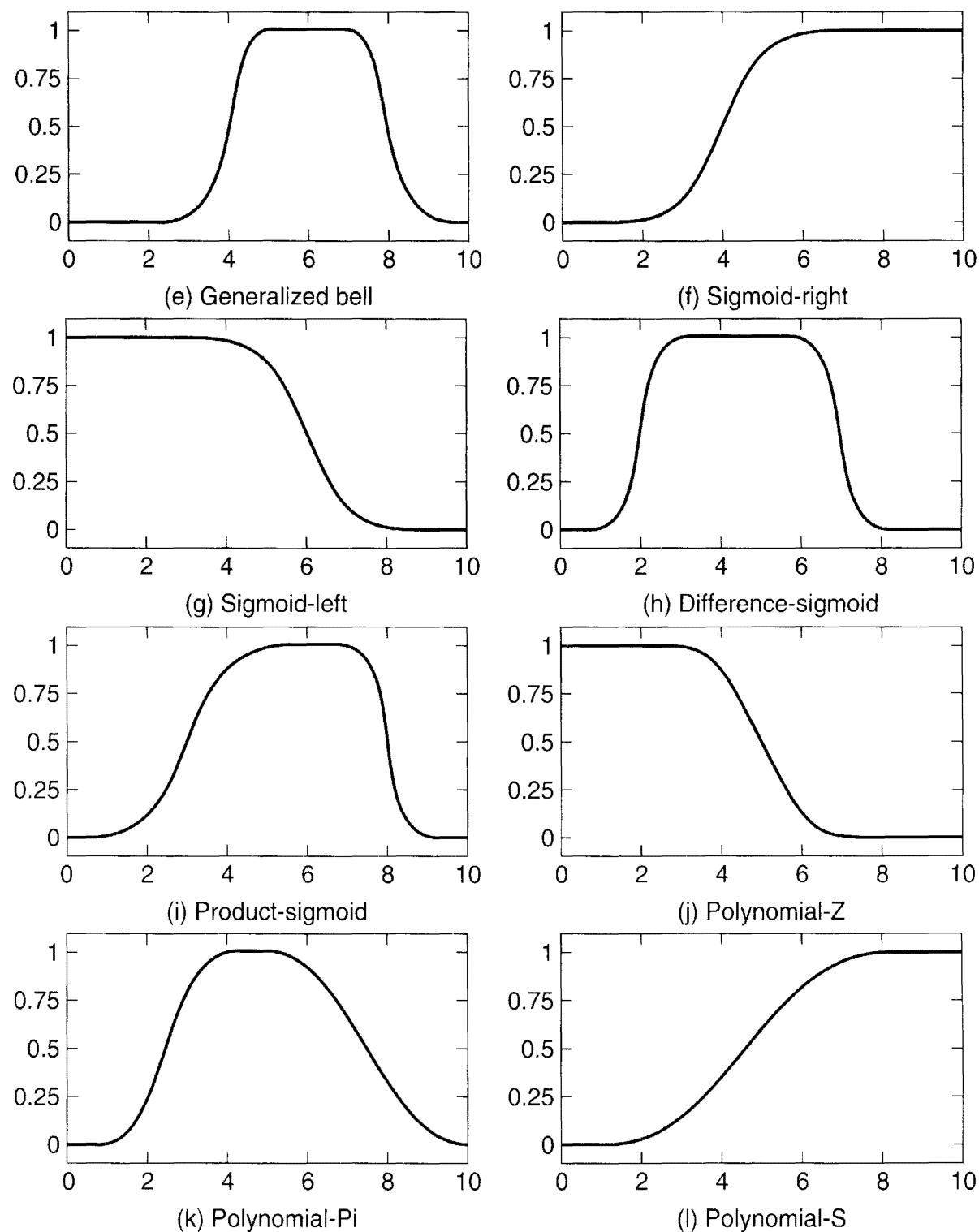


Figure 11.3 (Cont.) Different types of membership functions; (e) Generalized bell, (f) Sigmoid-right, (g) Sigmoid-left, (h) Difference-Sigmoid, (i) Product-Sigmoid, (j) Polynomial-Z, (k) Polynomial-Pi, (l) Polynomial-S

11.2.2 Operations on Fuzzy Sets

The basic properties of Boolean logic are also valid for FL. Figure 11.4 shows the logical operations of OR, AND, and NOT on fuzzy sets A and B using triangular MFs and compares them with the corresponding Boolean operations on the right. Let $\mu_A(x)$, $\mu_B(x)$ denote the degree of membership of a given element x in the universe of discourse X (denoted by $x \in X$).

Union: Given two fuzzy sets A and B , defined in the universe of discourse X , the union ($A \cup B$) is also a fuzzy set of X , with the membership function given as

$$\begin{aligned}\mu_{A \cup B}(x) &\equiv \max[\mu_A(x), \mu_B(x)] \\ &\equiv \mu_A(x) \vee \mu_B(x)\end{aligned}\quad (11.1)$$

where the symbol “ \vee ” is a maximum operator. This is equivalent to Boolean OR logic.

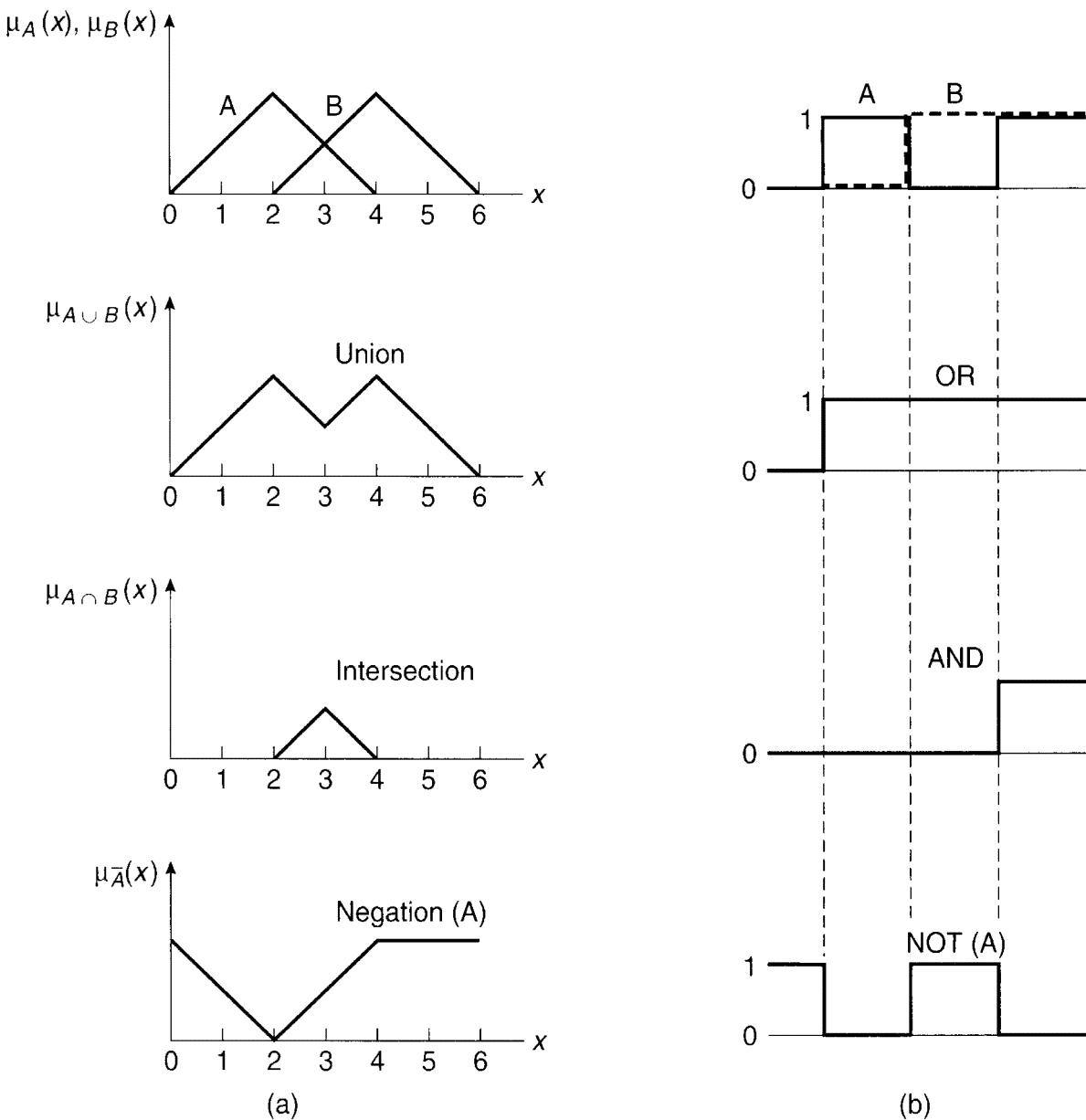


Figure 11.4 Logical operations of (a) Fuzzy sets, (b) Crisp sets

Intersection: The intersection of two fuzzy sets A and B in the universe of discourse X , denoted by $A \cap B$, has the membership function given by

$$\begin{aligned}\mu_{A \cap B}(x) &\equiv \min[\mu_A(x), \mu_B(x)] \\ &\equiv \mu_A(x) \wedge \mu_B(x)\end{aligned}\quad (11.2)$$

where “ \wedge ” is a minimum operator. This is equivalent to Boolean AND logic.

Complement or Negation: The complement of a given set A in the universe of discourse X is denoted by \bar{A} and has the membership function

$$\mu_{\bar{A}}(x) \equiv 1 - \mu_A(x) \quad (11.3)$$

This is equivalent to the NOT operation in Boolean logic.

In FL, we can also define the following operations:

Product of two fuzzy sets: The product of two fuzzy sets A and B defined in the same universe of discourse X is a new fuzzy set, $A.B$, with an MF that equals the algebraic product of the MFs of A and B ,

$$\mu_{A.B}(x) \equiv \mu_A(x) \cdot \mu_B(x) \quad (11.4)$$

which can be generalized to any number of fuzzy sets in the same universe of discourse.

Multiplying Fuzzy Set by a Crisp Number: The MF of fuzzy set A can be multiplied by a crisp number k to obtain a new fuzzy set called product $k.A$. Its MF is

$$\mu_{kA}(x) \equiv k \cdot \mu_A(x) \quad (11.5)$$

Power of a Fuzzy Set: We can raise fuzzy set A to a power m (positive real number) by raising its MF to m . The m power of A is a new fuzzy set, A^m , with MF

$$\mu_{A^m}(x) \equiv [\mu_A(x)]^m \quad (11.6)$$

Fuzzy set properties, as discussed above, are useful in performing additional operations using fuzzy sets. Consider the fuzzy sets A , B , and C defined over a common universe of discourse X . The following properties are valid for crisp and fuzzy sets, but some are more specific to fuzzy sets.

Double Negation:

$$\overline{\overline{A}} = A \quad (11.7)$$

Idempotency:

$$\begin{aligned}A \cup A &= A \\ A \cap A &= A\end{aligned}\quad (11.8)$$

Commutativity:

$$\begin{aligned} A \cap B &= B \cap A \\ A \cup B &= B \cup A \end{aligned} \tag{11.9}$$

Associative Property:

$$\begin{aligned} (A \cup B) \cup C &= A \cup (B \cup C) \\ (A \cap B) \cap C &= A \cap (B \cap C) \end{aligned} \tag{11.10}$$

Distributive Property:

$$\begin{aligned} A \cup (B \cap C) &= (A \cup B) \cap (A \cup C) \\ A \cap (B \cup C) &= (A \cap B) \cup (A \cap C) \end{aligned} \tag{11.11}$$

Absorption:

$$\begin{aligned} A \cap (A \cup B) &= A \\ A \cup (A \cap B) &= A \end{aligned} \tag{11.12}$$

De Morgan's Theorems:

$$\begin{aligned} \overline{A \cup B} &= \bar{A} \cap \bar{B} \\ \overline{A \cap B} &= \bar{A} \cup \bar{B} \end{aligned} \tag{11.13}$$

In fuzzy sets, all these properties can be expressed using the MF of the sets involved and the basic definition of union, intersection, and complement. For example, the distributive property in Equation (11.11) can be written in the

$$\begin{aligned} \mu_A(x) \vee (\mu_B(x) \wedge \mu_C(x)) &= (\mu_A(x) \vee \mu_B(x)) \wedge (\mu_A(x) \vee \mu_C(x)) \\ \mu_A(x) \wedge (\mu_B(x) \vee \mu_C(x)) &= (\mu_A(x) \wedge \mu_B(x)) \vee (\mu_A(x) \wedge \mu_C(x)) \end{aligned} \tag{11.14}$$

Similarly, the De Morgan's theorems in (11.13) can be written in the form

$$\begin{aligned} \overline{\mu_A(x) \vee \mu_B(x)} &= \mu_{\bar{A}}(x) \wedge \mu_{\bar{B}}(x) \\ \overline{\mu_A(x) \wedge \mu_B(x)} &= \mu_{\bar{A}}(x) \vee \mu_{\bar{B}}(x) \end{aligned} \tag{11.15}$$

11.3 FUZZY SYSTEM

A fuzzy inference system (or fuzzy system) basically consists of a formulation of the mapping from a given input set to an output set using FL, as indicated in Figure 11.1. This mapping process

provides the basis from which the inference or conclusion can be made. A fuzzy inference process consists of the following five steps:

- Step 1: Fuzzification of input variables
- Step 2: Application of fuzzy operator (AND, OR, NOT) in the IF (antecedent) part of the rule
- Step 3: Implication from the antecedent to the consequent (THEN part of the rule)
- Step 4: Aggregation of the consequents across the rules
- Step 5: Defuzzification

Let us first take a simple non-technical example of FL application and illustrate all the above five steps. Figure 11.5 shows a typical fuzzy inference system for restaurant tipping, where Food and Service are the input fuzzy variables (0–10 range) and Tip is the output fuzzy variable (0–25% range). The output is the aggregation of the evaluation of the three rules shown in the system. Normally, the tipping rules are evaluated in our mind to come to the decision of how much tip we should give. But, fuzzy systems theory can be applied to compute the precise output, which is explained in Figure 11.6.

In this example, the input variable Service is represented by three fuzzy sets (see Figure 11.2) Poor, Good, and Excellent, which correspond to curved MFs. The variable Food is represented by two fuzzy sets, Bad and Delicious, which correspond to straight-line MFs. The output variable Tip is represented by three sets Cheap, Average, and Generous, which correspond to triangular MFs. The universe of discourse for the input variables is 0–10, whereas for the output variable is 0%–25%. The processing of the three rules 1, 2, and 3 in the horizontal direction is shown in the figure. Consider, for example, that the score of the quality of service is 3. This crisp input, when referred to MF Poor, gives the output $\mu = 0.3$, which is the result of fuzzification (Step 1). If the score for Food is 8 and is referred to MF Bad, the result of fuzzification is $\mu = 0$, as shown in the figure. Once the

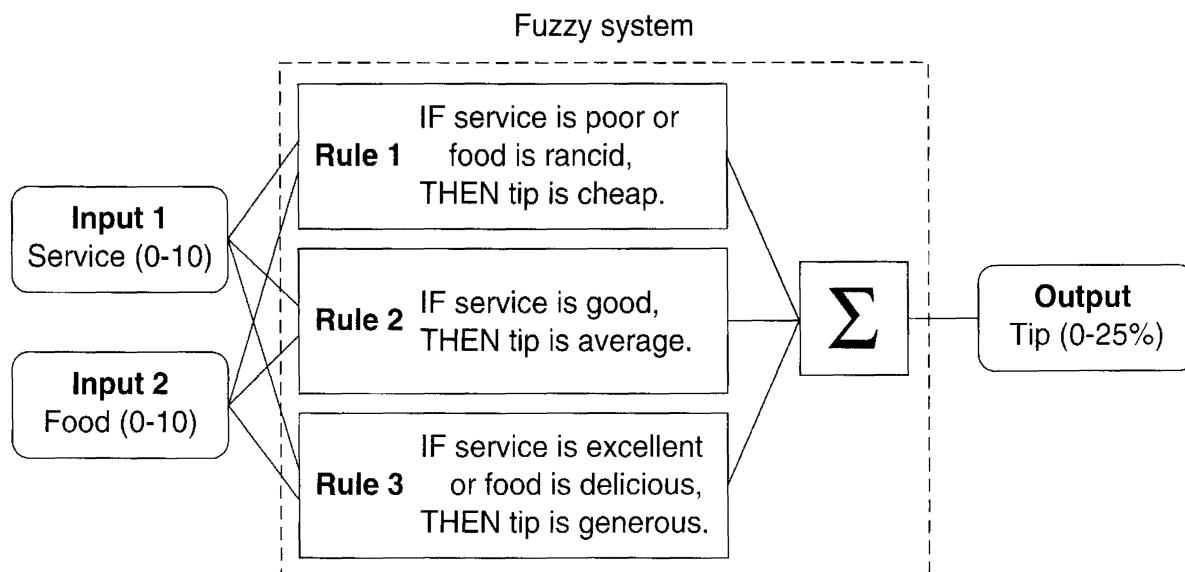


Figure 11.5 Fuzzy inference system for restaurant tipping [3]

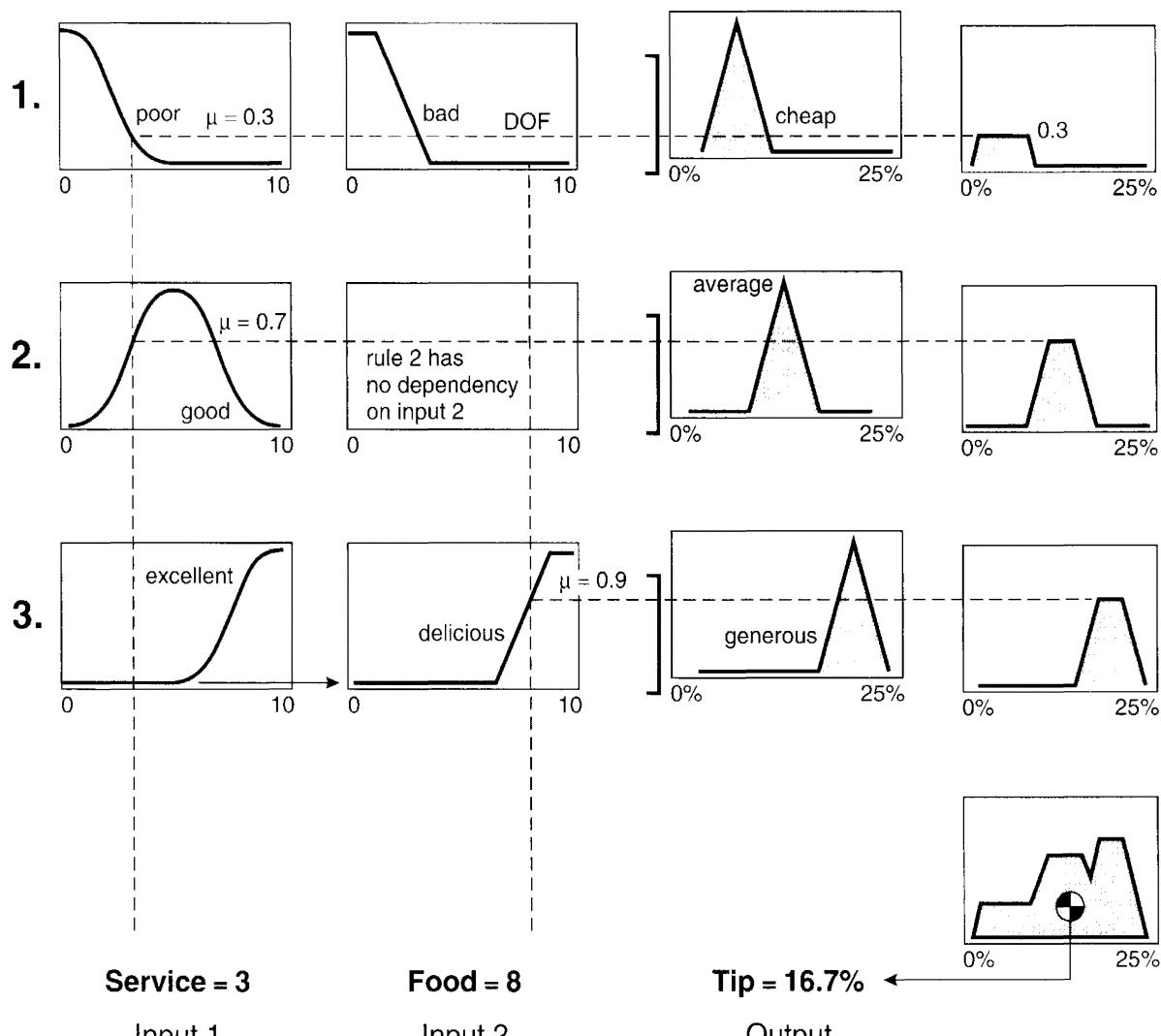


Figure 11.6 Information processing in fuzzy system for restaurant tipping

inputs have been fuzzified, we know the degree to which each part of the antecedent of a rule has been satisfied. In the rule, the OR or max operator is specified, and therefore, between the two values, 0.3 and 0, the result of the fuzzy operator is 0.3, that is, the 0.3 value is selected (Step 2). This is also defined as the degree of fulfillment (DOF) of a rule. If, on the other hand, the rule contains an AND or minimum operator, the value 0 will be selected. The implication step helps to evaluate the consequent part of a rule. In this rule, the output MF Cheap is truncated at the value $\mu = 0.3$ to give the fuzzy output (Step 3) shown. All three rules are evaluated in the same manner and their contributions are shown on the right. These outputs are combined or aggregated in a cumulative manner to result in the final fuzzy output (Step 4) shown at the bottom right of the figure. Finally, the fuzzy output (area) is converted to crisp output (Tip = 16.7%); a single number, which is defined as defuzzification (Step 5). Typically, it is the centroid or center of gravity of the area. Note that in Figure 11.5, the information is processed in the forward direction only in a parallel manner and the input/output mapping property is evident.

So far, we have given a simple example of a fuzzy restaurant tipping system to clarify some of the concepts in a fuzzy system. In general, there will be a matrix of rules similar to the

ES rule matrix shown in Figure 10.1. If, for example, there are 7 MFs for input variable X and 5 MFs for input variable Y , then there will be all together $7 \times 5 = 35$ rules (see Table 11.1).

11.3.1 Implication Methods

The implication step (Step 3) was introduced in the above example for the evaluation of individual rules. There are a number of implication methods in the literature, so we will study a few of the types that are frequently used.

11.3.1.1 Mamdani Type

Mamdani, one of the pioneers in the application of FL in control systems, proposed this implication method which, in fact, has been applied in the above tipping example. This is the most commonly used implication method. Let us again consider three rules in a fuzzy system, which are given in general form given as

Rule 1: IF X is negative small (NS) AND Y is zero (ZE)
THEN Z is positive small (PS)

Rule 2: IF X is zero (ZE) AND Y is zero (ZE)
THEN Z is zero (ZE)

Rule 3: IF X is zero (ZE) AND Y is positive small (PS)
THEN Z is negative small (NS)

where X and Y are the input variables, Z is the output variable, and NS, ZE, and PS are the fuzzy sets. Figure 11.7 explains the fuzzy inference system with the Mamdani method for inputs $X = -3$ and $Y = 1.5$. Note that all the rules have an AND operator. From the figure, the DOF of Rule 1 can be given as

$$DOF_1 = \mu_{NS}(X) \wedge \mu_{ZE}(Y) = 0.8 \wedge 0.6 = 0.6 \quad (11.16)$$

where \wedge = minimum operator and $\mu_{NS}(X)$ and $\mu_{ZE}(Y)$ are the MFs of X and Y , respectively. The rule output is given by the truncated MF PS', as shown in the figure. Similarly, for Rules 2 and 3, we can write

$$DOF_2 = \mu_{ZE}(X) \wedge \mu_{ZE}(Y) = 0.4 \wedge 0.6 = 0.4 \quad (11.17)$$

$$DOF_3 = \mu_{ZE}(X) \wedge \mu_{PS}(Y) = 0.4 \wedge 1.0 = 0.4 \quad (11.18)$$

The corresponding fuzzy output MFs are ZE' and NS', respectively, as indicated in the figure. The total fuzzy output is the union (OR) of all the component MFs,

$$\mu_{OUT}(Z) = \mu_{PS'}(Z) \vee \mu_{ZE'}(Z) \vee \mu_{NS'}(Z) \quad (11.19)$$

which is shown in the lower right part of the figure. The defuzzification to convert the fuzzy output to crisp output will be discussed later.

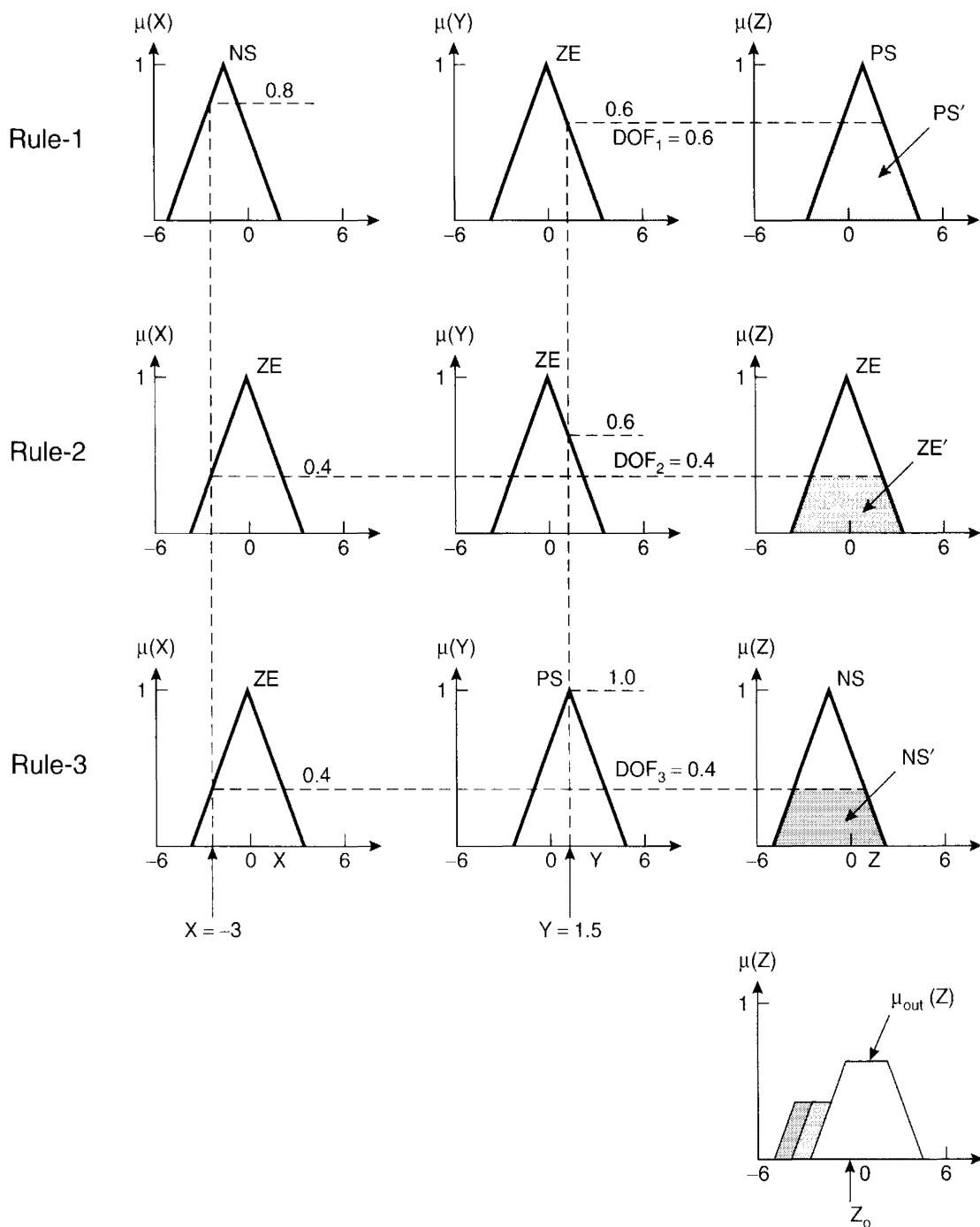


Figure 11.7 Three-rule fuzzy system using Mamdani method

11.3.1.2 Lusing Larson Type

In this method, the output MF is scaled instead of being truncated, as shown in Figure 11.8. In this case, the same three rules are considered as well as the same inputs, of $X = -3$ and $Y = 1.5$ giving DOFs of $\text{DOF}_1 = 0.6$, $\text{DOF}_2 = 0.4$, and $\text{DOF}_3 = 0.4$. The output MF PS of Rule 1 is scaled so that the output is PS' with a peak value of 0.6 as shown. Similarly, Rules 2 and 3 give output MFs ZE' and NS' , each with a peak value of 0.4 as indicated. The total output MF is given by Equation (11.19). The output area is somewhat different from that of the Mamdani method, and the corresponding crisp output will differ slightly.

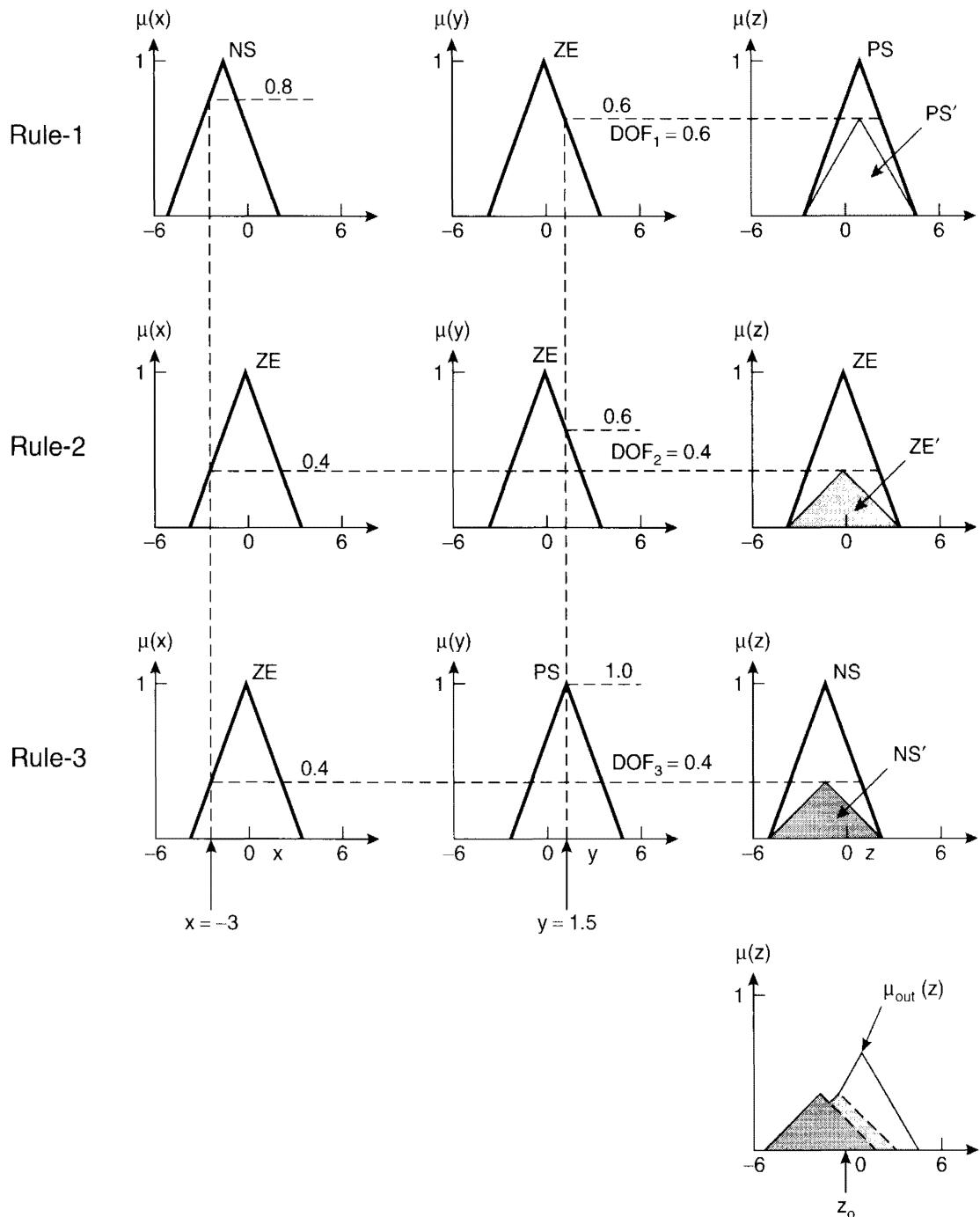


Figure 11.8 Three-rule fuzzy system using Lusing Larson method

11.3.1.3 Sugeno Type

The Sugeno, or Takagi-Sugeno-Kang method of implication was first introduced in 1985. The difference here is that unlike the Mamdani and Lusing Larson methods, the output MFs are only constants or have linear relations with the inputs. With a constant output MF (singleton), it is defined as the zero-order Sugeno method, whereas with a linear relation, it is

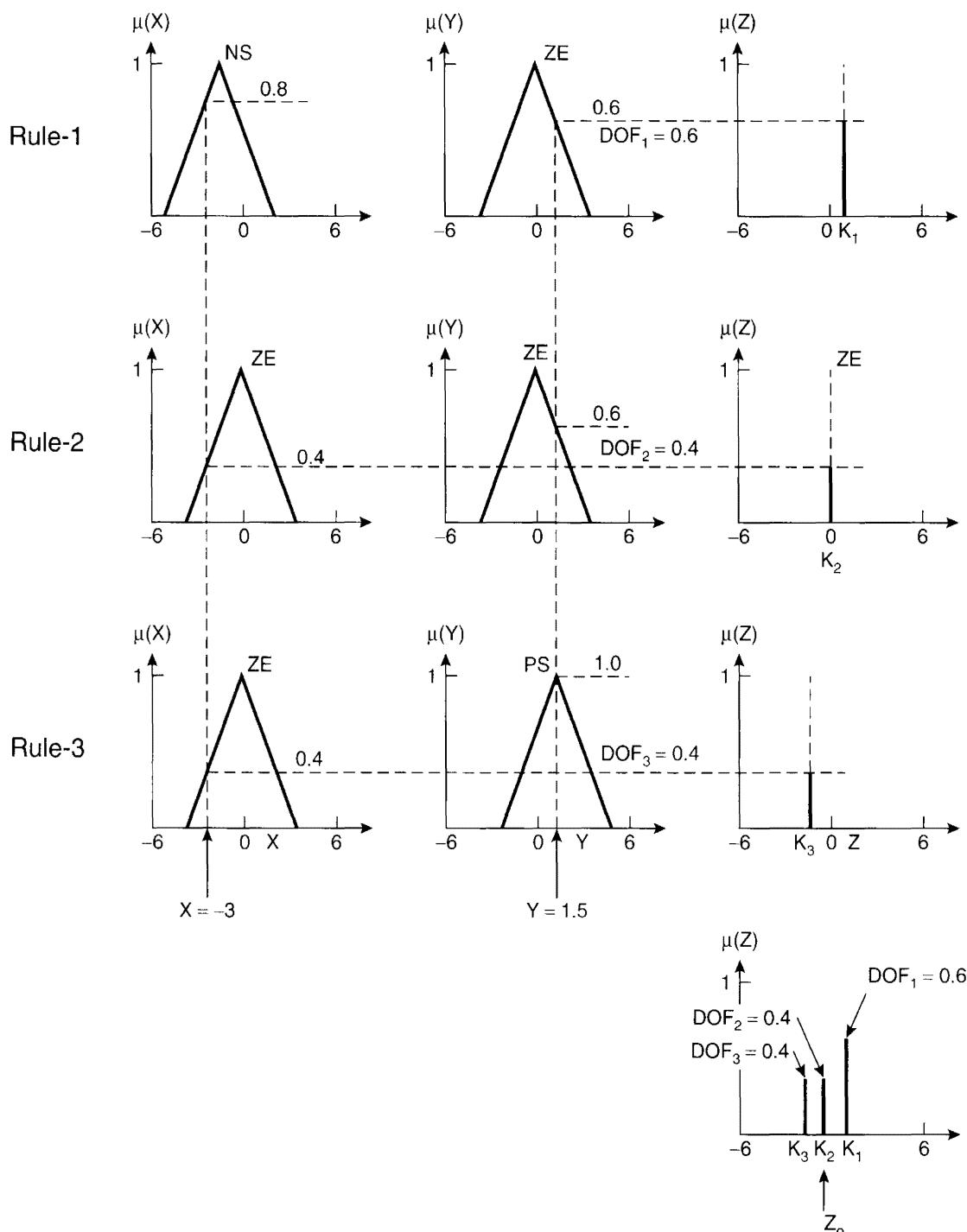


Figure 11.9 Three-rule fuzzy system using Sugeno (zero-order) method

known as the first-order Sugeno method. Figure 11.9 shows a three-rule fuzzy system using the Sugeno zero-order method where the rules read as

Rule 1: IF X is NS AND Y is ZE
THEN $Z = K_1$

Rule 2: IF X is ZE AND Y is ZE
THEN $Z = K_2$

Rule 3: IF X is ZE AND Y is PS
THEN $Z = K_3$

The constants K_1 , K_2 , and K_3 are crisply defined constants, as shown in the figure, in the consequent part of each rule. The output MF in each rule is a singleton spike, which is multiplied by the respective DOF to contribute to the fuzzy output of each rule. These MFs (truncated vertical segments) are then aggregated to constitute the total fuzzy output as shown in the figure. Fortunately, it can be shown that if the Mamdani, Lusing Larson, and Sugeno methods are applied to the same problem, the output will be approximately the same.

The more general first-order Sugeno method, as shown in Figure 11.10, has the rules in the form

Rule 1: IF X is NS AND Y is ZE
 THEN $Z = Z_1 = A_{01} + A_{11}X + A_{21}Y$

Rule 2: IF X is ZE AND Y is ZE
 THEN $Z = Z_2 = A_{02} + A_{12}X + A_{22}Y$

Rule 3: IF X is ZE AND Y is PS
 THEN $Z = Z_3 = A_{03} + A_{13}X + A_{23}Y$

where all the A 's are constants. An easy way to visualize a first-order system is to think of each rule as defining the location of a moving singleton, that is, the singleton output spikes can move around in a linear fashion in the output space depending on the input signal values. Higher order Sugeno methods are also possible, but are not of much practical use. In power electronics applications, we will later give one example with a Sugeno first-order method and the remaining examples with the Mamdani method. The Sugeno method is widely used in adaptive neuro-fuzzy inference systems (ANFISs) which are discussed in Chapter 12.

11.3.2 Defuzzification Methods

So far, we have discussed the different steps of a fuzzy inference system, except the final defuzzification method. The result of the implication and aggregation steps is the fuzzy output, which is the union of all the outputs of individual rules that are validated or “fired.” Conversion of this fuzzy output to crisp output is defined as defuzzification. We will now discuss a few important methods of defuzzification.

11.3.2.1 Center of Area (COA) Method

In the COA method (often called the center of gravity method) of defuzzification, the crisp output Z_0 of the Z variable is taken to be the geometric center of the output fuzzy value $\mu_{out}(Z)$ area, where $\mu_{out}(Z)$ is formed by taking the union of all the contributions of rules whose DOF > 0 (see, for example, the lower right part of Figure 11.7). The general expression for COA defuzzification is

$$Z_0 = \frac{\int Z \cdot \mu_{out}(Z)}{\int \mu_{out}(Z) dZ} \quad (11.20)$$

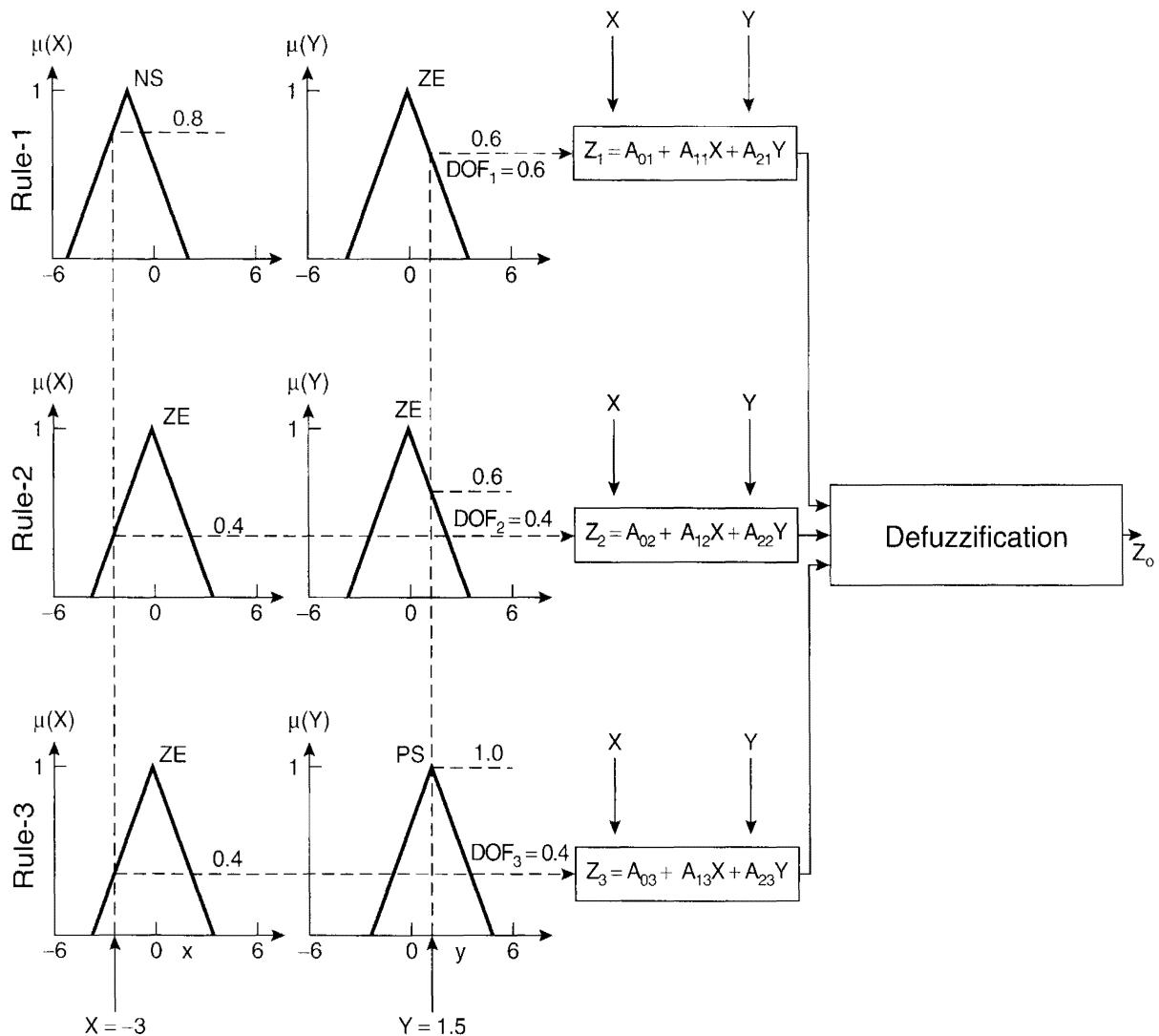


Figure 11.10 Three-rule fuzzy system using Sugeno (first-order) method

With a discretized universe of discourse, the expression is

$$Z_0 = \frac{\sum_{i=1}^n Z_i \mu_{out}(Z_i)}{\sum_{i=1}^n \mu_{out}(Z_i)} \quad (11.21)$$

For example, Figure 11.11 shows a simple fuzzy output for a two-rule system where the COA formula gives crisp output as

$$Z_0 = \frac{1 \cdot \frac{1}{3} + 2 \cdot \frac{2}{3} + 3 \cdot \frac{2}{3} + 4 \cdot \frac{2}{3} + 5 \cdot \frac{1}{3} + 6 \cdot \frac{1}{3} + 7 \cdot \frac{1}{3}}{\frac{1}{3} + \frac{2}{3} + \frac{2}{3} + \frac{2}{3} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3}} = 3.7 \quad (11.22)$$

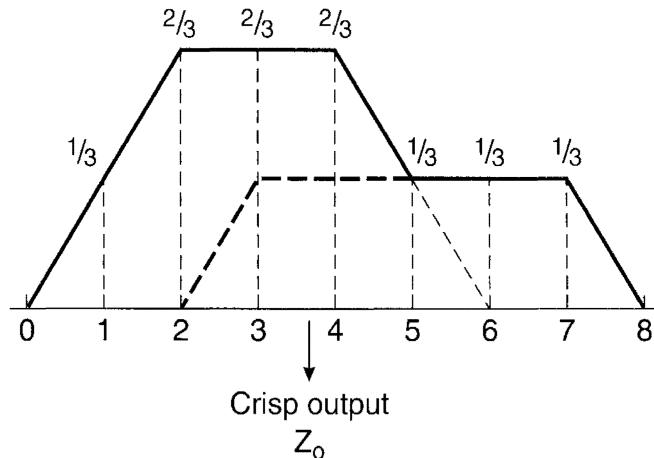


Figure 11.11 Defuzzification of output for a two-rule system

COA defuzzification is a well-known method and it is often used in spite of some amount of complexity in the calculation. Note that if the areas of two or more contributing rules overlap, the overlapping area is counted only once, which should be evident from Figure 11.11.

11.3.2.2 Height Method

In the height method of defuzzification, the COA method is simplified to consider only the height of each contributing MF at the mid-point of the base. For example, in Figure 11.11,

$$Z_0 = \frac{3 \cdot \frac{2}{3} + 5 \cdot \frac{1}{3}}{\frac{2}{3} + \frac{1}{3}} = 3.67 \quad (11.23)$$

which is slightly less than the 3.7 obtained by COA method.

11.3.2.3 Mean of Maxima (MOM) Method

The height method of defuzzification is further simplified in the MOM method, where only the highest membership function component in the output is considered. According to this method, the output $Z_0 = 3$ in Figure 11.11. If M such maxima are present, then the formula is

$$Z_0 = \sum_{m=1}^M \frac{Z_m}{M} \quad (11.24)$$

where $Z_m = m^{\text{th}}$ element in the universe of discourse, where the output MF is at the maximum value, and $M = \text{number of such elements}$.

11.3.2.4 Sugeno Method

In the Sugeno method, defuzzification is very simple. For example, in the zero-order method shown in Figure 11.9, the defuzzification formula is

$$Z_0 = \frac{K_1 \cdot DOF_1 + K_2 \cdot DOF_2 + K_3 \cdot DOF_3}{DOF_1 + DOF_2 + DOF_3} \quad (11.25)$$

whereas in the first-order method shown in Figure 11.10, the formula is

$$Z_0 = \frac{Z_1 \cdot DOF_1 + Z_2 \cdot DOF_2 + Z_3 \cdot DOF_3}{DOF_1 + DOF_2 + DOF_3} \quad (11.26)$$

11.4 FUZZY CONTROL

11.4.1 Why Fuzzy Control?

The control algorithm of a process that is based on FL or a fuzzy inference system, as discussed above, is defined as a fuzzy control. In general, a control system based on AI is defined as intelligent control. A fuzzy control system essentially embeds the experience and intuition of a human plant operator, and sometimes those of a designer and/or researcher of a plant. The design of a conventional control system is normally based on the mathematical model of a plant. If an accurate mathematical model is available with known parameters, it can be analyzed, for example, by a Bode or Nyquist plot, and a controller can be designed for the specified performance. Such a procedure is tedious and time-consuming, although CAD programs are available for such design. Unfortunately, for complex processes, such as cement plants, nuclear reactors, and the like, a reasonably good mathematical model is difficult to find. On the other hand, the plant operator may have good experience for controlling the process.

Power electronics system models are often ill-defined. Even if a plant model is well-known, there may be parameter variation problems. Sometimes, the model is multivariable, complex, and nonlinear, such as the dynamic $d-q$ model of an ac machine. Vector or field-oriented control of a drive can overcome this problem, but accurate vector control is nearly impossible, and there may be a wide parameter variation problem in the system. To combat such problems, various adaptive control techniques were discussed in Chapter 8. Fuzzy control, on the other hand, does not strictly need any mathematical model of the plant. It is based on plant operator experience and heuristics, as mentioned previously, and it is very easy to apply. Fuzzy control is basically an adaptive and nonlinear control, which gives robust performance for a linear or nonlinear plant with parameter variation. In fact, fuzzy control is possibly the best adaptive control among the techniques discussed so far.

11.4.2 Historical Perspective

The history of fuzzy control applications is very interesting. Since the development of FL theory by Zadeh in 1965, its first application to control a dynamic process was reported by

Mamdani in 1974, and by Mamdani and Assilian in 1975. These were extremely significant contributions because they stirred widespread interest by later workers in the field. Mamdani and Assilian were concerned with the control of a small laboratory steam engine. The control problem was to regulate the engine speed and boiler steam pressure by means of the heat applied to the boiler and the throttle setting of the engine. The process was difficult because it was nonlinear, noisy, and strongly coupled, and no mathematical model was available. The fuzzy control designed purely from the operator's experience by a set of IF...THEN rules was found to perform well and was better than manual control.

In 1976, Kickert and Lemke examined the fuzzy control performance of an experimental warm water plant, where the problem was to regulate the temperature of water leaving a tank at a constant flow rate by altering the flow of hot water in a heat exchanger contained in the tank. The success of Mamdani and Assilian's work led King and Mamdani (1977) to attempt to control the temperature in a pilot-scale batch chemical reactor by a fuzzy algorithm. Tong (1976) also applied FL to the control of a pressurized tank containing liquid. These results indicated that fuzzy control was very useful for complex processes and gave superior performance over conventional P-I-D control.

FL applications in power electronics and motor drives are somewhat recent. Li and Lau (1989) applied FL to a microprocessor-based servo motor controller, assuming a linear power amplifier. They compared the fuzzy-controlled system performance with that of P-I-D control and MRAC and demonstrated the superiority of the fuzzy system. Da Silva et al. (1987) developed a fuzzy adaptive controller and applied it to a four-quadrant power converter for the first time. Gradually, fuzzy control gathered momentum to other applications in the power electronics and drives areas.

11.4.3 Control Principle

Consider the fuzzy speed controller block in a vector-controlled drive system shown in Figure 11.12. The controller observes the pattern of the speed loop error signal and correspondingly updates the output DU so that the actual speed ω_r matches the command speed ω_r^* . There

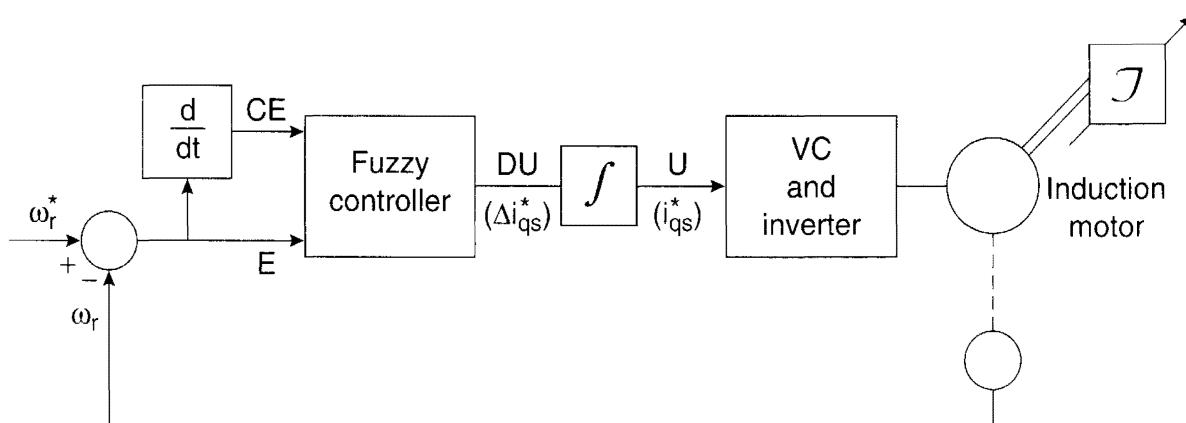


Figure 11.12 Fuzzy speed controller in vector-controlled drive system with variable moment of inertia (J)

are two input signals to the fuzzy controller, the error $E = \omega_r^* - \omega_r$ and the change in error, CE , which is related to the derivative dE/dt of error. In a discrete system, $dE/dt = \Delta E/\Delta t = CE/T_s$, where $CE = \Delta E$ in the sampling time T_s . With constant T_s , CE is proportional to dE/dt . The controller output DU in a vector-controlled drive is Δi_{qs}^* current. This signal is summed or integrated to generate the actual control signal U or current i_{qs}^* . From the physical operation principle of the system, we can write a simple control rule in FL as

IF E is near zero (ZE) AND CE is slightly positive (PS)
THEN the controller output DU is small negative

where E and CE are the input fuzzy variables, DU is the output fuzzy variable, and ZE, PS, and NS are the corresponding fuzzy set MFs. The implementation of this fuzzy control is illustrated in Figure 11.13 with the help of triangular MFs. With the values of $E = -1$ and $CE = 1.8$, as shown, the Mamdani method will give the output DU or $\Delta i_{qs}^* = -2$ amps. As discussed in the previous section, generally more than one fuzzy rule is fired and the contribution of the individual rules is combined at the output. Figure 11.14 illustrates the principle of two-rule control with the Mamdani method where the rules are

Rule 1: IF $E = ZE$ AND $CE = NS$ THEN $DU = NS$

Rule 2: IF $E = PS$ AND $CE = NS$ THEN $DU = ZE$

where DU represents the output. For the given rule base of the control system, the fuzzy controller computes a meaningful control action for a specific input condition of the variables. The term “composition” is often used for the inference to generate the output fuzzy control signal. There are a number of composition rules in the literature, but the most common one is the MAX-MIN (or SUP-MIN) composition, which is illustrated in Figure 11.14 for the two stated rules. Note that the output MF of each rule is given by a MIN (minimum) operator, whereas the combined fuzzy output is given by a MAX (maximum) operator.

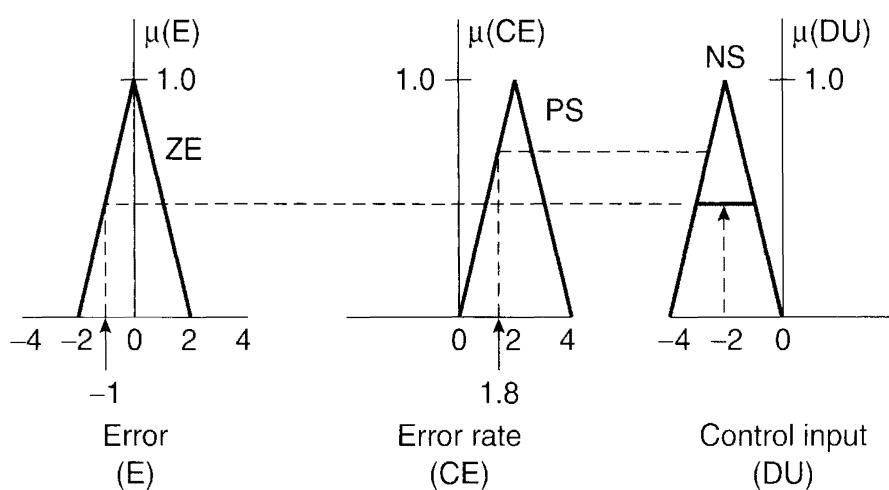


Figure 11.13 Single-rule fuzzy speed control principle

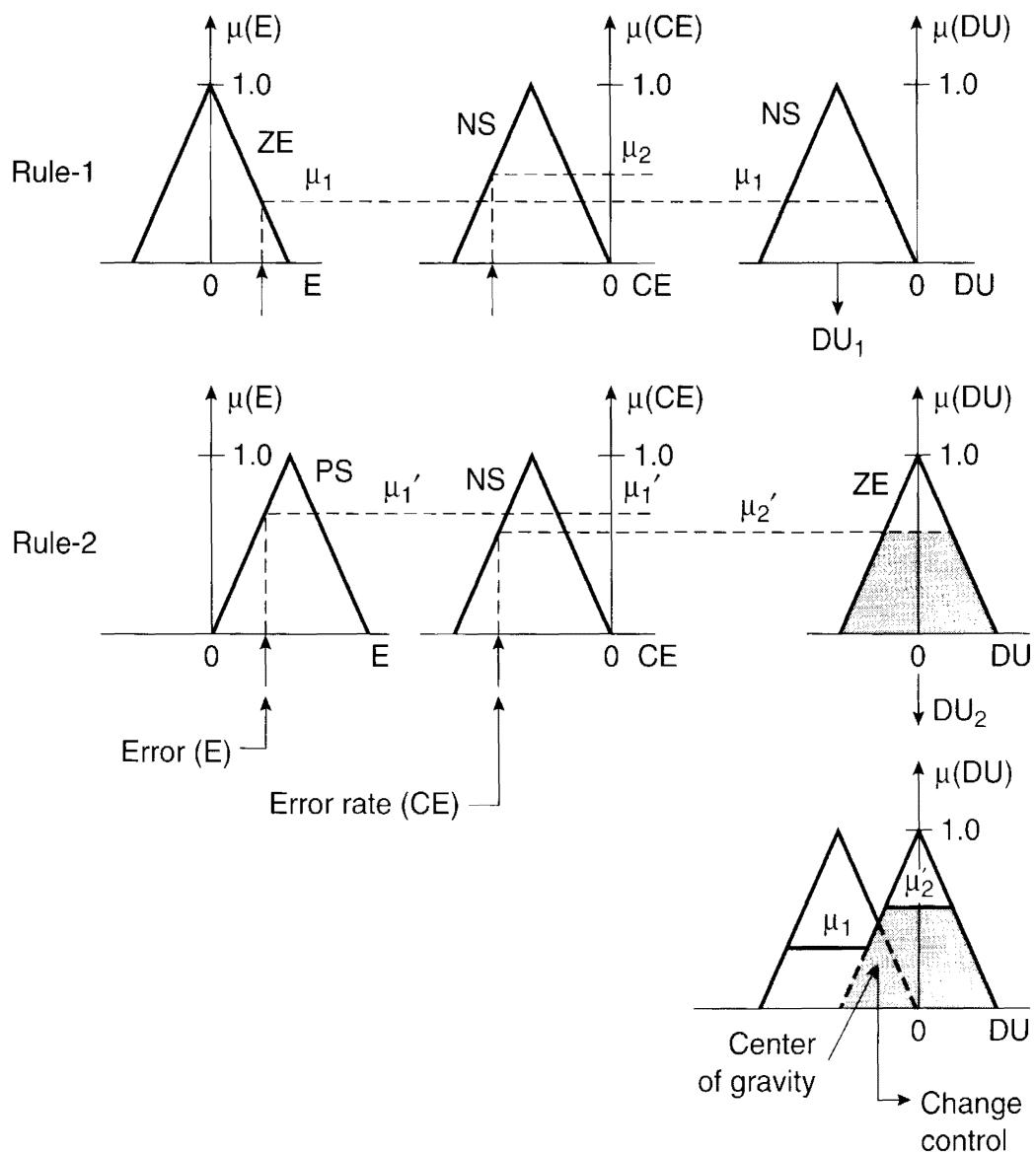


Figure 11.14 Two-rule fuzzy speed control principle

Since the fuzzy controller is basically an input/output static nonlinear mapping, we can write the controller action in the form

$$K_1 E + K_2 CE = DU \quad (11.27)$$

where K_1 and K_2 are nonlinear coefficients or gain factors. Including the summation process shown in Figure 11.12, we can write

$$\int DU = \int K_1 Edt + \int K_2 CEdt \quad (11.28)$$

or

$$U = K_1 \int Edt + K_2 E \quad (11.29)$$

which is nothing but a fuzzy P-I controller with nonlinear gain factors. Extending the same principle, we can write a fuzzy control algorithm for P and P-I-D control as follows:

Fuzzy P control example rule: IF $E = \text{positive small (PS)}$ THEN $U = \text{positive big (PB)}$

In other words, $KE = U$, where K is nonlinear gain

Fuzzy P-I-D control example rule: IF $E = \text{PS}$ AND $CE = \text{NS}$ AND $C^2E = \text{PS}$ THEN $DU = \text{ZE}$,

where C^2E is the derivative of CE . The control can be written in the form

$$K_1E + K_2CE + K_3C^2E = DU \quad (11.30)$$

and including the output integration

$$\int DU = \int K_1Edt + \int K_2CEDt + \int K_3C^2Edt \quad (11.31)$$

or

$$U = K_1 \int Edt + K_2 E + K_3 \frac{d}{dt}(E) \quad (11.32)$$

which is nothing but a *P-I-D* controller. The nonlinear adaptive gains in the fuzzy controller that are varied on-line give the power to the fuzzy controller to make the system response robust in the presence of parameter variation and load disturbance.

The general structure of a fuzzy feedback control system is shown in Figure 11.15. The loop error E and change in error CE signals are converted to the respective per unit signals e and ce by dividing by the respective scale factors, that is, $e = E/GE$ and $ce = CE/GC$. Similarly, the output plant control signal U is derived by multiplying the per unit output du by the scale factor GU , that is, $DU = du \cdot GU$, and then summed to generate the U signal.

The advantage of fuzzy control in terms of per unit variables is that the same control algorithm can be applied to all the plants of the same family. Besides, it becomes convenient to design the fuzzy controller. The scale factors can be constant or programmable; programmable scale factors can control the sensitivity of operation in different regions of control or the same

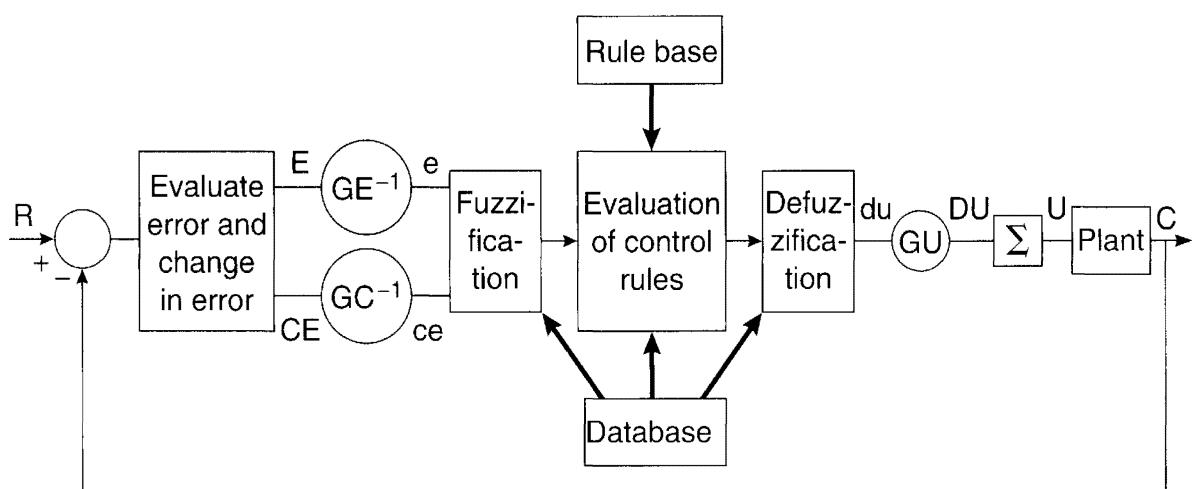


Figure 11.15 Structure of fuzzy control in feedback system

control strategy can be applied in similar response loops. The processes of fuzzification, evaluation of control rules from the rule base and database (defined as the knowledge base), and defuzzification have already been discussed.

11.4.4 Control Implementation

There are essentially two methods for implementation of fuzzy control. The first involves rigorous mathematical computation for fuzzification, evaluation of control rules, and defuzzification in real time. This is the generally accepted method and will be illustrated in our example applications described later. An efficient C program is normally developed with the help of a FL tool, such as the Fuzzy Logic Toolbox (described later) in the MATLAB (Math Works, Inc.) environment. The program is compiled and the object program is loaded in a DSP (digital signal processor) for execution. Commercial (digital or analog) ASIC chips are also available for implementation. The second method is the look-up table method, where all the input/output static mapping computation (fuzzification, evaluation of control rules, and defuzzification) is done ahead of time and stored in the form of a large look-up table for real-time implementation. Instead of one look-up table, there may be hierarchical tables (coarse, medium, and fine) [6]. Look-up tables require large amounts of memory for precision control, but their execution may be fast. A neural network (described in Chapter 12) can also be trained to emulate a fuzzy controller.

11.5 GENERAL DESIGN METHODOLOGY

The discussions given above will give guidance to readers for the formulation of an FL application for a certain problem and its implementation. A number of example applications in power electronics, which are given later, will make the concepts more clear. In summary, the general design procedure for fuzzy control can be given as follows:

1. First, analyze whether the problem has sufficient elements to warrant a FL application; otherwise, apply a conventional method. For example, in a linear feedback system where the mathematical model is known and there is no parameter variation or load disturbance problem, FL has little advantage. An ES or neural network can also be used, if necessary.
2. Get all the information from the operator of the plant to be controlled. Get information about the design and operational characteristics of the plant from the plant designer, if possible.
3. If a plant model is available, develop a simulation program with conventional control and study the performance characteristics.
4. Identify the functional elements where FL can be applied.
5. Identify the input and output variables of each fuzzy system.
6. Define the universe of discourse of the variables and convert to corresponding per unit variables as necessary.
7. Formulate the fuzzy sets and select the corresponding MF shape of each. For a sensitive variable, more fuzzy sets are needed. If a variable requires more precision near steady state, use more crowding of MFs near the origin.

8. Formulate the rule table. (This step and the previous one are the main design steps, which require intuition and experience about the process.)
9. If the mathematical plant model is available, simulate the system with the fuzzy controller. Iterate the fuzzy sets and rule table until the performance is optimized. For a plant without a model, the fuzzy system must be designed conservatively and then fine-tuned by the test results on the operating plant.
10. Implement the control in real time and further iterate to improve performance.

11.6 APPLICATIONS

Fuzzy logic has been widely applied in power electronic systems. Applications include speed control of dc and ac drives, feedback control of converter, off-line P-I and P-I-D tuning, nonlinearity compensation, on-line and off-line diagnostics, modeling, parameter estimation, performance optimization of drive systems based on on-line search, estimation for distorted waves, and so on. In this section, a few example applications from the literature will be reviewed.

11.6.1 Induction Motor Speed Control

Consider the fuzzy speed control system shown in Figure 11.12 [5][14], where the input signals are E and CE and the output signal is DU , as explained before. Figure 11.16 shows the fuzzy sets and corresponding triangular MF description of each signal.

The fuzzy sets are defined (the linguistic definition is immaterial) as follows:

$Z = \text{Zero}$	$PS = \text{Positive Small}$	$PM = \text{Positive Medium}$
$PB = \text{Positive Big}$	$NS = \text{Negative Small}$	$NM = \text{Negative Medium}$
$NB = \text{Negative Big}$	$PVS = \text{Positive Very Small}$	$NVS = \text{Negative Very Small}$

The universe of discourse of all the variables, covering the whole region, is expressed in per unit values. All the MFs are asymmetrical because near the origin (steady state), the signals require more precision. There are seven MFs for $e(pu)$ and $ce(pu)$ signals, whereas there are nine MFs for the output. All the MFs are symmetrical for positive and negative values of the variables. Table 11.1 shows the corresponding rule table for the speed controller. The top row and left column of the matrix indicate the fuzzy sets of the variables e and ce , respectively, and the MFs of the output variable $du(pu)$ are shown in the body of the matrix. There may be $7 \times 7 = 49$ possible rules in the matrix, where a typical rule reads as

IF $e(pu) = PS$ AND $ce(pu) = NM$ THEN $du(pu) = NS$

Some blocks in the rule table may remain vacant, giving less number of rules.

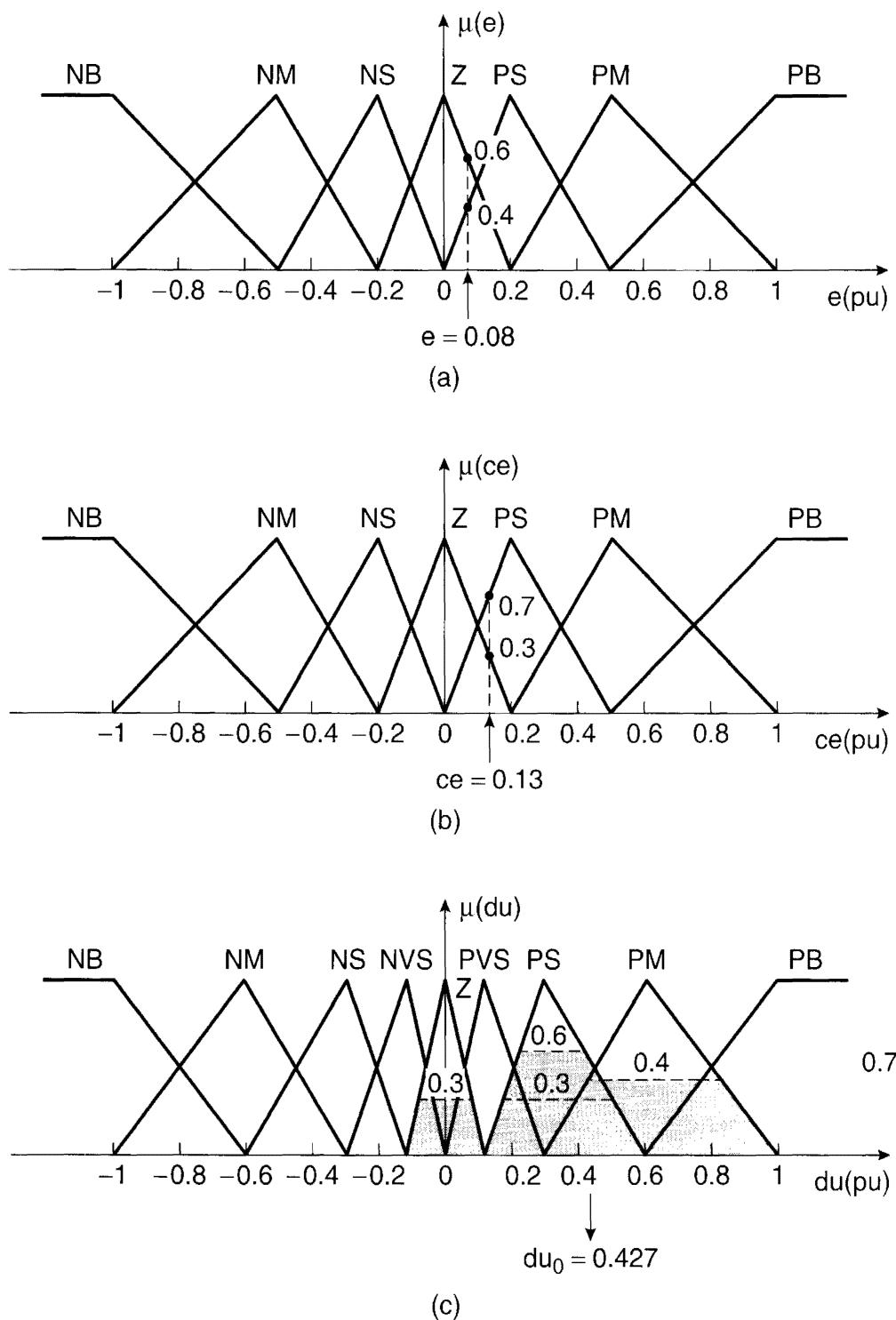


Figure 11.16 Membership functions for fuzzy speed control (a) Error ($e(\text{pu})$), (b) Change in error ($ce(\text{pu})$), (c) Change in output control ($du(\text{pu})$)

Table 11.1 Rule Matrix for Fuzzy Speed Control

e(pu)\ce(pu)	NB	NM	NS	Z	PS	PM	PB
NB	NVB	NVB	NVB	NB	NM	NS	Z
NM	NVB	NVB	NB	NM	NS	Z	PS
NS	NVB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PVB
PM	NS	Z	PS	PM	PB	PVB	PVB
PB	Z	PS	PM	PB	PVB	PVB	PVB

The general considerations in the design of the controller are:

1. If both $e(pu)$ and $ce(pu)$ are zero, then maintain the present control setting $du(pu) = 0$.
2. If $e(pu)$ is not zero but is approaching this value at a satisfactory rate, then maintain the present control setting.
3. If $e(pu)$ is growing, then change the control signal $du(pu)$ depending on the magnitude and sign of $e(pu)$ and $ce(pu)$ to force $e(pu)$ towards zero.

As mentioned above, the rule matrix and MF description of the variables are based on the knowledge of the system and their fine-tuning may be time-consuming for optimal performance. For a simulation-based system design, controller tuning with the help of the MATLAB Fuzzy Logic Toolbox (for example) may be reasonably fast. Recently, neural network and genetic algorithm techniques have been proposed for tuning MFs. Adaptive Neuro-Fuzzy Inference System (ANFIS) will be discussed in Chapter 12.

The algorithm for fuzzy speed control in detail can be summarized as follows. A numerical example is included in each step for clarity (refer to Figures 11.15 and 11.16)

1. Sample speeds ω_r^* and ω_r .
2. Compute error E , change in error CE , and their per unit values as follows:

$$E(k) = \omega_r^* - \omega_r$$

$$CE(k) = E(k) - E(k-1)$$

$$e(pu) = E(k)/GE$$

$$ce(pu) = CE(k)/GC$$

$$[E = 0.8, CE = 1.3, GE = 10, GC = 10, e(pu) = 0.8/10 = 0.08, ce(pu) = 1.3/10 = 0.13]$$

3. Identify the interval index I and J for $e(pu)$ and $ce(pu)$, respectively

$$[I = 1, J = 1]$$

4. Calculate the degree of membership of $e(pu)$ and $ce(pu)$ for the relevant fuzzy sets

$$[\mu_Z(e) = 0.6, \mu_{PS}(e) = 0.4, \mu_Z(ce) = 0.3, \mu_{PS}(ce) = 0.7]$$

5. Identify the four valid rules from Table 11.1 (stored as a look-up table) for Z and PS values of $e(pu)$ and $ce(pu)$. These are:

$$R1: \text{IF } e(pu) = Z \text{ AND } ce(pu) = Z \text{ THEN } du(pu) = Z$$

$$R2: \text{IF } e(pu) = Z \text{ AND } ce(pu) = PS \text{ THEN } du(pu) = PS$$

$$R3: \text{IF } e(pu) = PS \text{ AND } ce(pu) = Z \text{ THEN } du(pu) = PS$$

$$R4: \text{IF } e(pu) = PS \text{ AND } ce(pu) = PS \text{ THEN } du(pu) = PM$$

Calculate the DOF of each rule using the AND or min operator

$$[\text{DOF}_1 = \min \{\mu_Z(e), \mu_Z(ce)\} = \min \{0.6, 0.3\} = 0.3$$

$$\text{DOF}_2 = \min \{\mu_Z(e), \mu_{PS}(ce)\} = \min \{0.6, 0.7\} = 0.6$$

$$\text{DOF}_3 = \min \{\mu_{PS}(e), \mu_Z(ce)\} = \min \{0.4, 0.3\} = 0.3$$

$$\text{DOF}_4 = \min \{\mu_{PS}(e), \mu_{PS}(ce)\} = \min \{0.4, 0.7\} = 0.4]$$

6. Retrieve the amount of correction $du(pu)_i$ ($i = 1, 2, 3, 4$) corresponding to each rule in Table 11.1

$$[du(pu)_1 = 0 \text{ for } Z \text{ corresponding to } \text{DOF}_1 = 0.3$$

$$du(pu)_2 = 0.35 \text{ for } PS \text{ corresponding to } \text{DOF}_2 = 0.6$$

$$du(pu)_3 = 0.35 \text{ for } PS \text{ corresponding to } \text{DOF}_3 = 0.3$$

$$du(pu)_4 = 0.6 \text{ for } PM \text{ corresponding to } \text{DOF}_4 = 0.4]$$

7. Calculate the crisp output $du(pu)_0$ by the height defuzzification method

$$[du(pu)_0 = \frac{0.3 \times 0 + 0.6 \times 0.35 + 0.3 \times 0.35 + 0.4 \times 0.6}{0 + 0.35 + 0.35 + 0.6} = 0.427]$$

Figure 11.17 shows the typical speed loop response with stepped load torque T_L , and Figure 11.18 shows a similar response with four times the rotor inertia. The performances are superior to conventional P-I control. The robustness in the response is evident from the results.

11.6.2 Flux Programming Efficiency Improvement of Induction Motor Drive

It was discussed in Chapter 8 that in variable-frequency drives, the machines are normally operated at the rated flux to give maximum developed torque per ampere and optimum transient response. However, at light loads, this causes excessive core loss, impairing the efficiency of the drive. To improve the efficiency, flux programming at light loads by the open loop or close loop method was discussed. The efficiency optimization control based on an on-line search of optimum flux is very attractive. Figure 11.19 explains the on-line search method for an indirect vector-controlled induction motor drive (see Figure 11.20). Assume that the machine operates

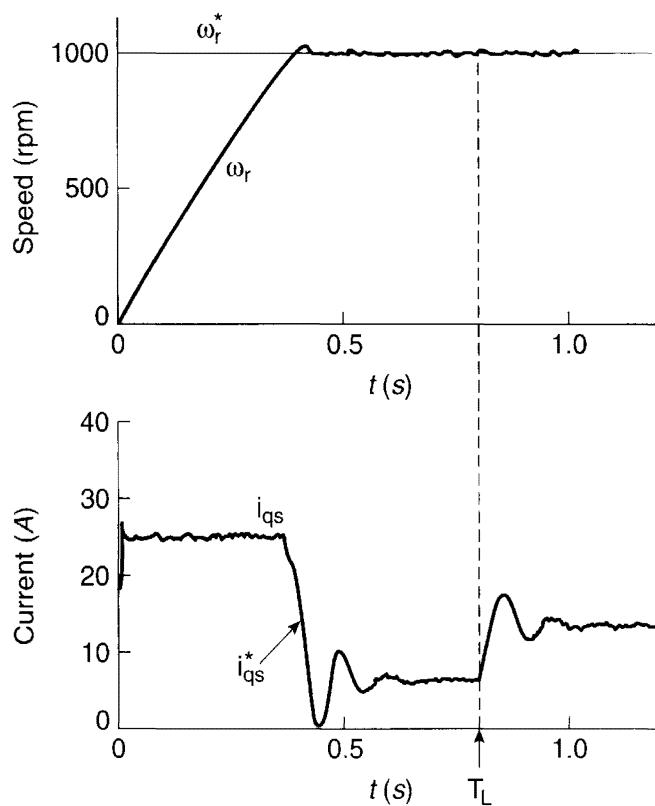


Figure 11.17 Fuzzy control response with steps of speed command and load torque with nominal inertia (J)

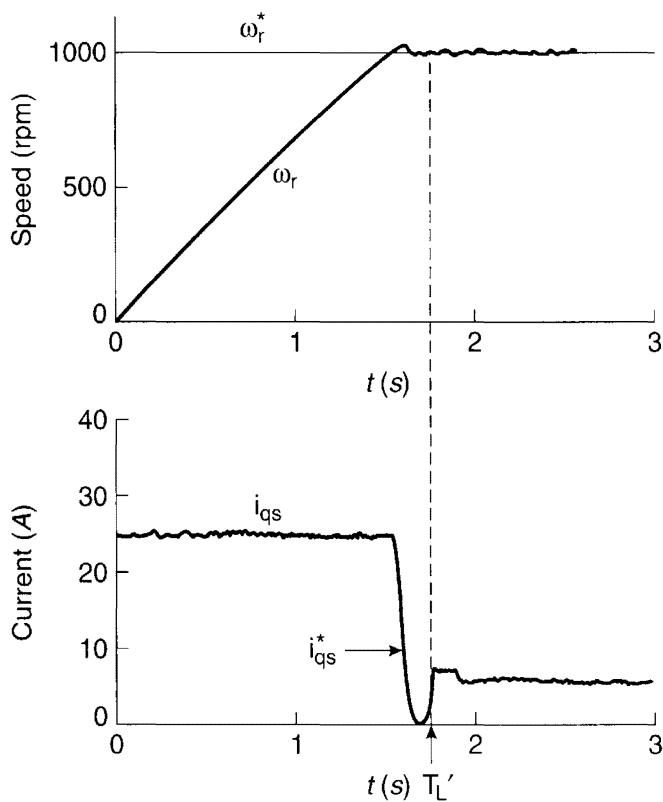


Figure 11.18 Fuzzy control response with steps of speed command and load torque with four times the nominal inertia ($4J$)

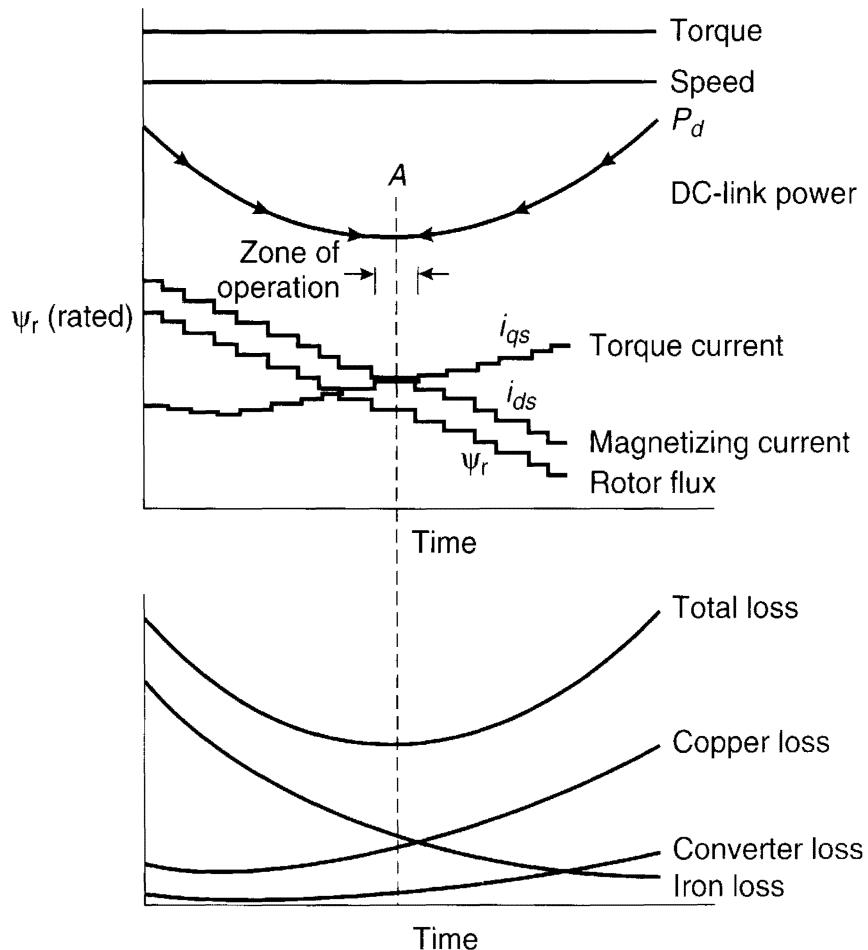


Figure 11.19 On-line search method of flux programming efficiency improvement control

initially at the rated flux in steady state with low load torque at a certain speed, as indicated. The rotor flux ψ_r is decremented in steps by reducing the magnetizing component i_{ds} of the stator current. This results in an increase of the torque component of current i_{qs} (due to speed control loop) so that the developed torque remains the same for steady-state operation. As the core loss decreases with a decrease of flux, the copper loss increases, but the system loss (machine and converter loss) decreases, improving the overall efficiency. This is reflected in the decrease of the dc link power P_d , as shown for the same output power. The search is continued until the system settles at the minimum input power (i.e., maximum efficiency) point, A. Any search attempt beyond point A adversely affects efficiency and forces the search direction such that operation always settles at point A. This type of algorithm has the advantages of the control not requiring knowledge of machine parameters, it has complete insensitivity to parameter variation, and it is universally applicable to any arbitrary machine.

Figure 11.20 shows the block diagram of vector control incorporating the FL-based efficiency optimizer [7]. The total control implemented by a DSP is indicated by the dashed outline. Although other methods of control can be used, the fuzzy control has the advantages of being able to handle noisy and inaccurate input signals, and the step size of the i_{ds} decrement is adaptive (initially large–then small), so that fast convergence in the control is attained. In Figure 11.20, the speed control loop generates the torque component of current i_{qs}^* , as shown. The vector rotator receives

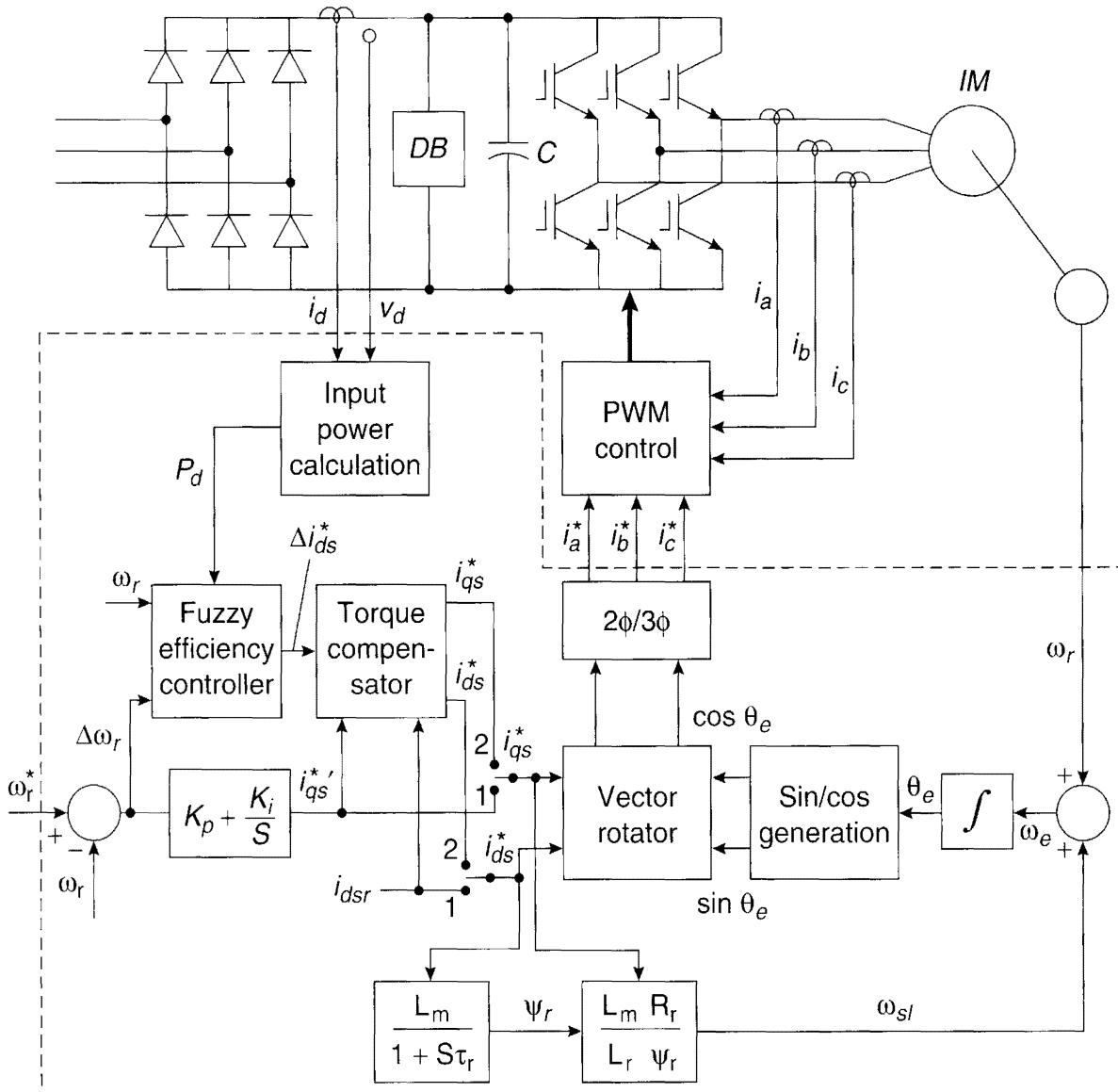


Figure 11.20 Indirect vector-controlled drive with flux programming efficiency optimizer

the torque and magnetizing current commands i_{qs}^* and i_{ds}^* , respectively, from two positions of a switch: (1) the transient position, where the magnetizing current is established to the rated value i_{dsr} and the torque current i_{qs}^{**} ; and (2) the steady-state position, where the magnetizing and torque currents i_{ds}^* , i_{qs}^* are generated by the fuzzy efficiency controller and feedforward torque compensator, which will be explained later. The fuzzy controller becomes effective only at steady-state condition, that is, when the speed loop error $\Delta\omega_r$ approaches zero. Figure 11.21 shows the fuzzy efficiency controller block diagram. The dc link power $P_d(k)$ is sampled and compared with the previous value to determine the decrement (or increment) $\Delta P_d(k)$. In addition, the last magnetization current segment's ($\Delta i_{ds}(pu)^*(k-1)$) polarity is reviewed. Based on these input signals, the decrement step of $\Delta i_{ds}(pu)$ is generated from the fuzzy inference system. The normalizing scale factors P_b and I_b are programmable and are given by the expressions

$$P_b = A\omega_r + B \quad (11.33)$$

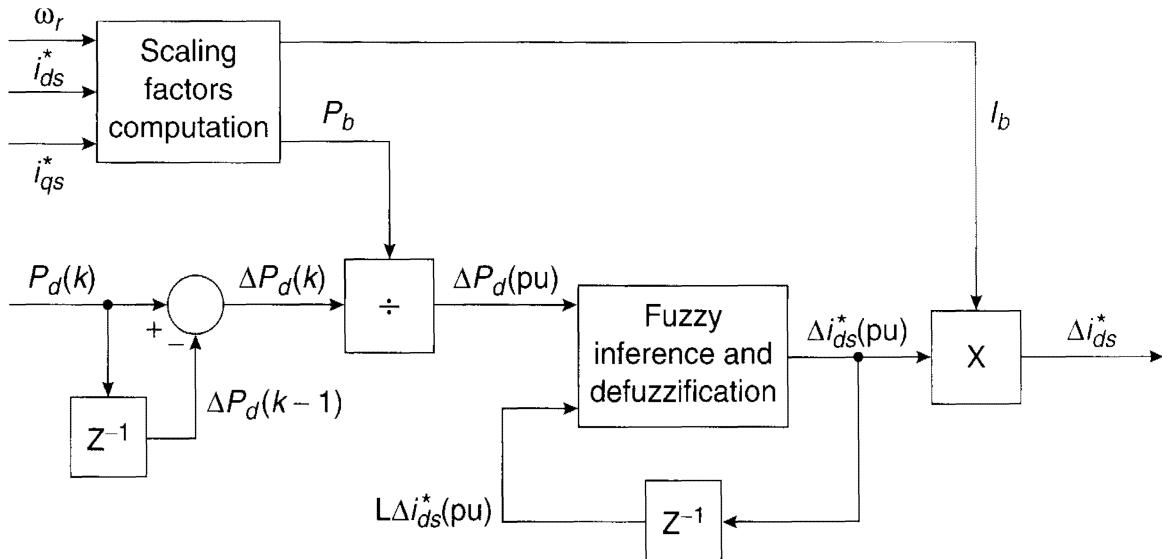


Figure 11.21 Efficiency optimizer control block diagram

and

$$I_b = C_1 \omega_r - C_2 \hat{T}_e + C_3 \quad (11.34)$$

where A , B , C_1 , C_2 , and C_3 are constants (determined by simulation study) and \hat{T}_e is the estimated torque given by

$$\hat{T}_e = K_t' i_d^* i_q^* \quad (11.35)$$

The programmable scale factors make these variables insensitive to the operating point on the torque-speed plane.

Variables $\Delta P_d(pu)$ and $\Delta i_d^*(pu)$ are each described by seven asymmetric triangular MFs, whereas $L\Delta i_d^*(pu)$ has only two (positive and negative) MFs, as shown in Figure 11.22. Table 11.2 shows the corresponding rule table where a typical rule reads as

IF $\Delta P_d(pu)$ = Negative Small (NS) AND $L\Delta i_d^*(pu)$ = Negative (N)
 THEN $\Delta i_d^*(pu)$ = Negative Small (NS)

The basic idea is that if the last control action indicates a decrease of dc link power, then proceed searching in the same direction, and the control magnitude should be somewhat proportional to the measured dc link power change. When the control action results in an increase of P_d ($\Delta P_d > 0$), the search direction is reversed. At steady state, the operation oscillates around point A (Figure 11.19) with very small step size.

11.6.2.1 Pulsating Torque Compensation

Figure 11.23 explains the principle of feedforward pulsating torque compensation. This compensator functions to compensate the loss of torque due to the decrement of i_{ds} by injecting an equivalent Δi_{qs}^* so that the developed torque remains the same. Otherwise, the slow compensation of i_{qs} by the speed loop creates a large pulsating torque at a low sampling frequency

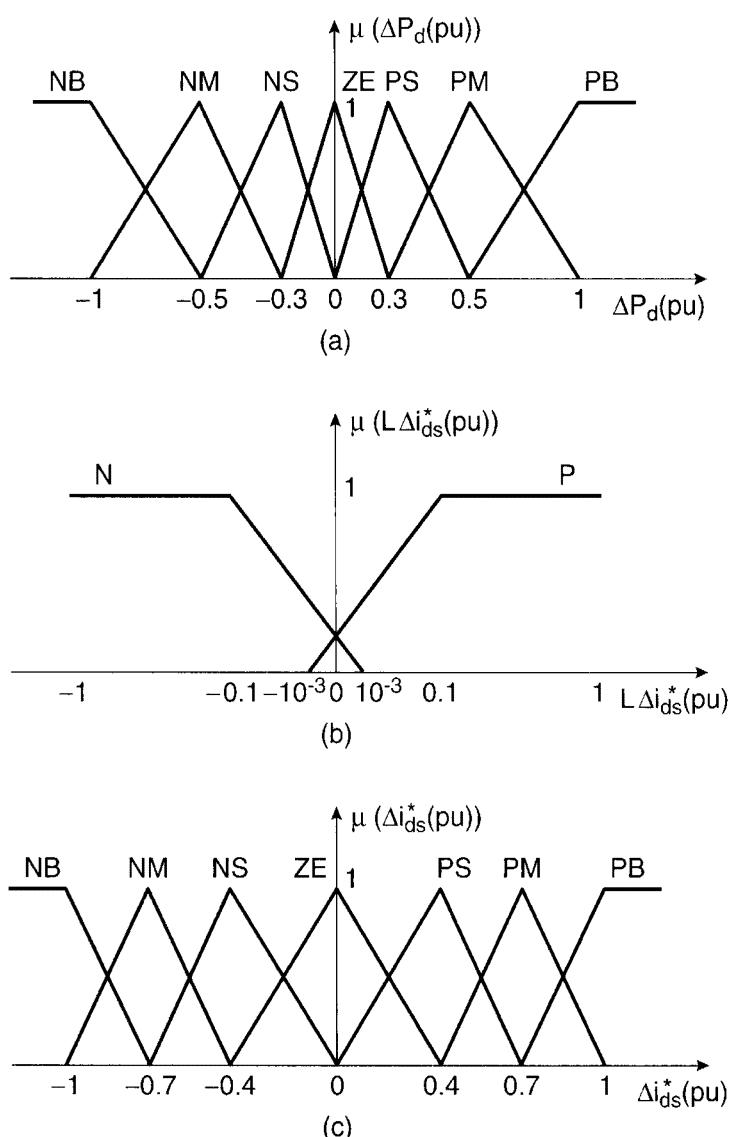


Figure 11.22 Membership functions of fuzzy variables

Table 11.2 Rule Matrix for Efficiency Improvement

$\Delta P_d(\text{pu}) \setminus L\Delta i_{ds}(\text{pu})$	PB	PM	NM
PB	PM	NM	
PM	PS	NS	
PS	PS	NS	
ZE	ZE	ZE	
NS	NS	PS	
NM	NM	PM	
NB	NB	PB	

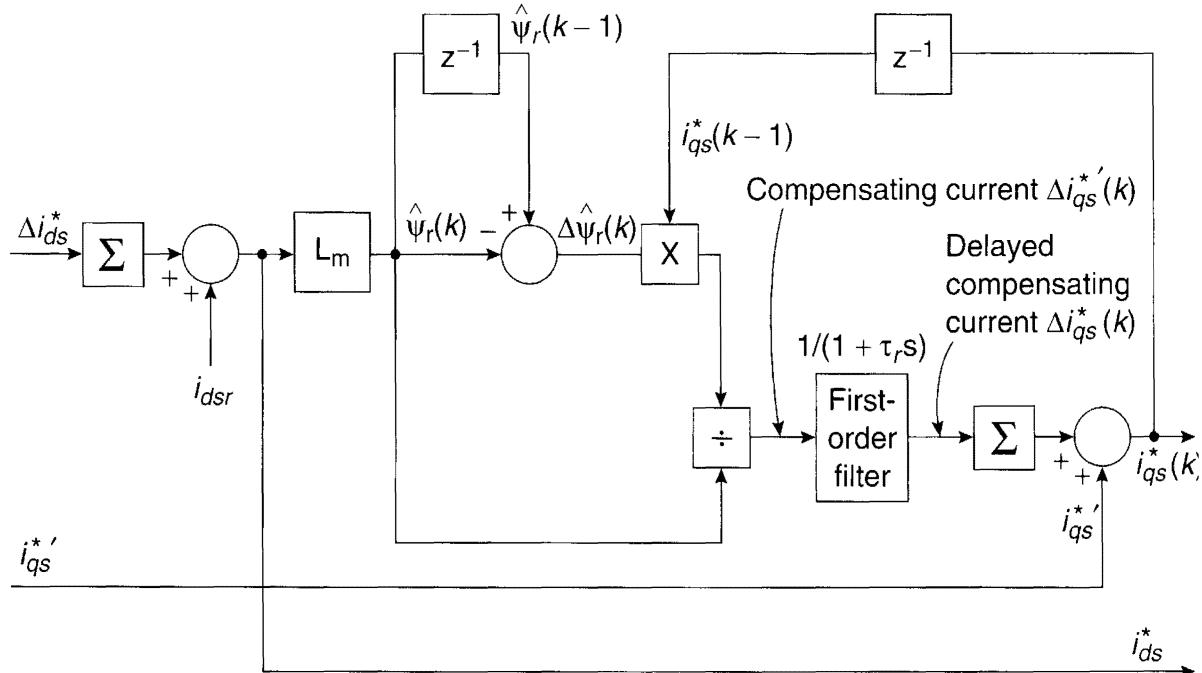


Figure 11.23 Feedforward pulsating torque compensation block diagram

(sometimes causing resonance effect), which may be harmful for the drive. The compensating current $\Delta i_{qs}^{*'}(k)$ can be calculated as

$$\Delta i_{qs}^{*'}(k) = \frac{\hat{\psi}_r(k-1) - \hat{\psi}_r(k)}{\hat{\psi}_r(k)} \cdot i_{qs}^*(k-1) \quad (11.36)$$

where $\hat{\psi}_r(k)$ is the rotor flux vector magnitude.

Since the response of $\Delta i_{qs}^{*'}(k)$ is practically instantaneous, whereas Δi_{ds}^* responds with the delay of rotor time constant τ_r for correct torque matching, the $\Delta i_{qs}^{*'}(k)$ signal must be delayed by the same τ_r in the first-order filter shown in the figure. Figure 11.24(a) shows the efficiency optimizer performance, and (b) indicates the response of the torque compensator. If the $\Delta\omega_r$ signal appears in Figure 11.20 due to a change of speed command or a load torque change, the fuzzy control is abandoned and full flux is established by transferring both switches to position 1. The control thus transitions from efficiency optimization mode to transient response optimization mode.

11.6.3 Wind Generation System

Wind electrical power generation systems have recently attracted a lot of attention because they are cost-effective, environmentally clean, and safe renewable power sources compared to fossil fuels and nuclear power generation. Compared to traditional variable-pitch, fixed-speed wind turbines (horizontal or vertical axis), variable-speed wind turbine (VSWT) systems with the help of power electronics have found more acceptance recently. Although power electronics are expensive, the larger energy capture of VSWT systems makes the life-cycle cost of these systems lower. In this section, we will describe a FL-based efficiency optimization and perfor-

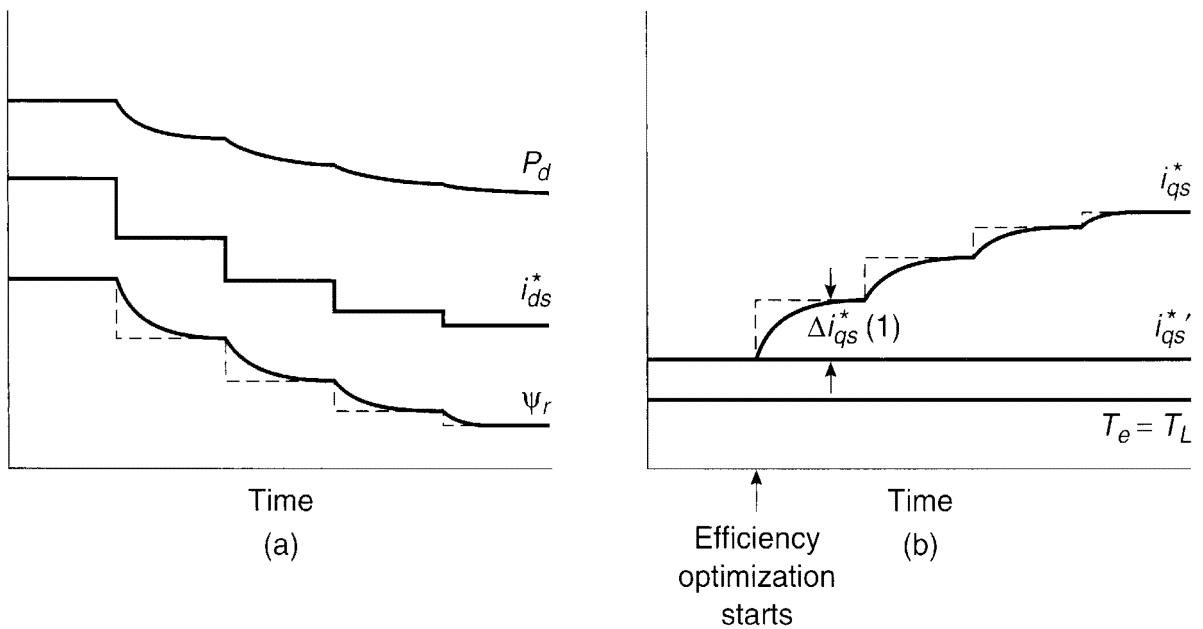


Figure 11.24 (a) Performance of fuzzy efficiency controller, (b) Response of feedforward torque compensator

mance enhancement control of a wind generation system using a cage-type induction generator. A description of the power system is essential before explaining the fuzzy control.

11.6.3.1 Wind Turbine Characteristics

Either horizontal or vertical axis wind turbines can be used in wind generation systems. The vertical Darrieus (egg beater) type, which is under consideration here, has the advantages that it can be installed on the ground, accepting wind from any direction. However, the disadvantages are that the turbine is not self-starting and there is a large pulsating torque, which depends on wind velocity and turbine speed. Figure 11.25 shows the torque-speed curves of a wind turbine at different wind velocities. At a particular wind velocity (V_{W1}), if the turbine speed ω_r decreases from ω_{r1} , the developed torque increases, reaches the maximum value at point B, and then decreases at lower ω_r . Superimposed on the family of curves is a set of constant power lines (dotted), indicating the points of maximum power output for each wind speed. This means that at a particular wind velocity, the turbine speed is to be varied to get maximum power delivery (highest aerodynamic efficiency), and this point (for example, C) deviates from the maximum torque point, as indicated in the figure. Since the torque-speed characteristics of a wind generation system are analogous to those of a motor-blower system (except the turbine runs in the reverse direction), the torque follows the square law characteristics $T_e = K \omega_r^2$, and the output power follows the cube law $P_0 = K \omega_r^3$, as indicated in the figure.

11.6.3.2 System Description

Figure 11.26 shows the block diagram of the wind generation system incorporating all the control elements [8]. The turbine at the left (a vertical type) is coupled to the cage-type induction generator through a speed-up gear ratio (not shown). The variable-frequency, variable-voltage power generated by the machine is rectified to dc by a PWM voltage-fed rectifier that also

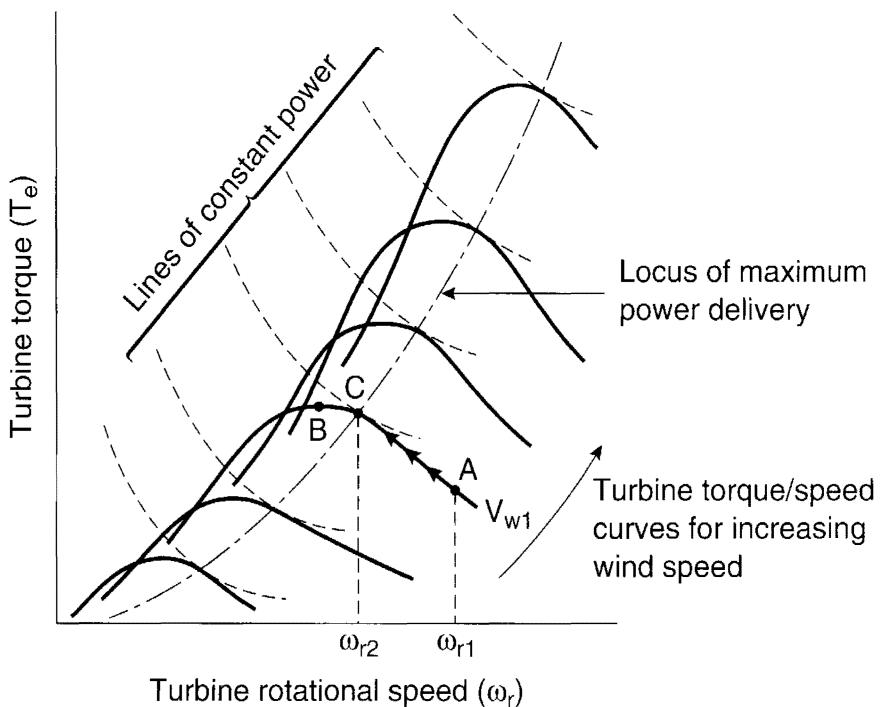


Figure 11.25 Torque-speed curves of fixed-pitch wind turbine at different wind velocities

supplies the excitation current (lagging) to the machine. The dc link power is inverted to 220 V, 60 Hz ac through a PWM inverter and fed to the utility grid. The line current is sinusoidal at unity power factor, as indicated. The generator speed ω_r is controlled by an indirect vector control with a torque control for stiffness and a synchronous current control in the inner loops. The machine flux ψ_r is controlled by open loop control of the excitation current i_{ds} , but in normal condition, the flux is set to the rated value for fast transient response. The line-side converter is also vector-controlled, using direct vector control and synchronous current control in the inner loops. The output power P_0 is controlled to control the dc link voltage V_d as shown in the figure. Because an increase in P_0 causes a decrease in V_d , the voltage loop error polarity has been inverted. The tight regulation of V_d within a small tolerance band requires feedforward power injection in the power control loop, as shown. The insertion of line filter inductance L_s creates some coupling effect, which is eliminated by a decoupler in the synchronous current control loops (not shown). The power can be controlled to flow easily in either direction. The vertical turbine is started with a motoring torque. As the speed develops, the machine goes into generating mode. The machine is shut down by regenerative braking.

11.6.3.3 Fuzzy Control

The system uses three fuzzy controllers whose functions can be summarized as follows:

FLC-1: On-line search of generator speed to maximize output power

FLC-2: On-line search of machine excitation current to optimize machine efficiency at light load

FLC-3: Robust control of generator speed

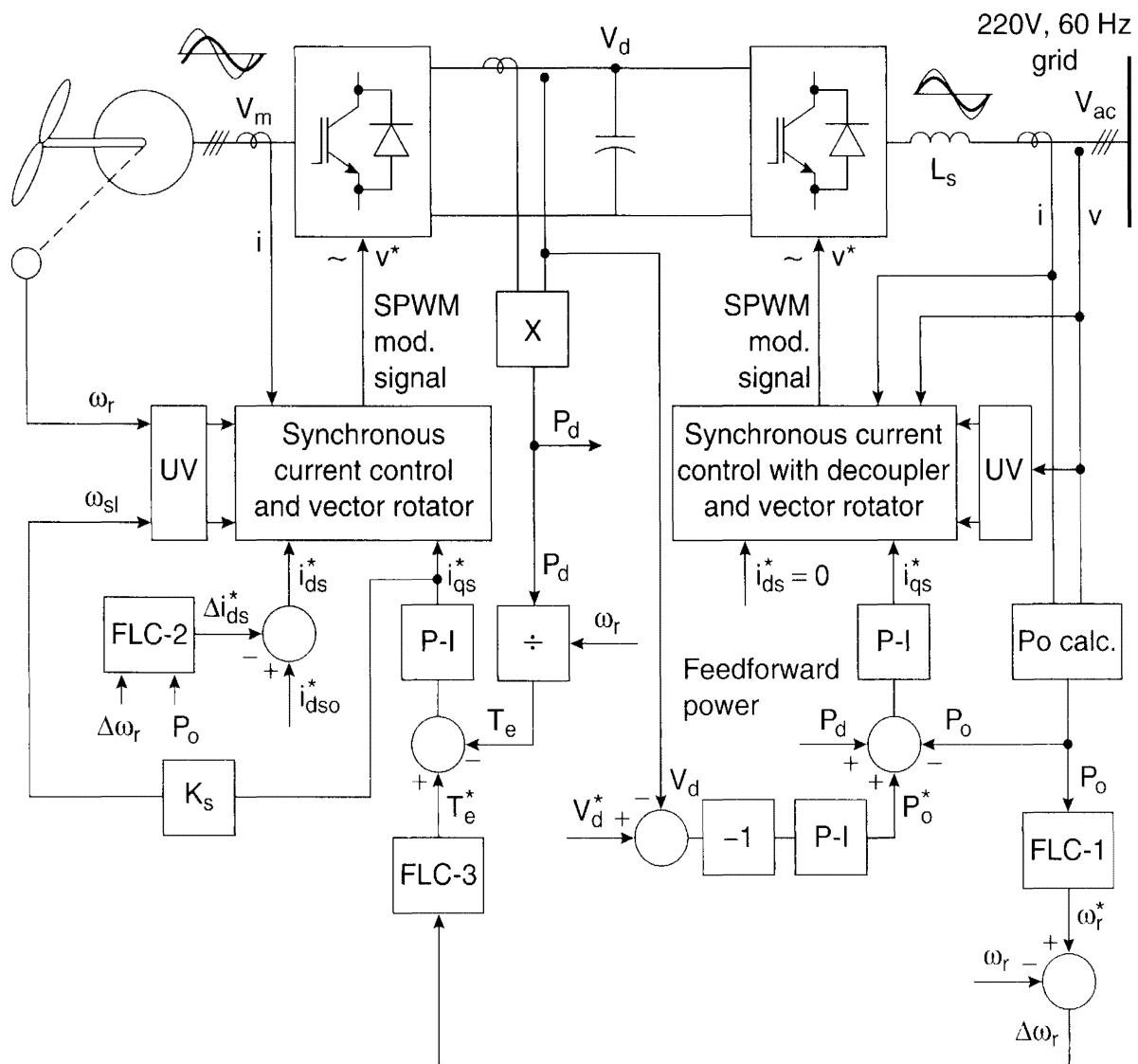


Figure 11.26 FL-based block diagram of wind generation system

Controllers FLC-2 and FLC-3 are essentially the same as described before, and will not be discussed further. With the cubic law power output, as mentioned before, the machine is at light load at reduced speed; therefore, flux programming control by an on-line search of the excitation current will improve its efficiency. The vertical turbine generates oscillatory torque which tends to make the machine speed jittery. Besides, the vortex in the wind velocity tends to disturb the speed. The fuzzy P-I control by FLC-3 makes the speed control robust against these disturbances.

Generator Speed Tracking Control (FLC-1) – The fuzzy controller FLC-1 maximizes the aerodynamic efficiency of the turbine at any wind velocity to extract maximum power output from the wind turbine. Since power is the product of torque and speed and turbine power equals line output power (neglecting losses in the system), the torque-speed curves of Figure 11.25 can be translated into line power (P_0) – generator speed (ω_r) curves, as shown approximately in Figure 11.27. For a particular wind velocity, the function of FLC-1 is to search the generator speed until the system settles down at the maximum output power condition. The figure also shows the

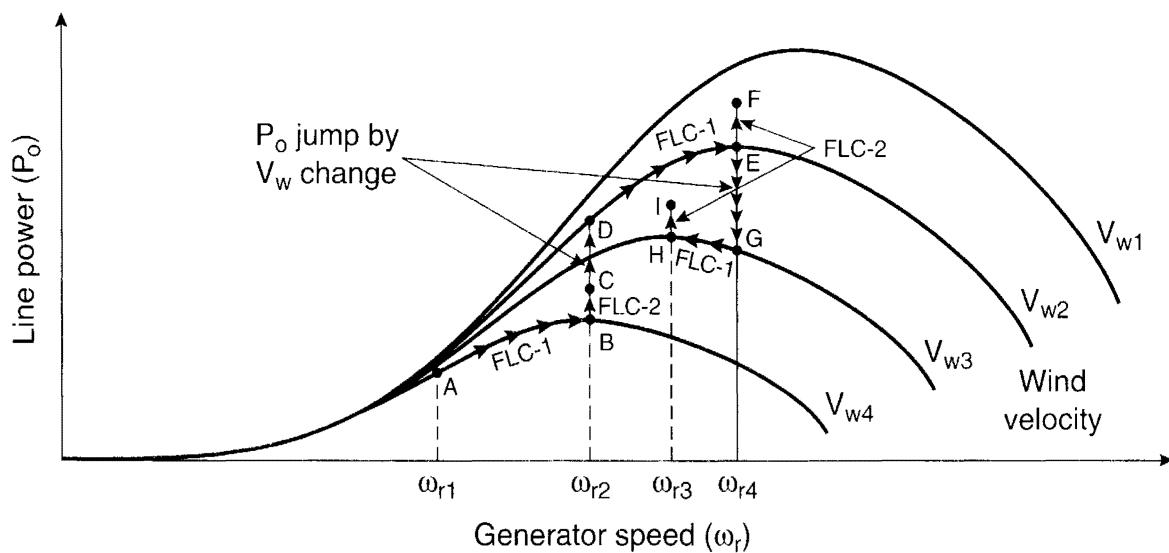


Figure 11.27 Fuzzy control of FLC-1 and FLC-2 showing increase of line power

effect of FLC-2 control. For example, at a wind velocity of V_{W4} in Figure 11.27, the power output will be at A if the generator speed is ω_{r1} . FLC-1 will alter the speed in steps on the basis of an on-line search until it reaches speed ω_{r2} , where the output power is maximum at B. At the end of FLC-1 control, FLC-2 control becomes effective and the improvement in generator efficiency increases the power output and brings the operating point to C. If the wind velocity increases to V_{W2} , the output power will jump to point D, and FLC-1 will bring the operating point to E by searching when the speed to ω_{r4} is attained. Then, FLC-2 control will bring the operating point to F. The profile for a decrease of wind velocity to V_{W3} is also indicated in the figure.

Figure 11.28 shows the block diagram of FLC-1. By incrementing (or decrementating) speed ω_r^* , the corresponding increment (or decrement) of output power P_0 is estimated. If ΔP_0 is positive with the last positive $\Delta\omega_r^*$ ($L\Delta\omega_r^*$), the search is continued in the same direction. If, on the other hand, $+ \Delta\omega_r^*$ causes $-\Delta P_0$, the direction of search is reversed. The speed oscillates by a small increment when it reaches the optimum condition. The normalized form of variables ΔP_0 (pu), $\Delta\omega_r^*$ (pu), and $L\Delta\omega_r^*$ (pu) are described by triangular MFs, as shown in Figure 11.29, and the control rules are given by the matrix in Table 11.3. In Figure 11.28, the output $\Delta\omega_r$ is added with some amount of the $L\Delta\omega_r^*$ signal to avoid local minima due to wind vortex and torque ripple. Again, the controller operates on a per unit basis so that the response is insensitive to system variables and the algorithm is universal to any similar system. The scale factors KPO and KWR , shown in Figure 11.28, are a function of generator speed so that the control becomes somewhat insensitive to speed variation. Although scale factors are normally generated by mathematical expression, in this case, they are generated by fuzzy computation indicating that KPO and KWR increase with speed. Figure 11.30 gives the MFs and rule matrix for scale factor computation. The whole fuzzy controller is designed from heuristic knowledge of the system.

Again, the advantages of fuzzy control are obvious. As mentioned previously, it will accept noisy and inaccurate signals and it provides an adaptively decreasing step size in the search that leads to fast convergence. Also, note that wind velocity information is not needed and the system parameter variation does not affect the search.

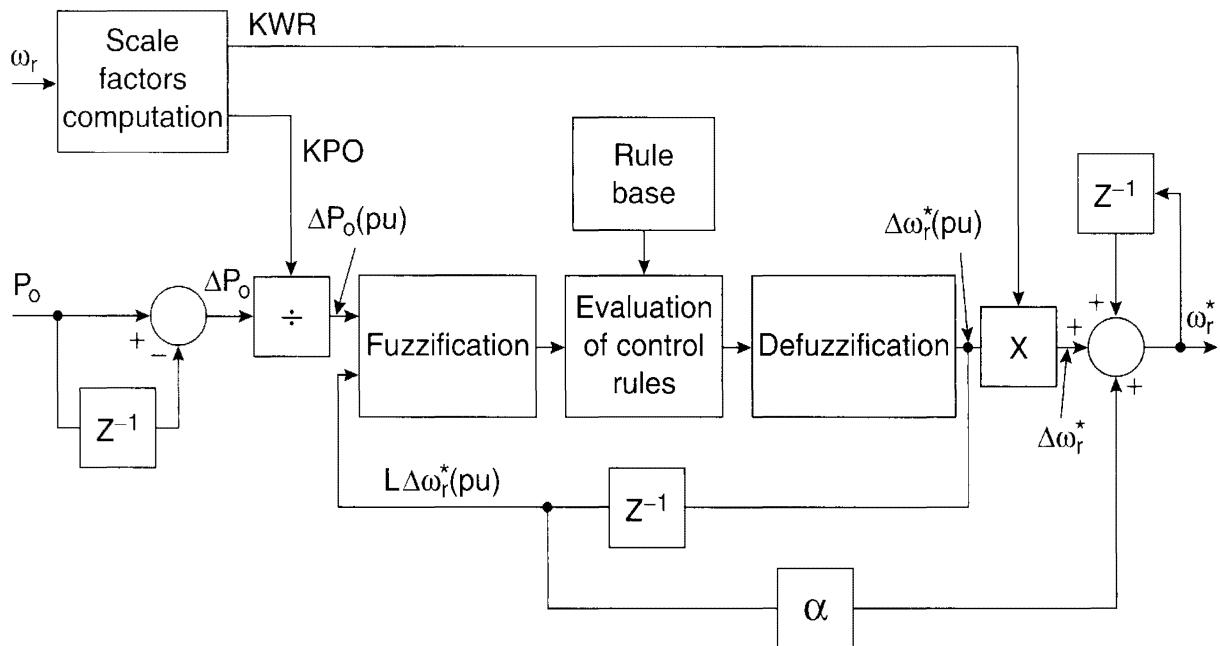


Figure 11.28 Block diagram of fuzzy controller FLC-1

Table 11.3 Rule Matrix for FLC-1

$\Delta P_o(\text{pu}) \backslash L\Delta\omega_r^*(\text{pu})$	P	ZE	N
PVB	PVB	PVB	NVB
PB	PB	PVB	NB
PM	PM	PB	NM
PS	PS	PM	NS
ZE	ZE	ZE	ZE
NS	NS	NM	PS
NM	NM	NB	PM
NB	NB	NVB	PB
NVB	NVB	NVB	PVB

Figure 11.31 shows the system efficiency improvement by FLC-1 and FLC-2 with variable wind velocity considering the generator operation at constant 940 rpm. As the wind velocity increases from a low value, the efficiency gain by FLC-2 decreases because of higher generator loading. The gain due to FLC-1 by tuning the generator speed to optimal value can be very high at low wind velocity although P_0 may be small. It decreases to zero near 0.7(pu) wind velocity

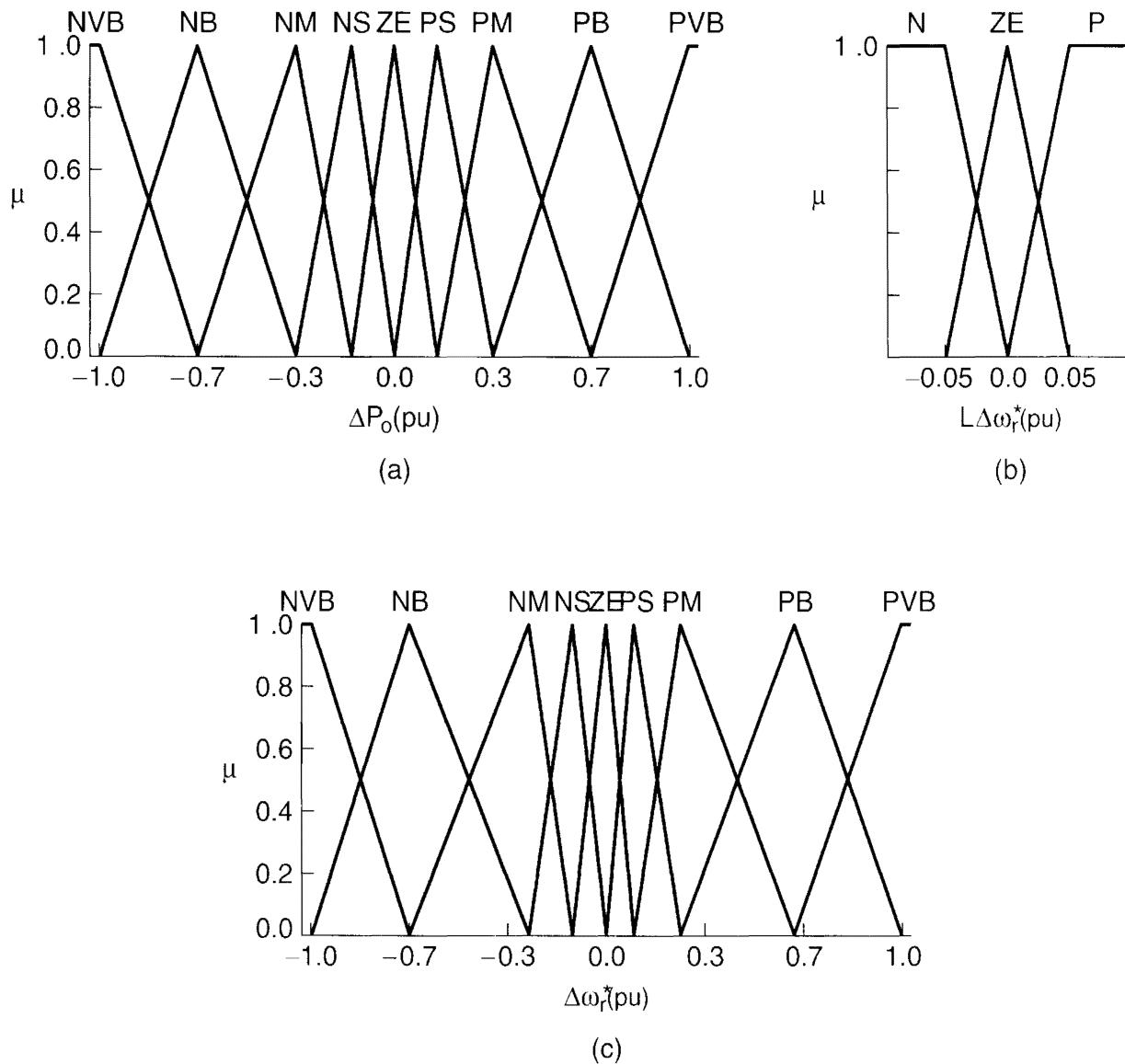
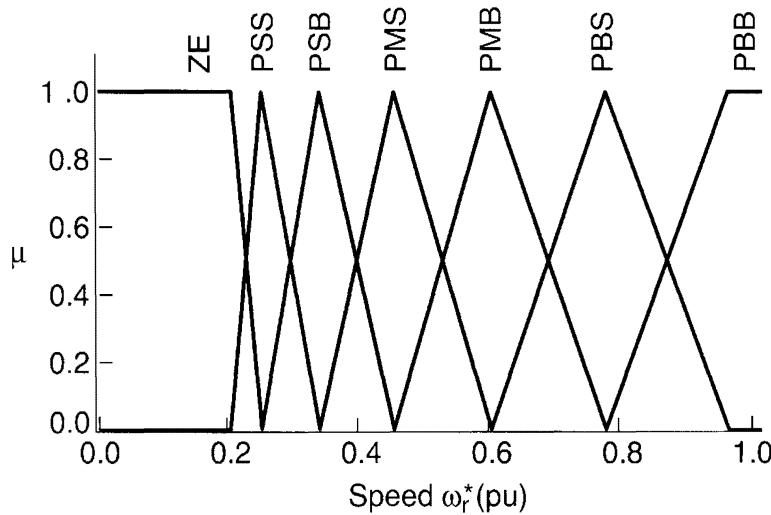


Figure 11.29 Membership functions of the variables in FLC-1

(where the generator speed of 940 rpm is optimal) and then increases again. This efficiency variation can be explained with the help of Figure 11.27.

11.6.4 Slip Gain Tuning of Indirect Vector Control

On-line slip gain (K_s) tuning of an indirect vector controlled induction motor drive with the help of MRAC was discussed in Chapter 8 (see Figure 8.38). FL principles can be applied in solving similar problems. Figure 11.32 shows the block diagram of an indirect vector-controlled drive where a fuzzy tuning controller has been incorporated. Figure 11.33 shows the details of fuzzy MRAC-based tuning controller [10]. The scheme depends on reference model computation of reactive power Q^* and the d -axis voltage v_{ds}^* at the machine terminal for the ideally tuned condition.



ω_r	KPO	KWR
PSS	40	25
PSB	210	40
PMS	300	40
PMB	375	50
PBS	470	50
PBB	540	60

Figure 11.30 Membership functions and rule table for scale factor computation

11.6.4.1 Derivation of Q^* and v_{ds}^*

From the d^e - q^e equivalent circuits of Figure 2.23, the stator equations are

$$v_{qs} = R_s i_{qs} + \frac{d}{dt}(\psi_{qs}) + \omega_e \psi_{ds} \quad (2.92)$$

$$v_{ds} = R_s i_{ds} + \frac{d}{dt}(\psi_{ds}) - \omega_e \psi_{qs} \quad (2.93)$$

At steady-state condition under vector control (see Figure 8.36), we can write

$$\frac{d}{dt}(\psi_{qs}) = 0 \quad (11.37)$$

$$\frac{d}{dt}(\psi_{ds}) = 0 \quad (11.38)$$

$$\psi_{ds} = L_s i_{ds} \quad (11.39)$$

$$\psi_{qs} = L_s i_{qs} - \frac{L_m}{L_r} i_{qs} L_m = (L_s - \frac{L_m^2}{L_r}) i_{qs} \quad (11.40)$$

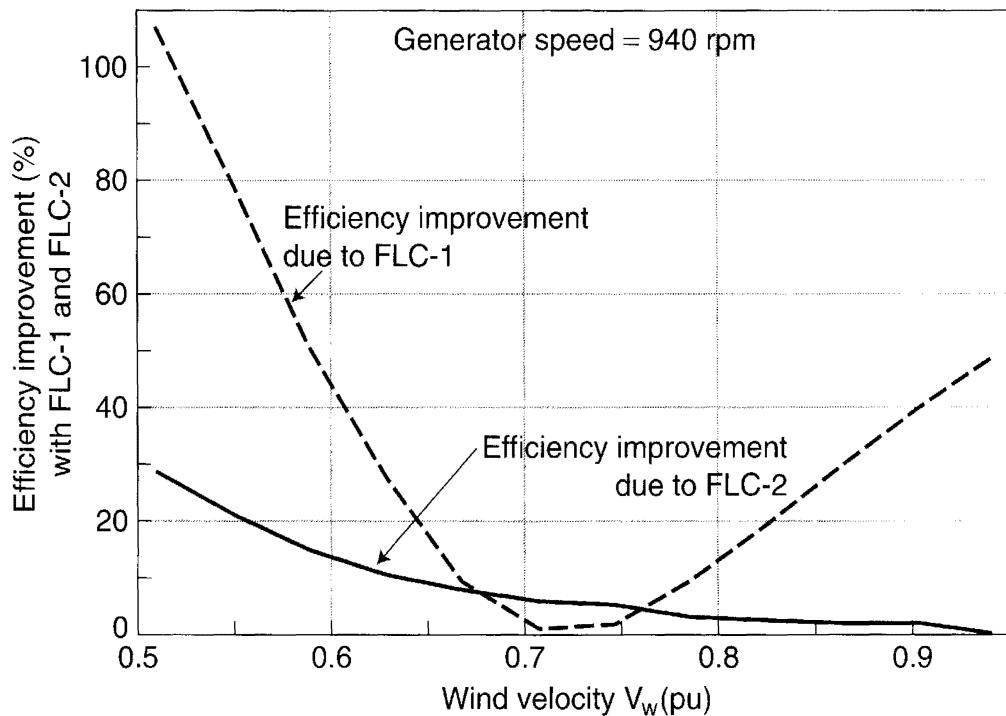


Figure 11.31 Efficiency improvement by controllers FLC-1 and FLC-2 at different wind velocities (1.0 pu = 31.5 mph)

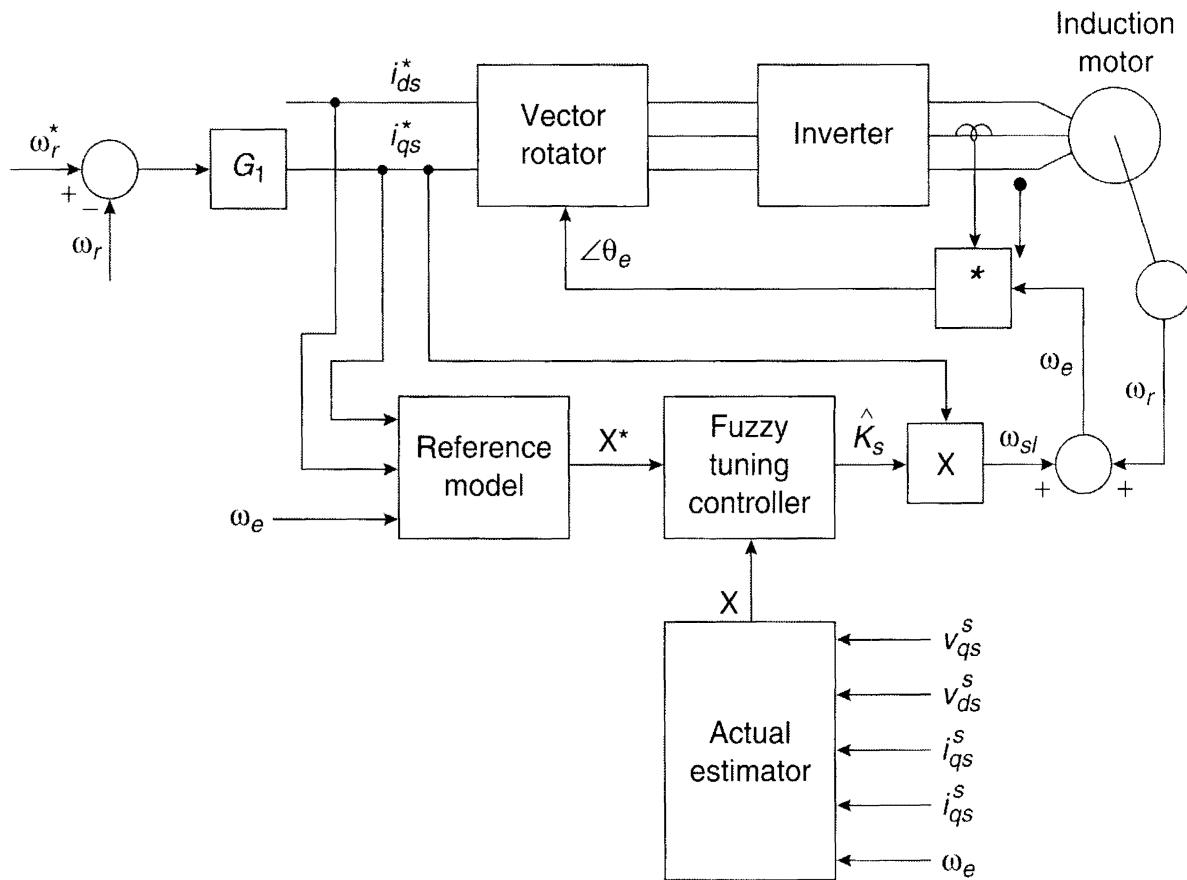


Figure 11.32 Indirect vector control with fuzzy slip gain tuner

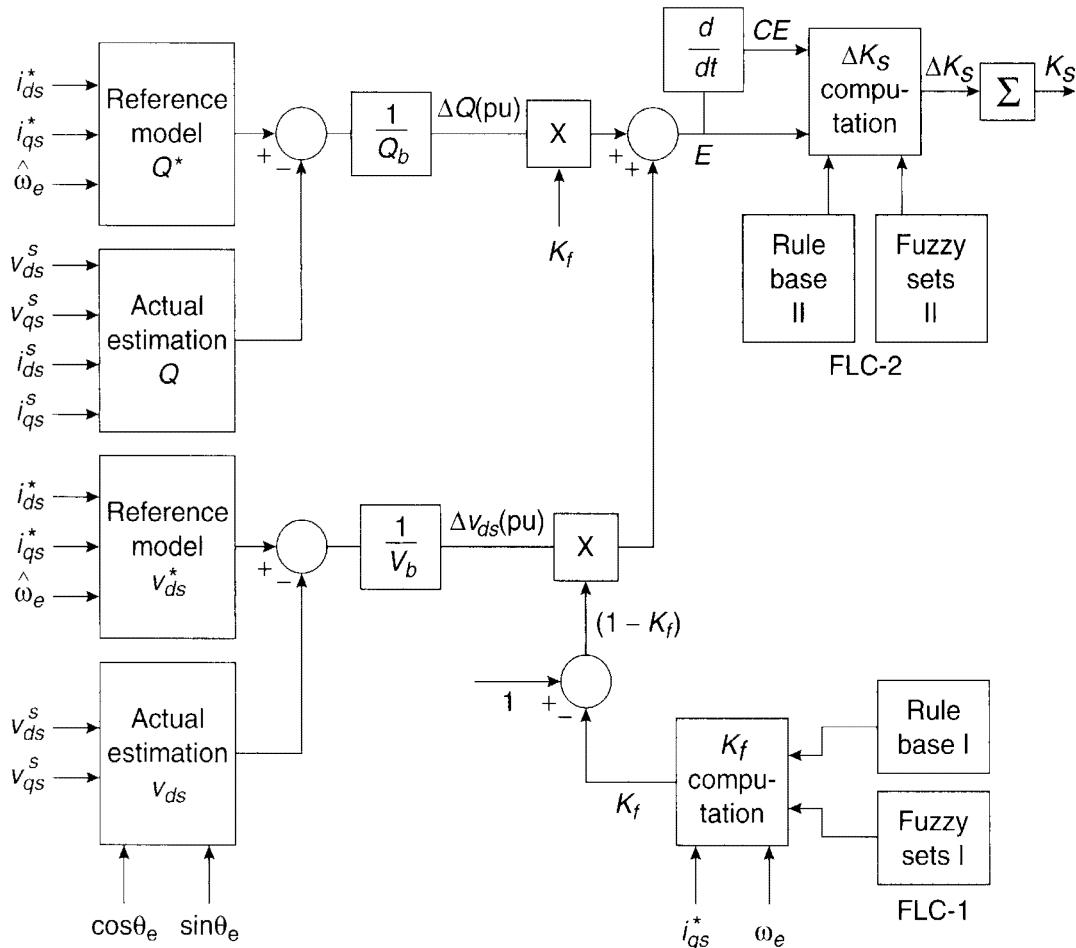


Figure 11.33 FL-based MRAC slip gain tuning control block diagram

Substituting Equations (11.37)–(11.40) in (2.92) and (2.93) and neglecting resistance R_s , we get

$$v_{qs} = \omega_e L_s i_{ds} \quad (11.41)$$

$$v_{ds} = -\omega_e (L_s - \frac{L_m^2}{L_r}) i_{qs} \quad (11.42)$$

The steady-state reactive power expression in a machine is given as [11]

$$Q = v_{qs} i_{ds} - v_{ds} i_{qs} \quad (11.43)$$

Substituting Equations (11.41) and (11.42) in (11.43), the reference model Q^* at the ideally tuned condition is

$$Q^* = \hat{\omega}_e (L_s i_{ds}^{*1} + L_\sigma i_{qs}^{*2}) \quad (11.44)$$

where $L_\sigma = L_s - L_m^2/L_r$. By knowing i_{ds}^* , i_{qs}^* and the estimated value of ω_e (see Equation (8.70)), the reference model Q^* , as shown in Figure 11.33, can be estimated.

The reference model v_{ds}^* is given from Equations (2.93) and (11.40) as

$$v_{ds}^* = R_s i_{ds}^* - \hat{\omega}_e L_\sigma i_{qs}^* \quad (11.45)$$

Note that the reference models are functions of machine parameters which might vary and thus contribute inaccuracy.

The foregoing reference models are then compared with the actual estimation of the respective quantities given by

$$Q = v_{qs}^s i_{ds}^s - v_{ds}^s i_{qs}^s \quad (11.46)$$

$$v_{ds} = v_{qs}^s \sin \theta_e + v_{ds}^s \cos \theta_e \quad (2.75)$$

where $\cos \theta_e$ and $\sin \theta_e$ are the unit vector components. The loop errors are divided by the respective scale factors to derive the per unit variables $\Delta Q(\text{pu})$ and $\Delta v_{ds}(\text{pu})$ for manipulation in the fuzzy controller. There are, in fact, two fuzzy controllers in Figure 11.33, and each is designed with the respective rule base and fuzzy sets. Controller FLC-1, as shown, generates a weighting factor K_f that permits the appropriate distribution of the Q control and v_{ds} control on the torque-speed (i.e., $i_{qs} - \omega_e$) plane. The objective is to assign a high sensitivity to the tuning control by the dominant use of the Q control in the low-speed, high-torque region and the v_{ds} control in the high-speed, low-torque region. The FLC-1 control is simple and is given by an 2×2 rule matrix. A typical rule can be given as

IF speed (ω_e) is high (H) AND torque (i_{qs}) is low (L)
THEN weighting factor (K_f) is low (L)

The combined error signal for both loops in Figure 11.33 is given as

$$E = K_f \Delta Q(\text{pu}) + (1 - K_f) \Delta v_{ds}(\text{pu}) \quad (11.47)$$

Fuzzy controller FLC-2 generates the corrective incremental slip gain ΔK_s based on the combined detuning error E and its derivative CE , as shown. This is basically the fuzzy P-I controller as discussed before for the speed control of the motor. The objective is to provide an adaptive feedback control for fast convergence at any operating point, irrespective of the strength of the E and CE signals. Note that in the ideally tuned condition of the system, both the reference model and actual estimation signals will match, and correspondingly, the E and CE signals will be zero and the slip gain K_s will be set to the correct value K_{s0} . If the system becomes detuned (for example, with rotor resistance variation), the actual Q and v_{ds} signals will deviate from their respective reference values and the resulting error will alter K_s until the system becomes tuned, that is, $E = 0$. The effects of detuning on torque and rotor flux transients are

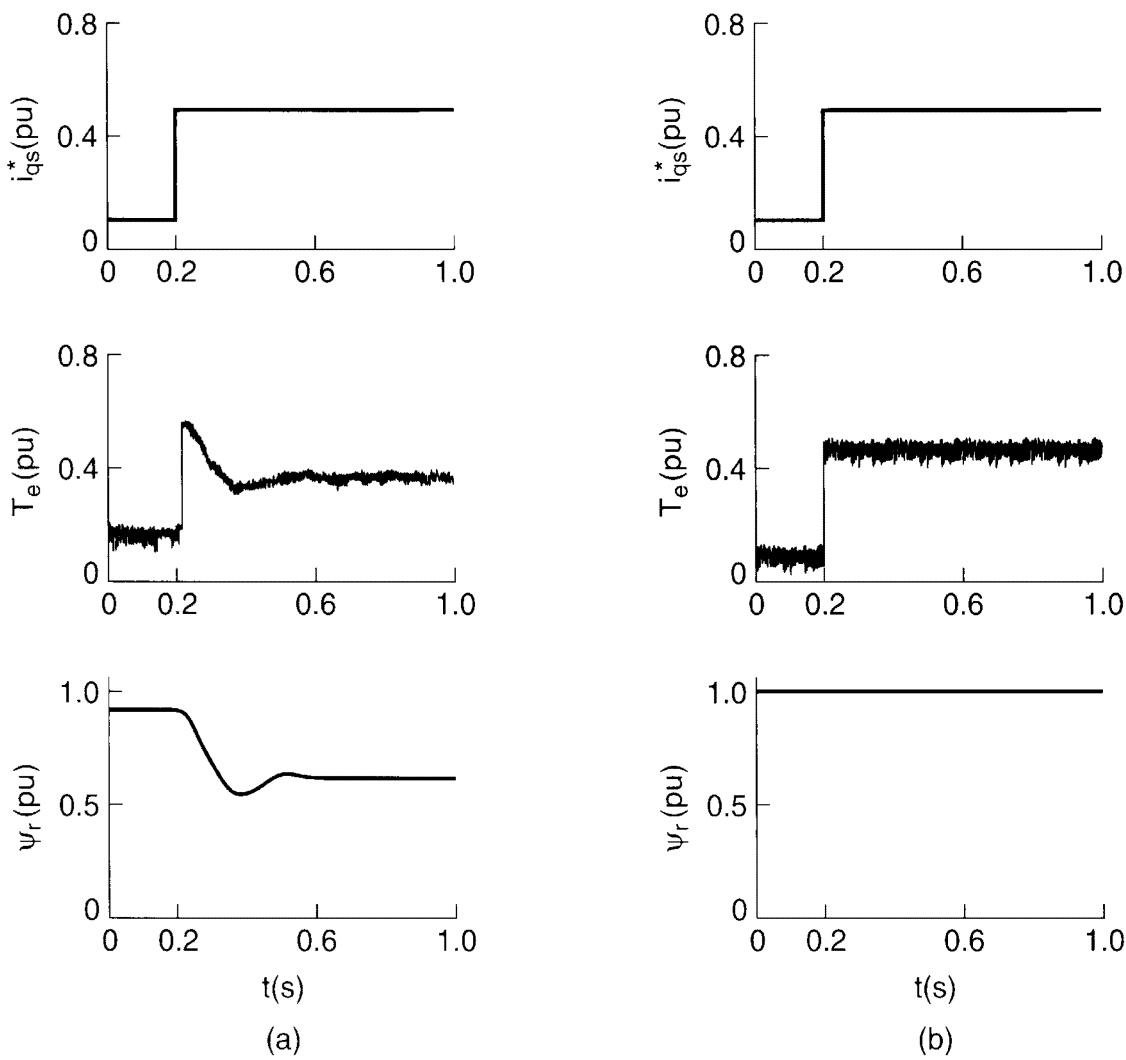


Figure 11.34 Fuzzy tuner performance: (a) Detuned slip gain, (b) Tuned slip gain

illustrated in Figure 11.34. Figure 11.34(a) shows a detuned system where the slip gain was set to twice the correct value ($K_s/K_{s0} = 2$), resulting in higher order dynamics for torque and flux transients as well as reduced torque and flux at steady-state condition. In Figure 11.34(b), on the other hand, the tuned slip gain condition gives ideal transient response.

11.6.5 Stator Resistance R_s Estimation

It was mentioned in Chapter 8 that compensation for the stator resistance variation is important for the correct estimation of flux (ψ_s or ψ_r), torque (T_e), speed (ω_r), and frequency (ω_c), particularly at low speed. Since the stator resistance variation is primarily a function of the stator winding temperature T_s , information about T_s is necessary for the compensation. The average value of T_s can be determined by mounting thermistor probes at several points in the stator, but some type of “sensorless” estimation is desirable.

FL principles can be applied for the approximate estimation of stator resistance by estimating the stator temperature [12]. Consider that a small 5 hp standard (NEMA Class B) induction motor with a shaft-mounted cooling fan and a set of five distributed thermistors

mounted on the stator (for measurement of the stator temperature) is under test. The motor is operated with vector control at the rated flux condition. The machine is mounted on a dynamometer, which is operated at speed control mode. At each speed (i.e., frequency) setting, the drive torque (i.e., the stator current) is varied in steps and the stator temperature rise ($\Delta T_{ss} = T_s - T_A$, where T_A = ambient temperature) is recorded at the steady-state condition. Figure 11.35 shows the experimentally determined ΔT_{ss} curves as functions of stator current I_s and frequency ω_e with ambient temperature $T_A = 25^\circ \text{C}$. Note that at higher frequencies (i.e., speed), the iron loss increases, which tends to a higher temperature rise, but the dominant cooling effect of the shaft-mounted fan essentially decreases the temperature. The curve below the minimum stator current, which corresponds to the magnetizing current for the rated flux, was extrapolated to the vertical axis. The temperature rise is small at low stator currents for the range of frequency variation.

The experimental curves in Figure 11.35 were used to formulate the fuzzy MFs in Figure 11.36 and the corresponding rule matrix in Table 11.4. Basically, the fuzzy estimator algorithm interpolates the $\Delta T_{ss}(pu)$ signal as a function of stator current and frequency. Numerous MFs with crowding at low frequency indicates more accurate estimation of $\Delta T_{ss}(pu)$ near zero speed.

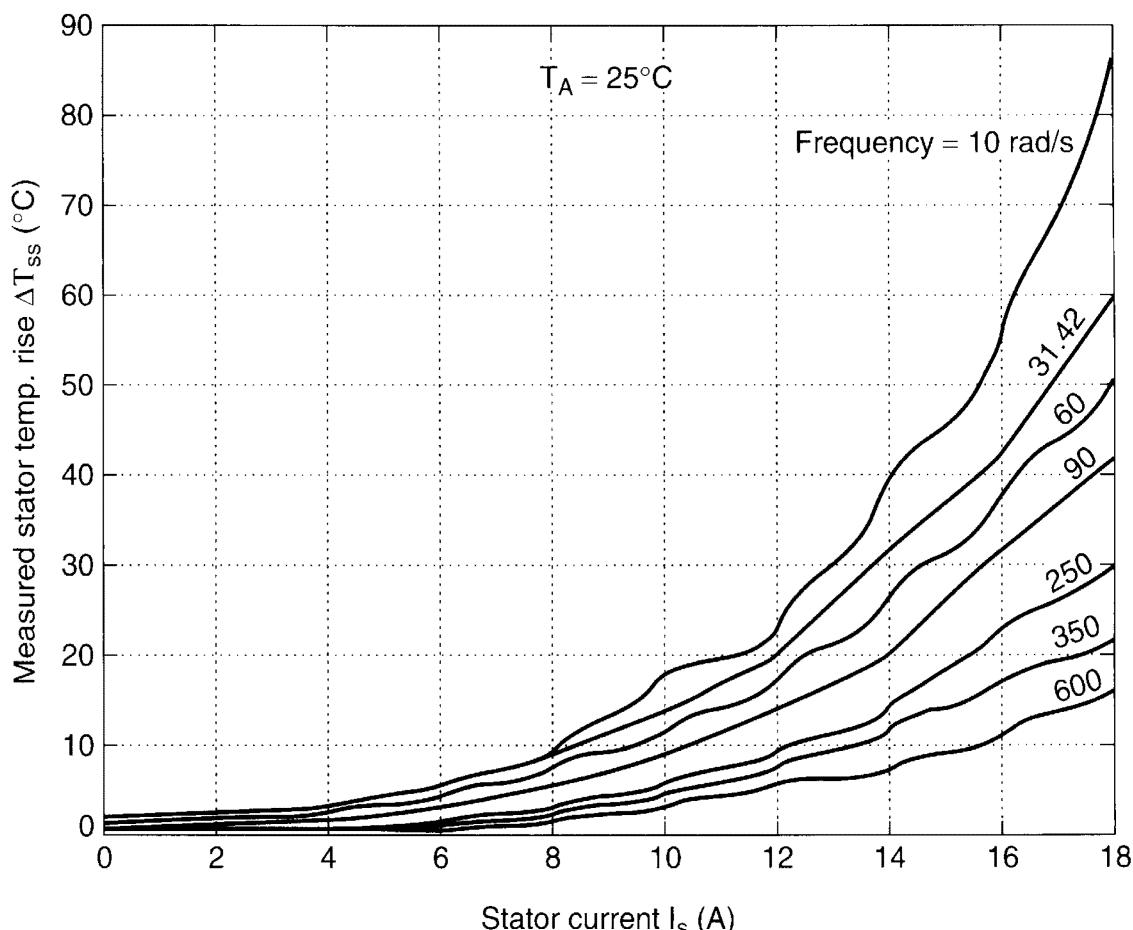


Figure 11.35 Measured stator temperature rise vs. stator current at different frequencies (at steady state)

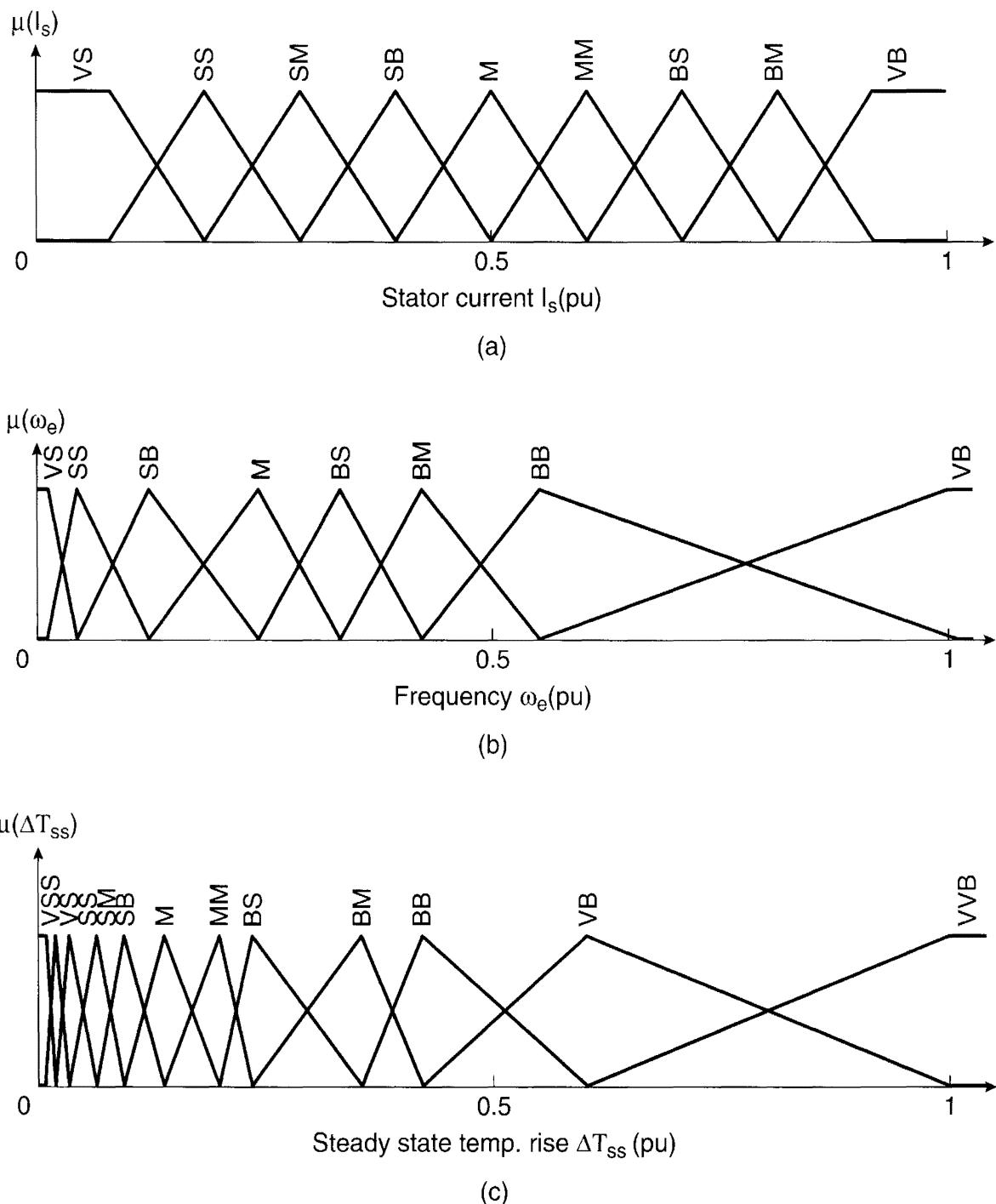


Figure 11.36 Fuzzy estimation MFs, (a) Stator current $I_s(\text{pu})$, (b) Frequency $\omega_e(\text{pu})$, (c) Steady state temperature rise $\Delta T_{ss}(\text{pu})$

Figure 11.37 shows the complete estimation block diagram of stator resistance, which includes a thermal time constant curve and the thermistor network (shown dotted). The thermistor network was used for the generation of Figure 11.35 (as mentioned before), the calibration of T_s , and the estimation of the machine thermal time constant, which will be discussed next. In the present machine, cooling or heat transfer occurs by natural convection as well as by forced cooling by a shaft-mounted fan. The fan's cooling effect is related to the machine's speed.

Table 11.4 Rule Base for $\Delta T_{ss}(pu)$ Estimation

$I_s(pu)$	VS	SS	SB	M	BS	BM	BB	VB
$\omega_e(pu)$	VS	VS	VVS	VVS	VVS	VVS	VVS	VVS
VS	VS	VS	VVS	VVS	VVS	VVS	VVS	VVS
SS	SS	SS	VS	VS	VVS	VVS	VVS	VVS
SM	SM	SM	SS	SS	VS	VS	VVS	VVS
SB	SB	SB	SM	SM	SS	SS	VS	VS
M	MM	M	SB	SB	SM	SM	SS	SS
MM	BS	MM	M	M	SB	SB	SM	SM
BS	BB	BM	MM	MM	M	M	SB	SM
BM	VB	BB	BM	BS	BS	MM	M	SB
VB	VVB	VB	BB	BM	BM	BS	MM	M

The dynamic thermal model of the machine can be approximately represented by a first-order low-pass filter $1/(1 + \tau S)$, as indicated in the figure, where τ = approximate thermal time constant, which is a nonlinear function of speed (or frequency). The relation between τ and ω_e , as shown in Figure 11.37, was determined experimentally as follows: The machine, operating at rated flux, was mounted on the dynamometer and steps of torque (or stator current) were applied at different speed settings. The thermal time constant τ in each case was determined from the average transient temperature rise data on the thermistor network, as mentioned before.

In Figure 11.37, the steady-state ΔT_{ss} can be estimated from the measured (or estimated) values of stator current I_s and frequency ω_e through the fuzzy algorithm described by Figure 11.36 and Table 11.4. Basically, it is fuzzy interpolation of ΔT_{ss} in Figure 11.35 as mentioned before. Generally, four rules will be valid at any instant, and a typical rule may be

IF stator current ($I_s(pu)$) is small-medium (SM)
 AND the frequency ($\omega_e(pu)$) is medium (M)
 THEN the temperature rise ($\Delta T_{ss}(pu)$) is small-small (SS)

Once the steady-state ΔT_{ss} is estimated by the fuzzy estimator, it is converted to a dynamic temperature rise through the low-pass filter and added to the ambient temperature T_A to derive the actual stator temperature T_s . Then, the derivation of actual resistance R_s by the linear expression shown in the figure becomes straightforward. The measured T_s by the thermistor network helps to calibrate the estimated T_s and iterate the estimation algorithm. Finally, the thermistor network is removed, and the estimation algorithm is applied to all machines of the same family. Figure

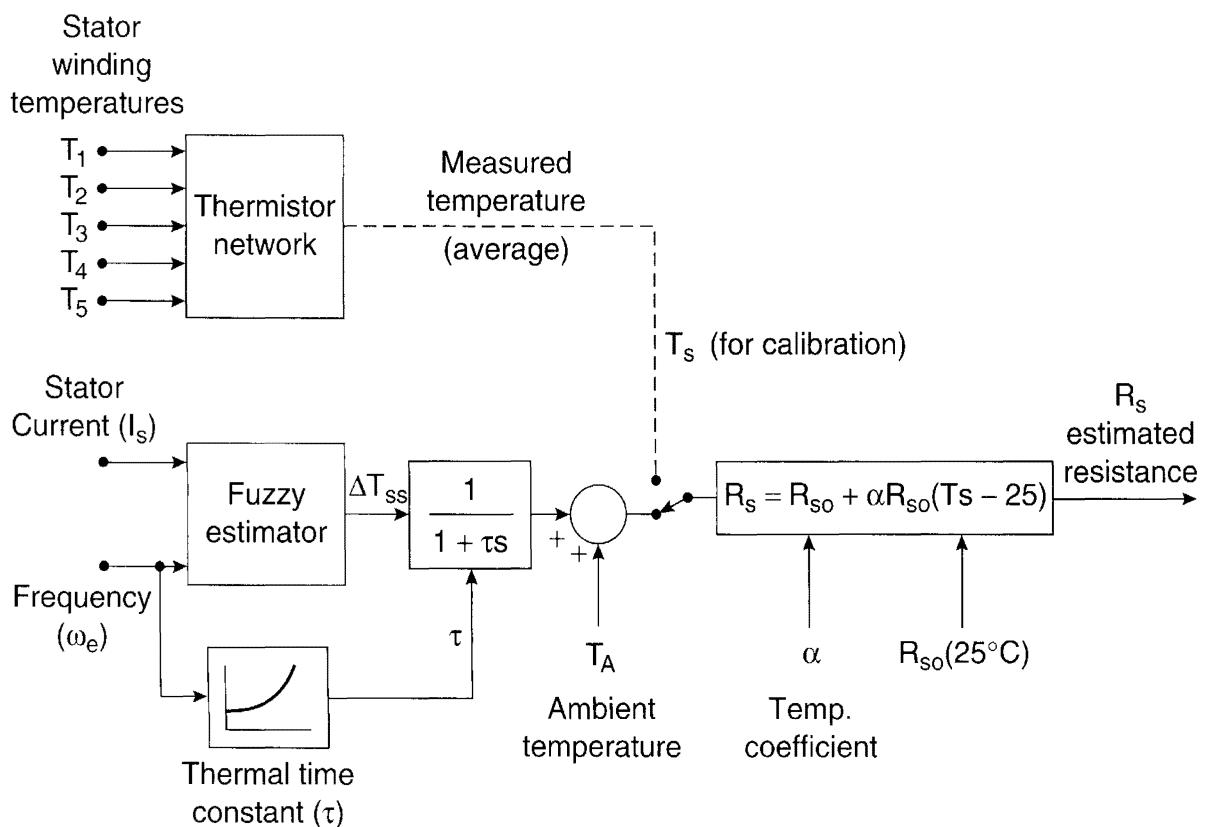


Figure 11.37 Fuzzy estimation block diagram of stator resistance R_s (shown with calibrating thermistor network)

11.38(a) shows the typical T_s estimation accuracy as a function of time at different stator currents, but constant speed, and Figure 11.38(b) shows the corresponding R_s estimation tracking accuracy.

11.6.6 Estimation of Distorted Waves

Power converters are characterized for generation of complex voltage and current waves, and it is often necessary to determine their parameters, such as total rms current I_s , fundamental rms current I_f , active power P , reactive power Q , displacement factor (DPF), distortion factor (DF), and power factor (PF). These parameters can be measured by electronic instrumentation (hardware and software) or estimated by mathematical model, FFT analysis, etc. FL principles can be applied for fast and reasonably accurate estimation of these parameters due to input/output nonlinear mapping (or pattern recognition) property [13]. Of course, fuzzy algorithm development is laborious and requires a large number of iterations for good accuracy. In this section, we will illustrate the estimation of a diode rectifier line current wave by two fuzzy algorithms: Mamdani method and Sugeno (first-order) method.

A typical line current wave for a three-phase diode bridge rectifier with balanced line voltage supply is shown in the lower part of Figure 11.39. The rectifier supplies a voltage-fed inverter-machine load. The parameters of the wave are characterized by the width W and height H of the pulse, which vary with loading for any given value of line voltage and line inductance.

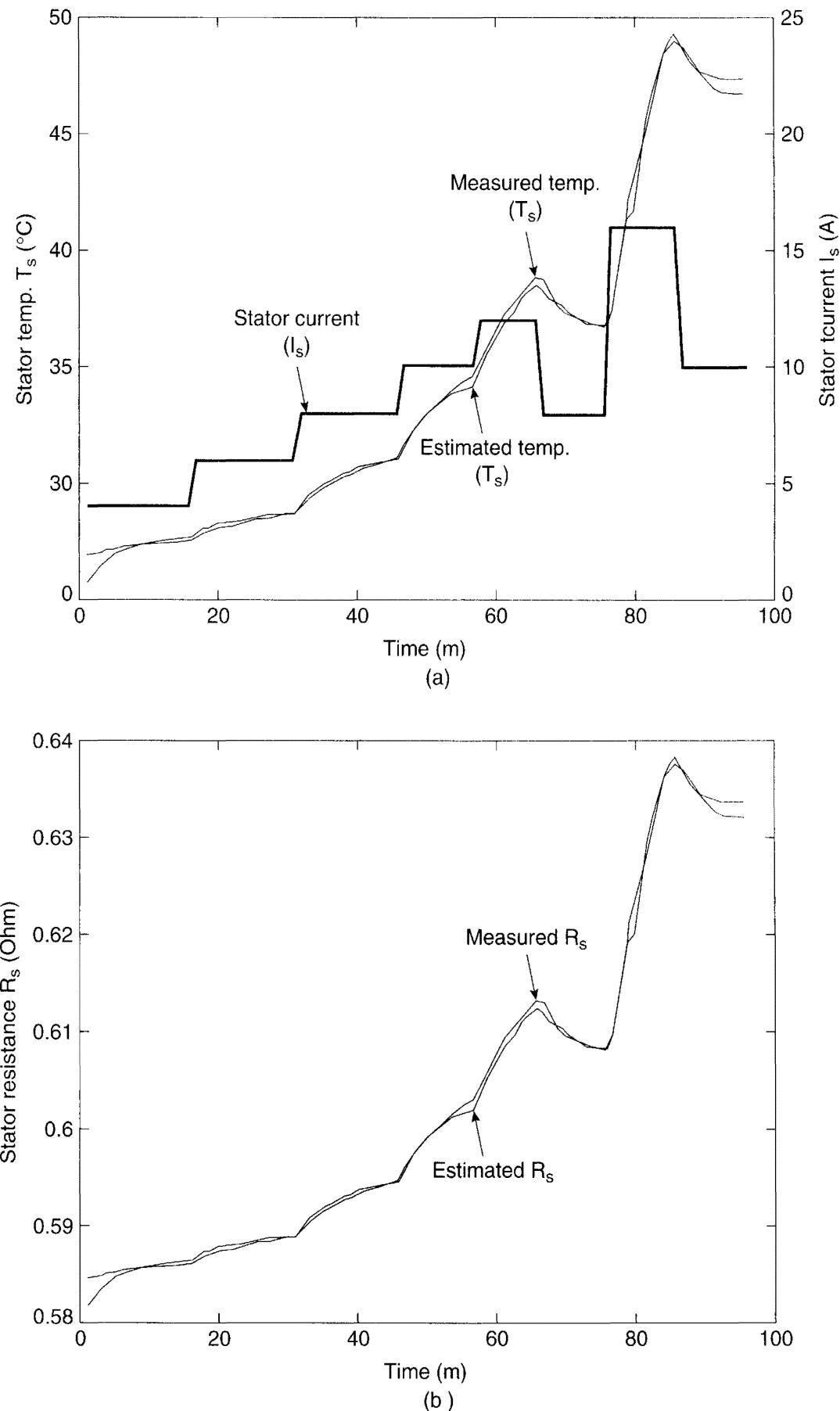


Figure 11.38 (a) Stator temperature estimator performance with dynamically varying stator current but at constant speed [357 rpm], (b) Corresponding resistance estimator performance

11.6.6.1 Mamdani Method

The fuzzy estimation for the rectifier input current wave by the Mamdani method is explained in Figure 11.39. The wave parameters W and H are defined by 6 and 11 MFs, respectively, giving all together 66 rules in the rule matrix(not shown). The number of MFs for rms current I_s and fundamental rms current I_f is 16, but the DPF has only 6 (same as W) MFs. In Figure 11.39, generally four rules are fired, as indicated, and an example rule is

IF $H = \text{PMS}$ AND $W = \text{PSB}$
THEN $I_s = \text{PMM}$, $I_f = \text{PSB}$ and $\text{DPF} = \text{PMS}$

Since the PF is given directly by the relation

$$PF = DPF \cdot \frac{I_f}{I_s} \quad (11.48)$$

it is determined directly from the estimated I_s , I_f , and DPF. The MFs and rule matrix are iterated on the basis of simulation results until the desired accuracy is obtained. Note that one rule gives multiple outputs, and the asymmetrical MF sets are nonidentical because each output has a different degree of nonlinearity.

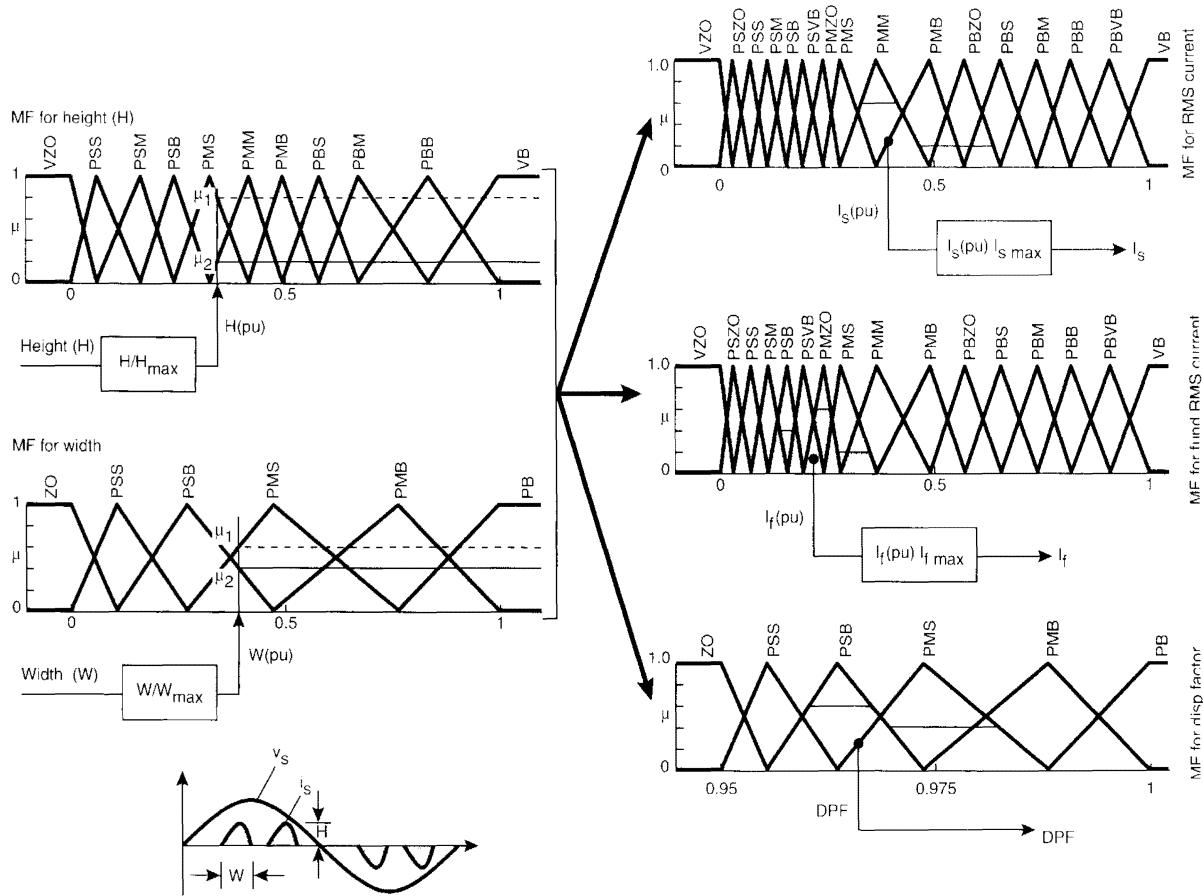


Figure 11.39 Fuzzy estimation for rectifier input current wave (Mamdani method)

11.6.6.2 Sugeno Method

The estimation, as discussed above, is now repeated using the Sugeno first-order method to assess its validity. Figure 11.40 explains the algorithm, where basically fuzzy and analytical methods are combined. Here, the parameter W is considered as the input variable and is described by 8 MFs. The actual W and H parameters of the wave are fed as input to the output linear equations, as shown. This method needs example data to fine-tune the coefficients of the linear equations for correct estimation. Multi-regression linear analysis was used on the data to determine the coefficients, which were then fine-tuned by the simulation results. There are only 8 rules in this case, compared to 66 rules in the Mamdani method.

The estimations in both methods are then compared with the actual values in Figure 11.41 with a gradually increasing inverter machine load current in the dc link. Both methods give very good accuracy in the estimation except the PF , which shows a large error due to the cumulative error contribution of I_s , I_f , and the DPF.

11.7 FUZZY LOGIC TOOLBOX

In this section, we will introduce a commonly used fuzzy system development tool and then give an example application for a drive system. The Fuzzy Logic Tool box (Math Works, Inc.) [3] is a user-friendly fuzzy program development tool in the MATLAB environment. Development can be done using either graphical user interface (GUI) or command-line functions. The toolbox also includes fuzzy clustering and an adaptive neuro-fuzzy inference system (ANFIS) technique. The ANFIS will be discussed in Chapter 12. Once a fuzzy program is developed, its performance can be tested by embedding it in the SIMULINK simulation of the system for fine-tuning, and then, the final stand-alone C program (with the fuzzy inference engine) can be generated, compiled, and downloaded to a DSP for real-time implementation. Or else, the program can be embedded in other external applications.

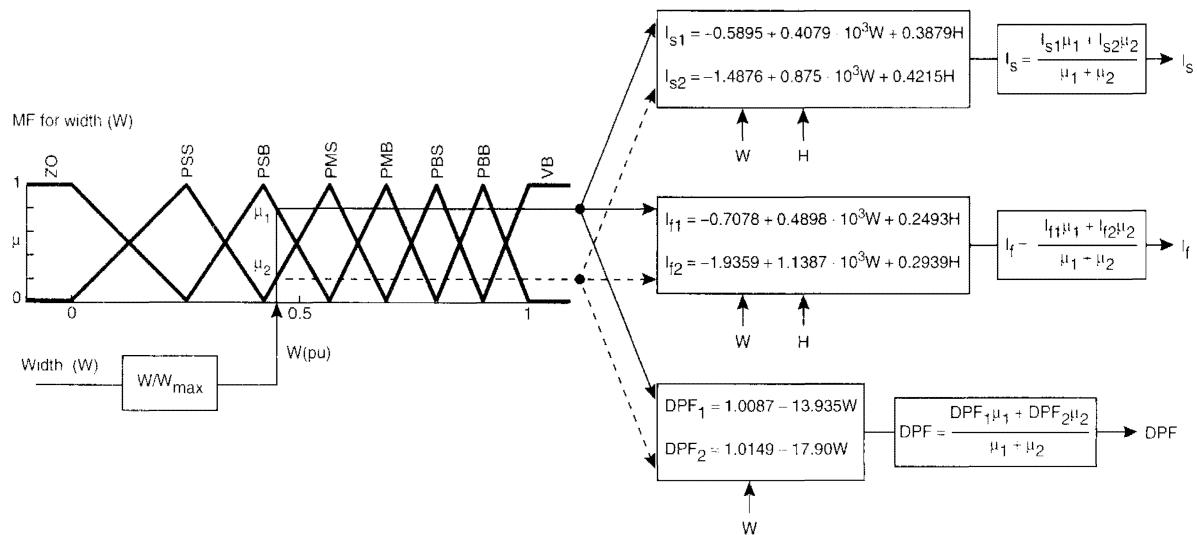


Figure 11.40 Fuzzy estimation for rectifier input current wave (Sugeno method)

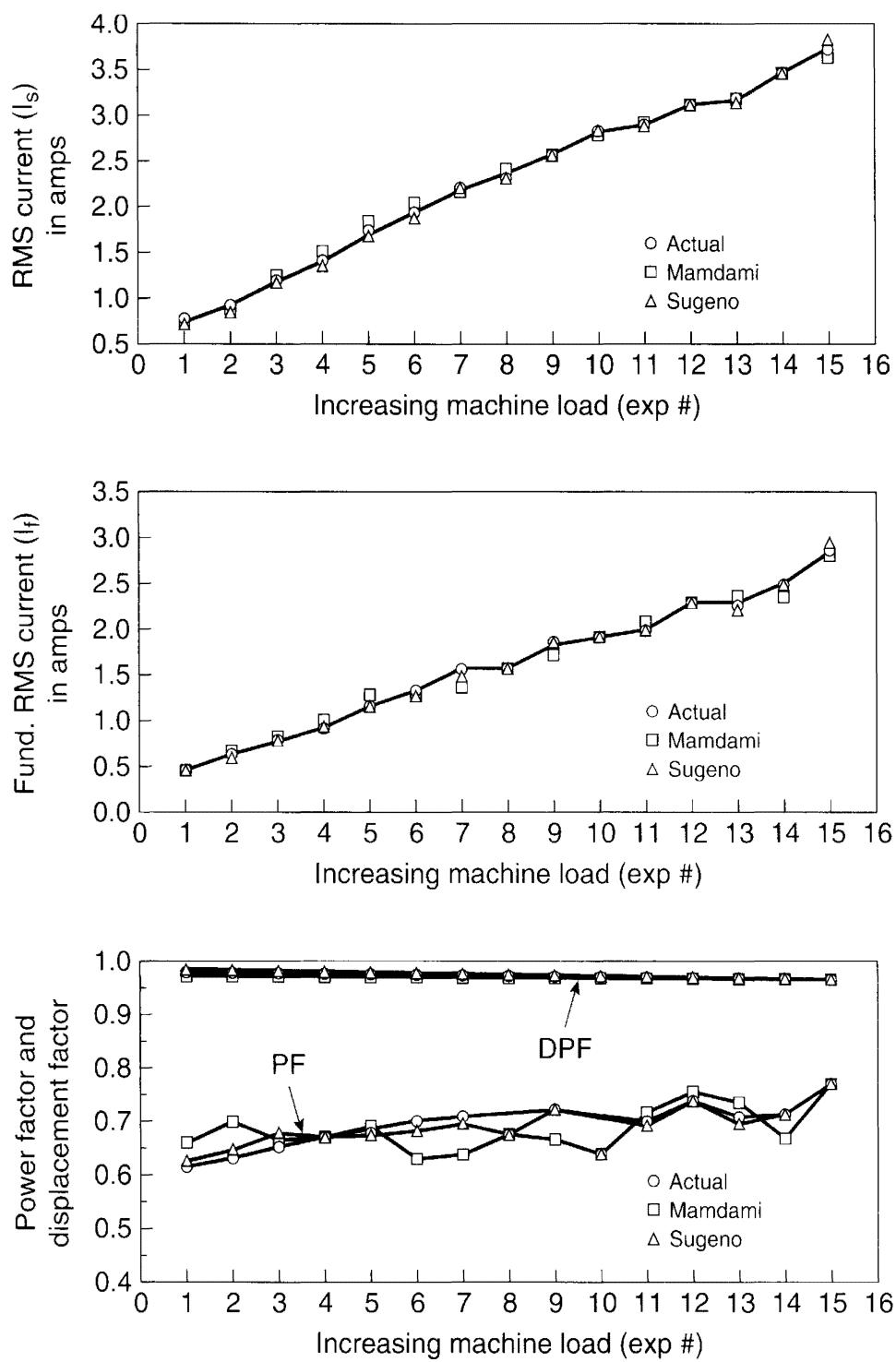


Figure 11.41 Fuzzy estimation accuracy comparison

There are five primary graphical tools for building, editing, and observing fuzzy inference systems in the Fuzzy Logic Toolbox. They are

- Fuzzy Inference System (FIS) Editor
- Membership Function (MF) Editor
- Rule Editor

- Rule Viewer
- Surface Viewer

11.7.1 FIS Editor

The FIS Editor, as shown in Figure 11.42, displays general information about a fuzzy system. At the top left, the names of the defined input fuzzy variables are indicated, and at the right, the output variables are shown. The MFs shown in the boxes are simple icons and do not indicate the actual MFs. Below this, the system name and inference method (either Mamdani or Sugeno) are indicated. At the lower left, the various steps of the inference process, which are user-selectable, are shown. At the lower right, the name of the input or output variable, its associated MF type, and its range are displayed. All the Editor and Viewer boxes depict the development of the restaurant tipping system, which was mentioned previously.

11.7.2 Membership Function Editor

The MF Editor displays and permits editing of all the MFs associated with the input and output variables. Figure 11.43 shows the user interface of the MF Editor. At the upper left, the FIS variables whose MFs can be set are shown. Each setting includes a selection of the MF type and the number of MFs of each variable. At the lower right, there are controls that permit you to

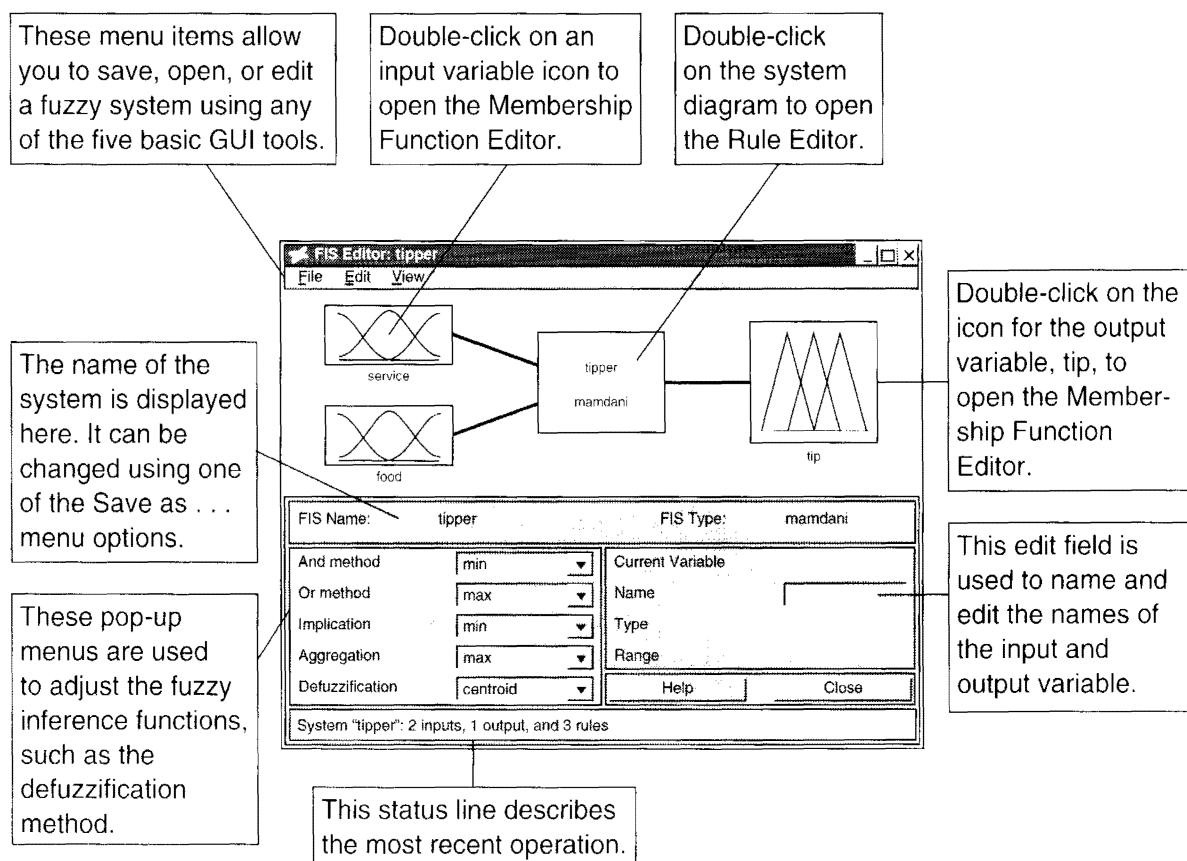


Figure 11.42 User interface for FIS Editor (Courtesy of Math Works, Inc.)

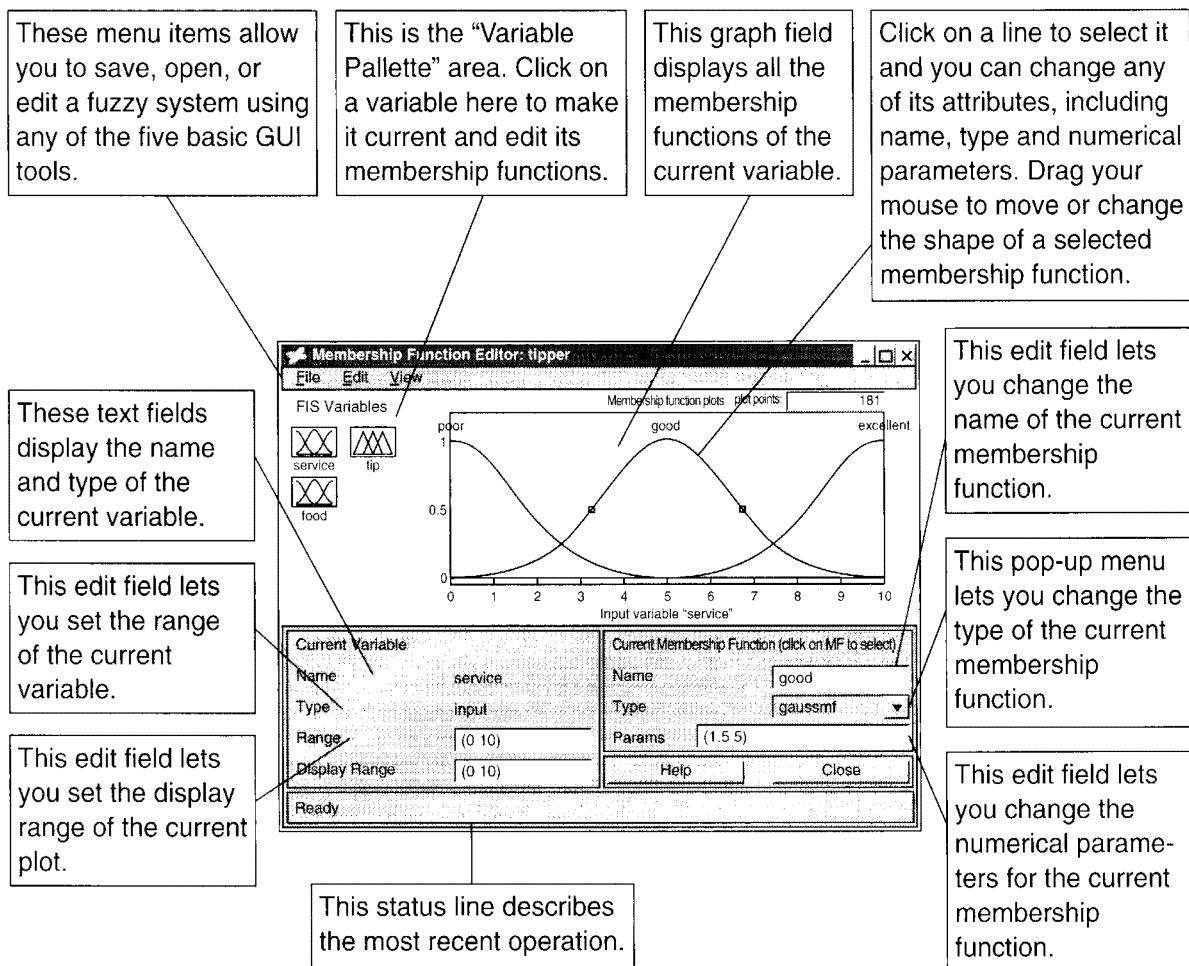


Figure 11.43 User interface for MF Editor (Courtesy of Math Works, Inc.)

change the name, type, and parameters (shape) of each MF, once you have selected it, the MFs of the current variable, which are being edited, are displayed in the graph. At lower left, information about the current variable is given. In the text field, the range (universe of discourse) and display range of the current plot of the variable under consideration can be changed.

11.7.3 Rule Editor

Once the rule matrix is designed on paper and the fuzzy variables are defined in the FIS Editor, construction of the actual rules by the Rule Editor is fairly easy, as shown in Figure 11.44. The logical connectives of rules, AND, OR, and NOT can be selected by buttons. The rules can be changed, deleted, or added, as desired.

11.7.4 Rule Viewer

Once the fuzzy algorithm has been developed, the Rule Viewer, as shown in Figure 11.45, essentially gives a micro view of the FIS, where the operation and contribution of each rule is explained in detail. Each rule is a row of plots, and each column is a variable. Three rules in the restaurant tipping system (connected by OR logic) are depicted in the figure. For a certain setting of the input variables (in this case: 5, 5), the output contribution of each rule, the total

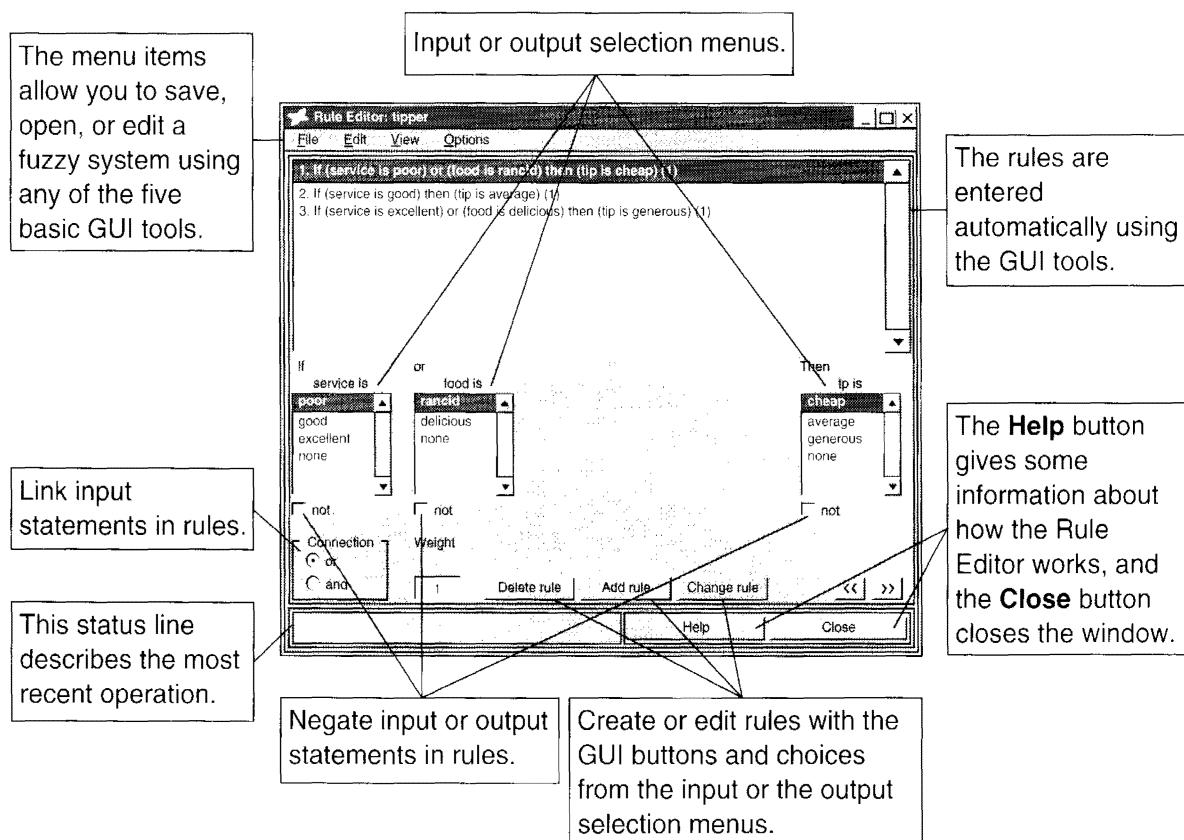


Figure 11.44 User interface for Rule Editor (Courtesy of Math Works, Inc.)

fuzzy output, and the corresponding defuzzified output are shown. The Rule Viewer, with its roadmap of operation of the whole fuzzy algorithm, permits fine-tuning of the MFs and rules.

11.7.5 Surface Viewer

After the fuzzy algorithm has been developed, the Surface Viewer permits you to view the mapping relations between the input variables and output variables, as shown in Figure 11.46. The plot may be three-dimensional, as shown, or two-dimensional. For a larger number of input/output variables, the variables for the Surface Viewer can be selected. Again, by closely examining the Surface Viewer, the algorithm can be iterated.

11.7.6 Demo Program for Synchronous Current Control

So far, we have discussed the basic elements of the Fuzzy Logic Toolbox that work in the MATLAB environment. Now, we will consider an example application for a drive. Figure 11.47 shows an indirect vector-controlled induction motor drive that incorporates fuzzy synchronous current control loops. The command i_{ds}^* , which corresponds to the rated rotor flux, is constant. Command i_{qs}^* is generated from the outer speed control loop, which has P-I control. In this example, a fuzzy P-I control will be developed for the i_{qs} and i_{ds} loops with the help of the Toolbox, and their response will be compared with the respective traditional P-I control in a simulated system.

Figure 11.48 shows the MFs of the fuzzy i_{ds} control loop. All the fuzzy variables, error e , change in error ce , and change in output cu (same as du in Figure 11.16), are described in normal-

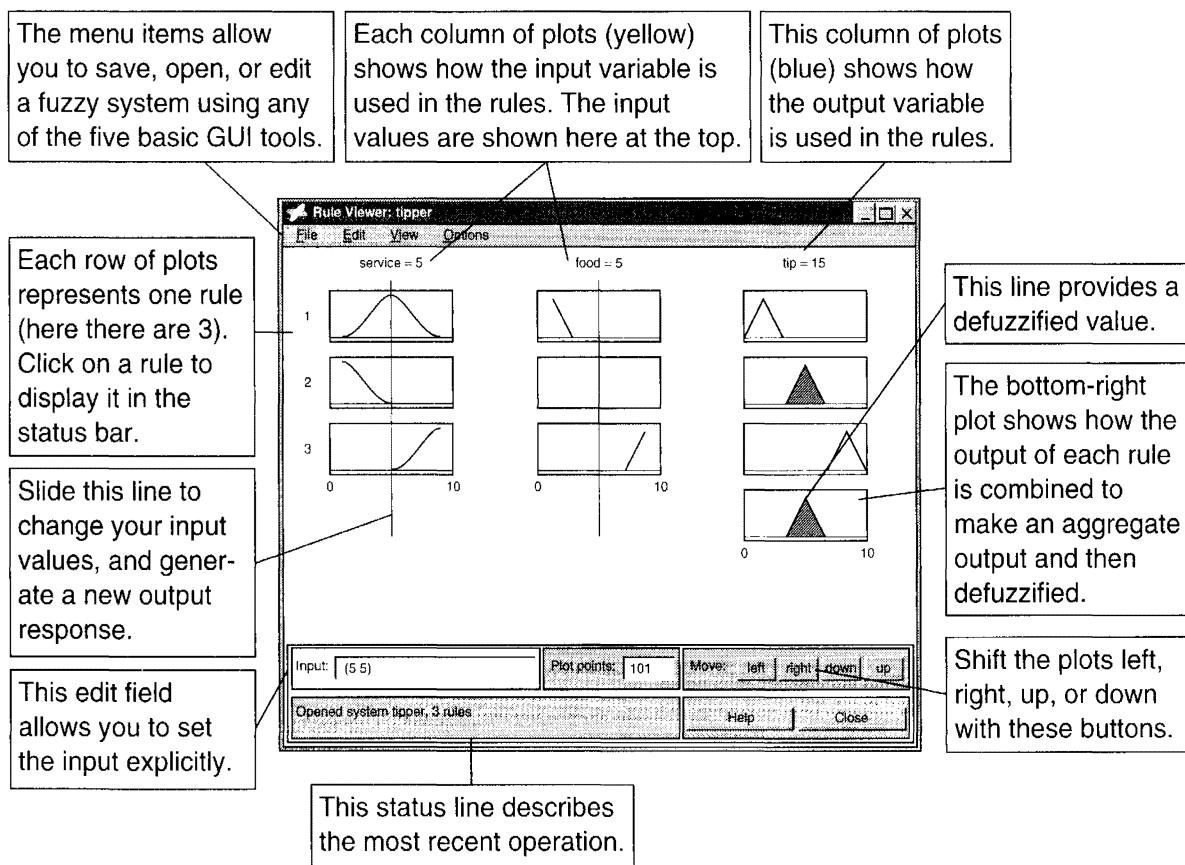


Figure 11.45 User interface for Rule Viewer (Courtesy of Math Works, Inc.)

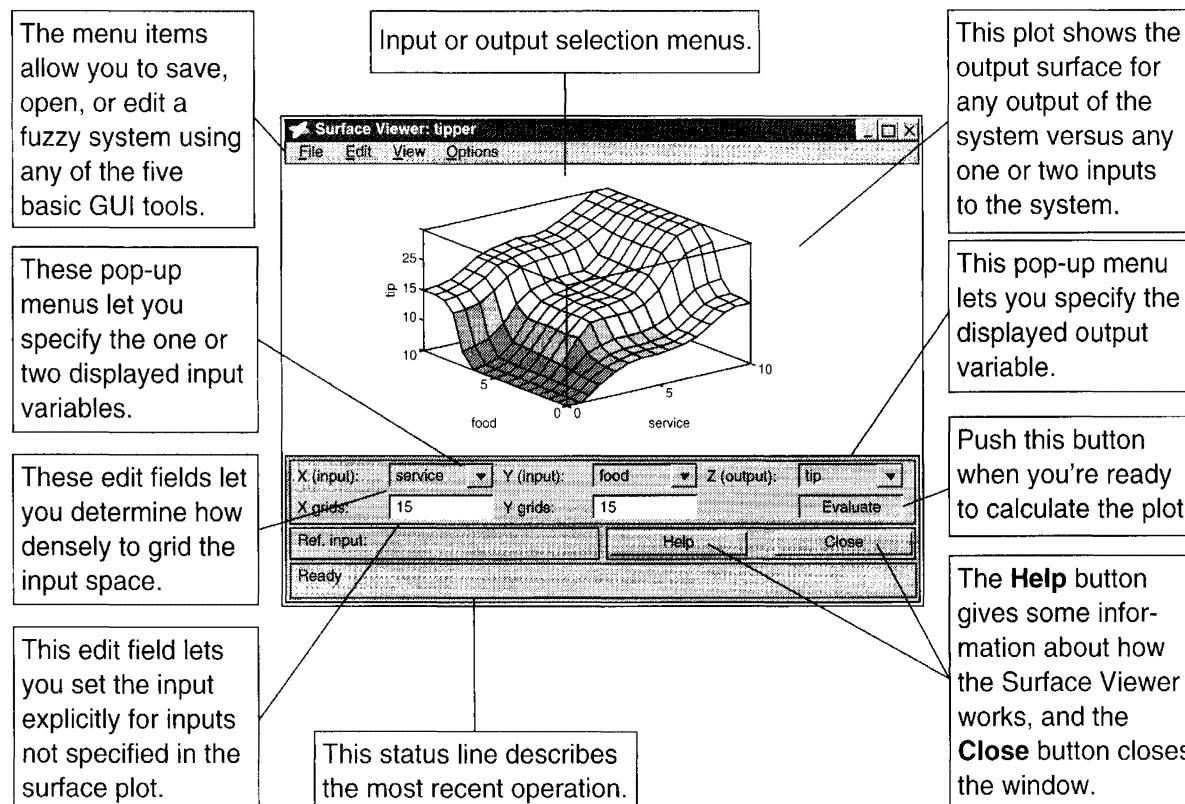


Figure 11.46 User interface to control Surface Viewer (Courtesy of Math Works, Inc.)

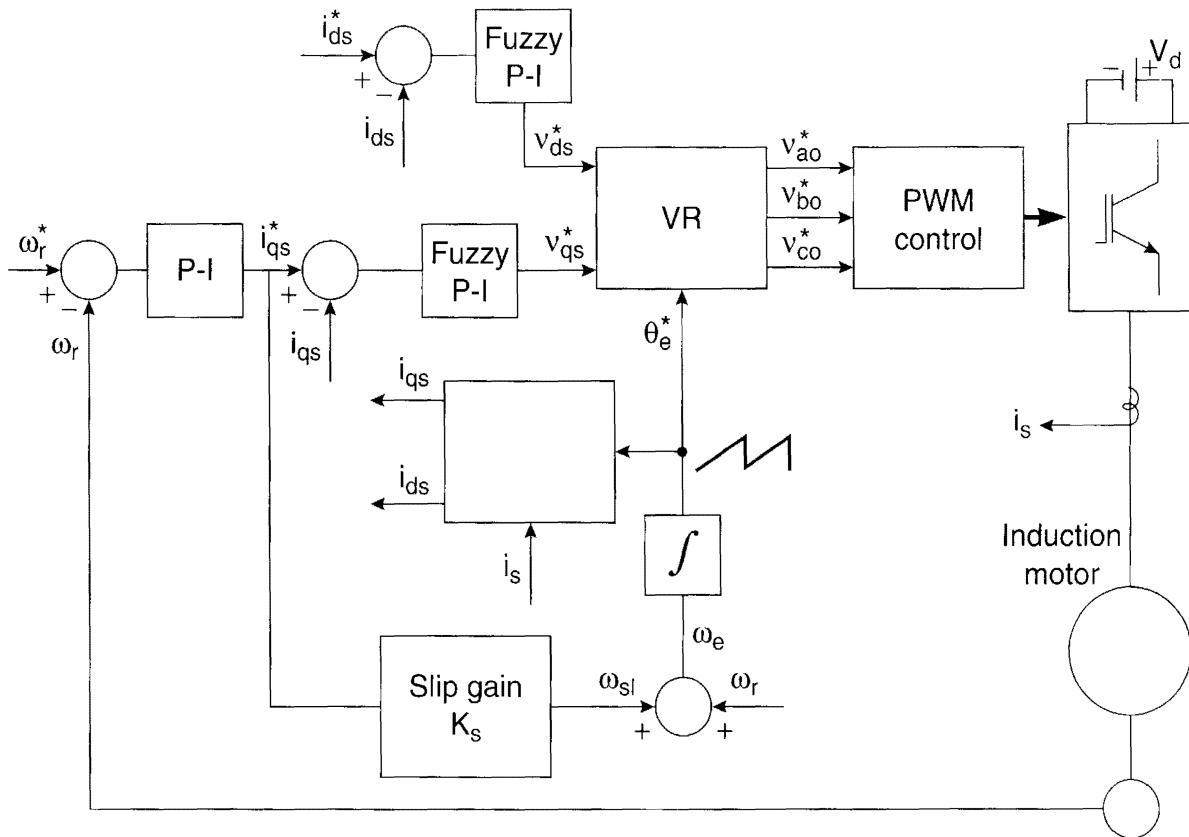


Figure 11.47 Indirect vector-controlled drive incorporating fuzzy i_{qs} and i_{ds} controls

ized form, as shown. Variables e and ce have 7 asymmetrical triangular MFs each, whereas cu has 11 MFs of similar shape. There are $7 \times 7 = 49$ rules all together, which are generated with the help of the Toolbox, as shown in Table 11.5. The MFS and rule matrix are the same for the i_{qs} loop.

As mentioned previously, the MFs and rule table are generated from the experience of the system operation. Once the preliminary control algorithms for both loops are developed, they are incorporated into the SIMULINK simulation of the drive system shown in Figure 11.47.

Figure 11.49 shows the simplified SIMULINK simulation block diagram of Figure 11.47. A discussion of SIMULINK was given in Chapter 5. The details of the “PWM inverter” and “induction motor d - q model” were given in Figures 5.67 and 5.68, respectively. The block “vector controller,” which incorporates the fuzzy controllers, is shown in detail in Figure 11.50, and it is self-explanatory. In this figure, the block “command voltage generator” is essentially the same as the “VR” block in Figure 11.47. The synchronous currents i_{qs} and i_{ds} are generated from the block “abc-syn,” which simulates the following equations:

$$i_{qs}^s = \frac{2}{3}i_a - \frac{2}{3}i_b - \frac{1}{3}i_c \quad (2.72)$$

$$i_{ds}^s = -\frac{1}{\sqrt{3}}i_b + \frac{1}{\sqrt{3}}i_c \quad (2.73)$$

$$i_{qs} = i_{qs}^s \cos \omega_e t - i_{ds}^s \sin \omega_e t \quad (2.74)$$

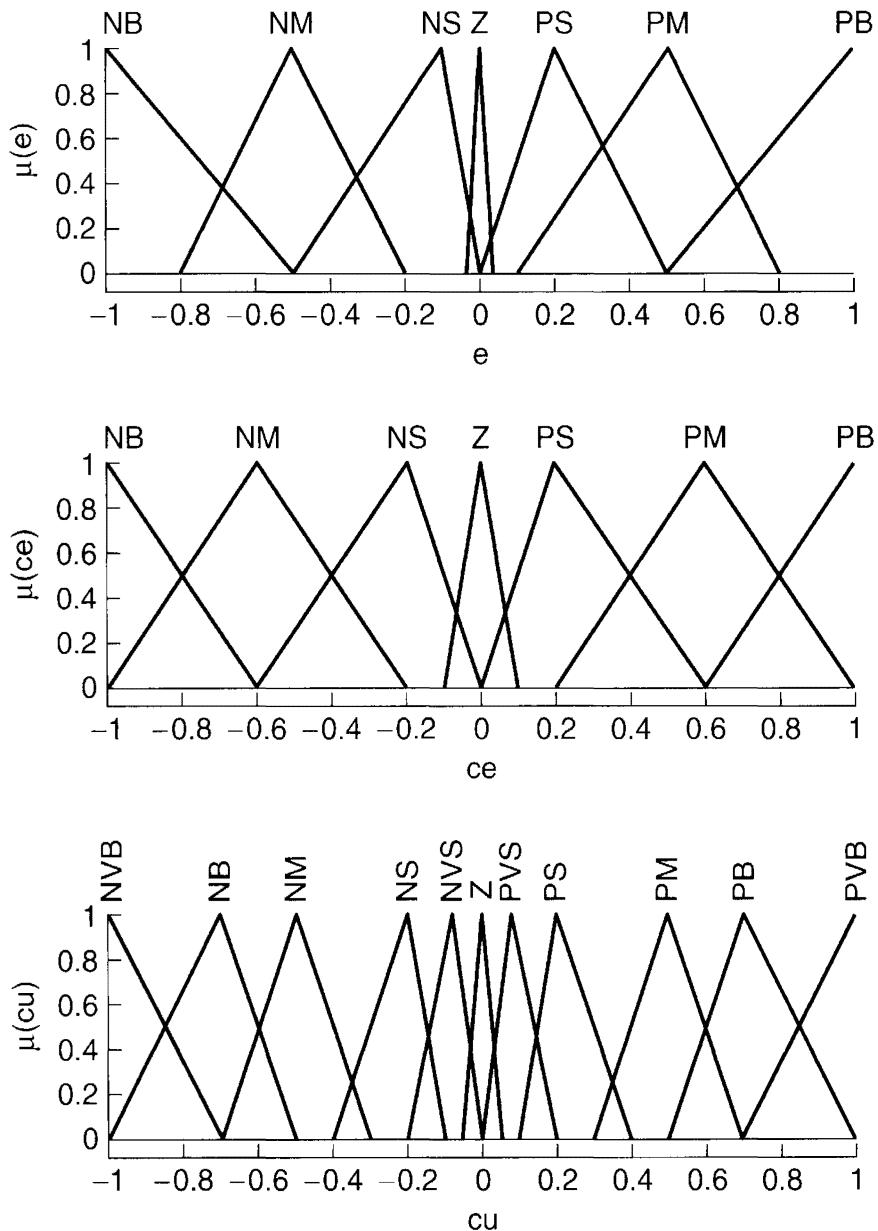


Figure 11.48 Fuzzy controller MFs for i_{ds} loop

$$i_{ds} = i_{qs}^s \sin \omega_e t + i_{ds}^s \cos \omega_e t \quad (2.75)$$

Once the fuzzy controllers were developed and incorporated into the simulated drive system, the simulation performance helped in the iteration of the controllers. The fuzzy controllers were fine-tuned in several stages of that iteration.

Figure 11.51 shows the controller surface, where the horizontal axes are e and ce , and the vertical axis is the output signal cu . The surface indicates that cu is positive high when both e and ce are positive high. On the other hand, when e and ce are negative high, the output signal cu is also negative high.

Table 11.5 Fuzzy controller rule table for i_{ds} loop

1. If (e is NB) and (ce is NB) then (cu is NVB)
2. If (e is NB) and (ce is NM) then (cu is NVB)
3. If (e is NB) and (ce is NS) then (cu is NB)
4. If (e is NB) and (ce is Z) then (cu is NM)
5. If (e is NB) and (ce is PS) then (cu is NS)
6. If (e is NB) and (ce is PM) then (cu is NVS)
7. If (e is NB) and (ce is PB) then (cu is Z)
8. If (e is NM) and (ce is NB) then (cu is NVB)
9. If (e is NM) and (ce is NM) then (cu is NB)
10. If (e is NM) and (ce is NS) then (cu is NM)
11. If (e is NM) and (ce is Z) then (cu is NS)
12. If (e is NM) and (ce is PS) then (cu is NVS)
13. If (e is NM) and (ce is PM) then (cu is Z)
14. If (e is NM) and (ce is PB) then (cu is PVS)
15. If (e is NS) and (ce is NB) then (cu is NB)
16. If (e is NS) and (ce is NM) then (cu is NM)
17. If (e is NS) and (ce is NS) then (cu is NS)
18. If (e is NS) and (ce is Z) then (cu is NVS)
19. If (e is NS) and (ce is PS) then (cu is Z)
20. If (e is NS) and (ce is PM) then (cu is PVS)
21. If (e is NS) and (ce is PB) then (cu is PS)
22. If (e is Z) and (ce is NB) then (cu is NM)
23. If (e is Z) and (ce is NM) then (cu is NS)
24. If (e is Z) and (ce is NS) then (cu is NVS)
25. If (e is Z) and (ce is Z) then (cu is Z)
26. If (e is Z) and (ce is PS) then (cu is PVS)
27. If (e is Z) and (ce is PM) then (cu is PS)
28. If (e is Z) and (ce is PB) then (cu is PM)
29. If (e is PS) and (ce is NB) then (cu is NS)
30. If (e is PS) and (ce is NM) then (cu is NVS)
31. If (e is PS) and (ce is NS) then (cu is Z)
32. If (e is PS) and (ce is Z) then (cu is PVS)
33. If (e is PS) and (ce is PS) then (cu is PS)
34. If (e is PS) and (ce is PM) then (cu is PM)
35. If (e is PS) and (ce is PB) then (cu is PB)
36. If (e is PM) and (ce is NB) then (cu is NVS)
37. If (e is PM) and (ce is NM) then (cu is Z)
38. If (e is PM) and (ce is NS) then (cu is PVS)
39. If (e is PM) and (ce is Z) then (cu is PS)
40. If (e is PM) and (ce is PS) then (cu is PM)
41. If (e is PM) and (ce is PM) then (cu is PB)
42. If (e is PM) and (ce is PB) then (cu is PVB)
43. If (e is PB) and (ce is NB) then (cu is Z)
44. If (e is PB) and (ce is NM) then (cu is PVS)
45. If (e is PB) and (ce is NS) then (cu is PS)
46. If (e is PB) and (ce is Z) then (cu is PM)
47. If (e is PB) and (ce is PS) then (cu is PB)
48. If (e is PB) and (ce is PM) then (cu is PVB)
49. If (e is PB) and (ce is PB) then (cu is PVB)

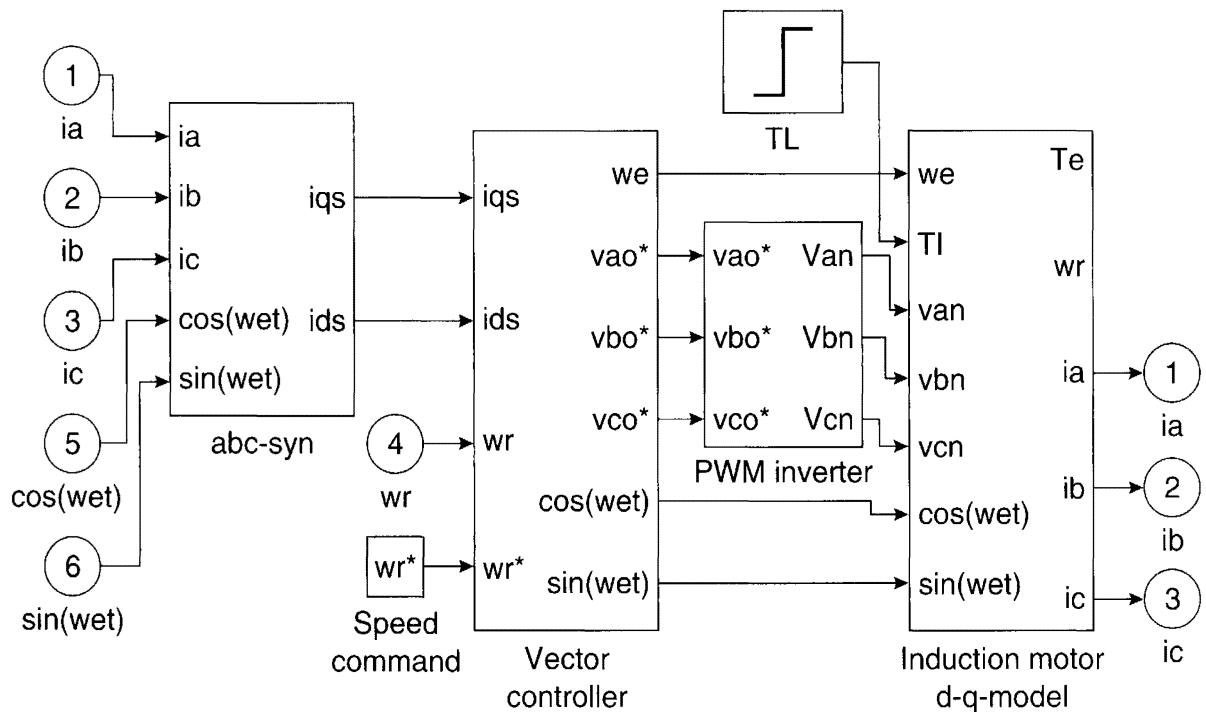


Figure 11.49 SIMULINK simulation block diagram of the drive system in Figure 11.47

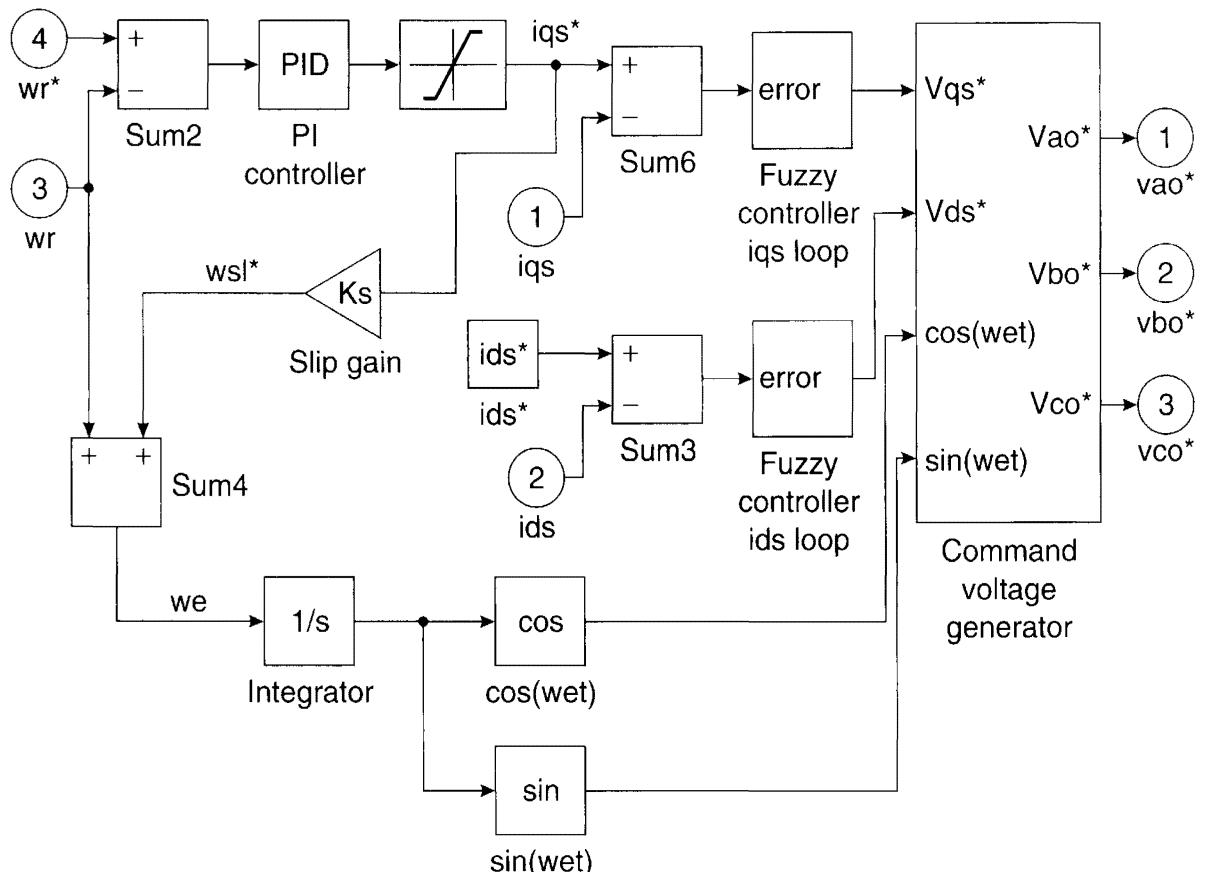


Figure 11.50 SIMULINK simulation block diagram of "vector control" showing fuzzy i_{qs} and i_{ds} controls

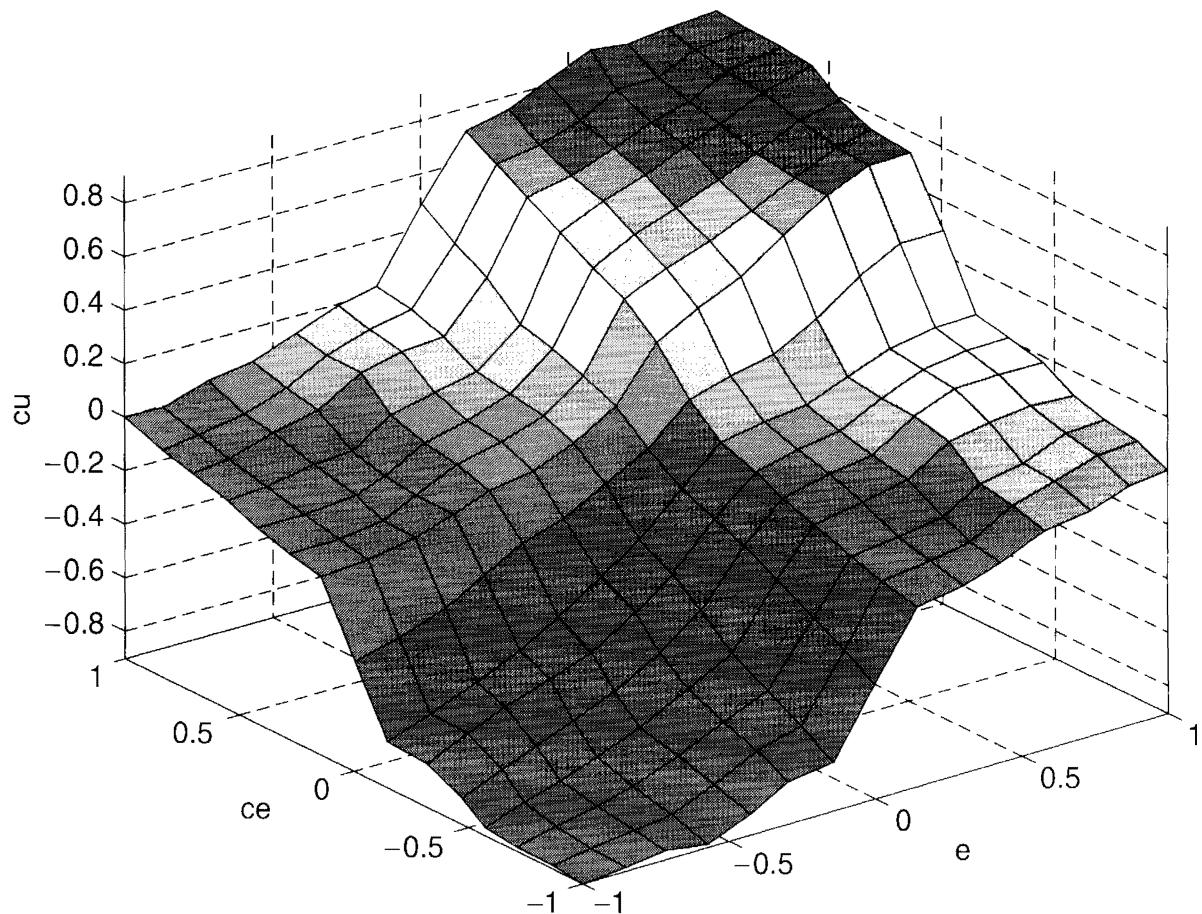


Figure 11.51 Control surface of fuzzy controller

The drive system was simulated with both fuzzy controllers and ordinary P-I controllers and their performances were compared. Figure 11.52 shows the response of the i_{qs} loop with the fuzzy control as well as the P-I control with a stepped speed command, as indicated. The responses are found to be essentially identical. The corresponding response for the i_{ds} loop is shown in Figure 11.53. In this case, the fuzzy control indicates robustness, eliminating transients in the P-I control.

11.8 GLOSSARY

Aggregation – Combination of consequents of each rule.

Antecedent – The initial (or “IF”) part of a fuzzy rule.

COA (center of area) defuzzification – A method of calculating crisp output from the center of gravity of the output membership function (also called centroid defuzzification).

Consequent – The final (or “THEN”) part of a rule.

Degree of membership – A number between 0 and 1 that represents the output value of a membership function.

Defuzzification – The process of transforming a fuzzy output of a fuzzy system into a crisp output.

DOF (degree of fulfillment) – The degree to which the antecedent part of a fuzzy rule is satisfied.

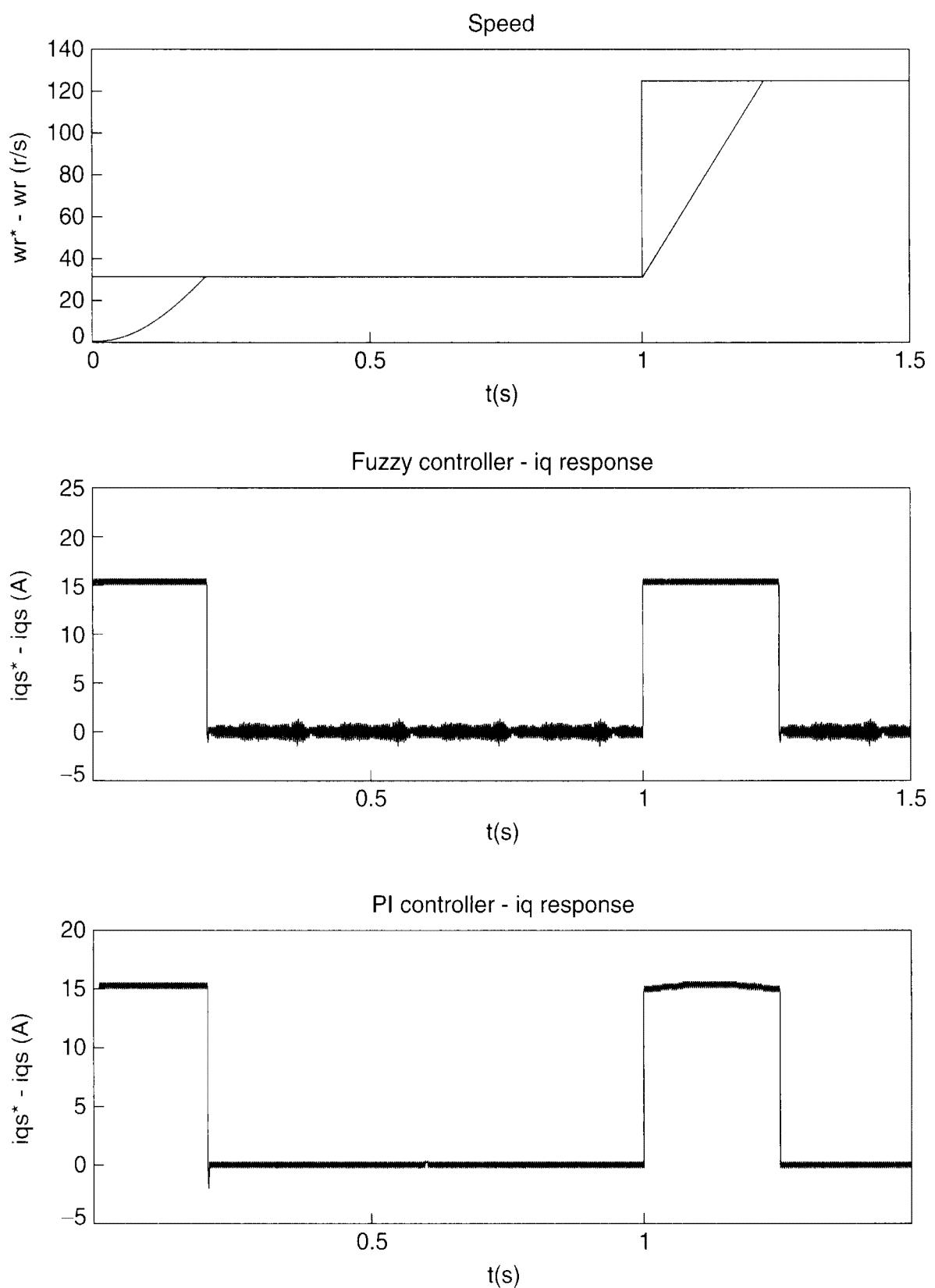


Figure 11.52 Response of i_{qs} loop with fuzzy control and P-I control

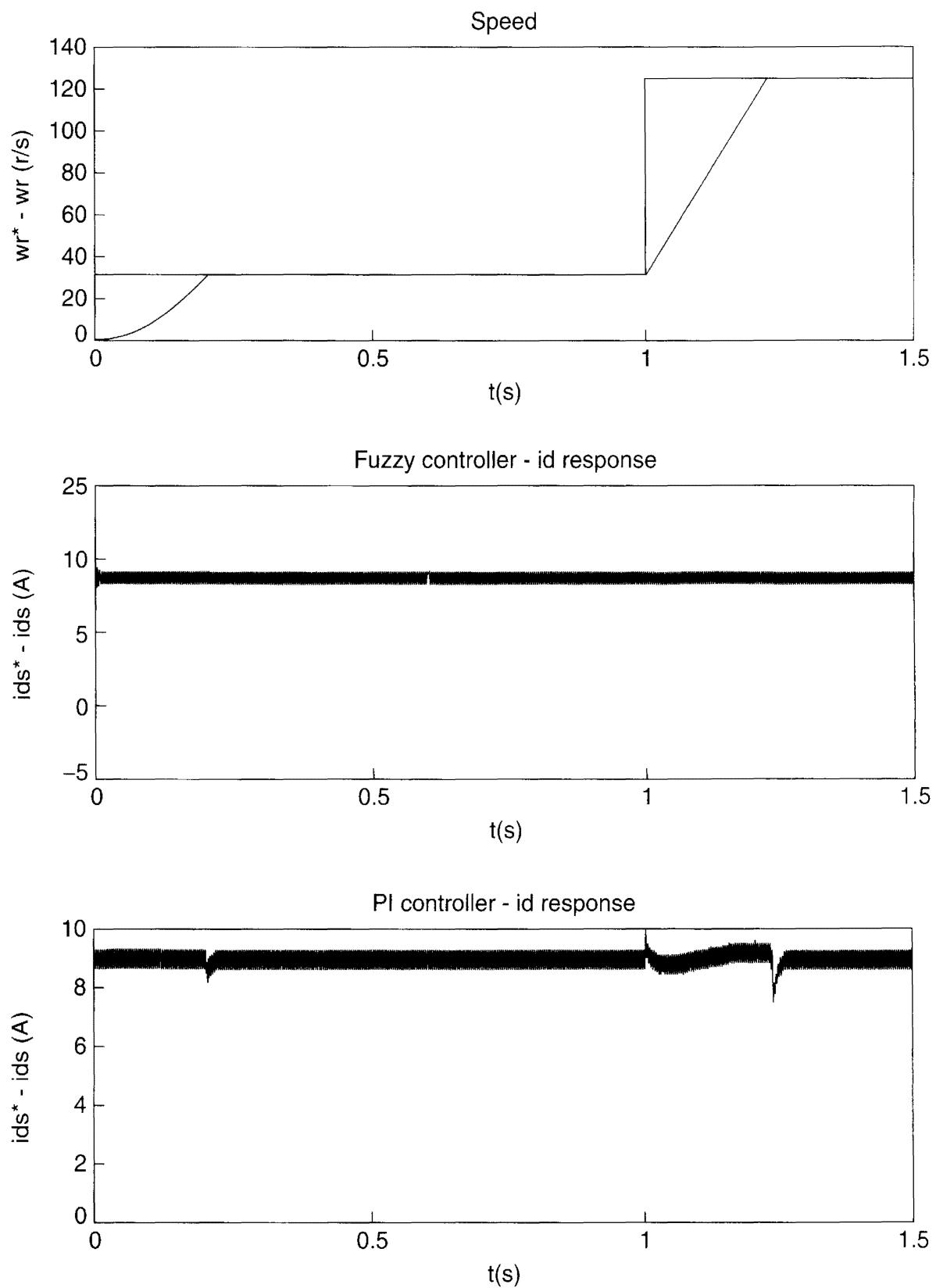


Figure 11.53 Response of i_{ds} loop with fuzzy control and P-I control

Fuzzification – The process of generating a membership value for a fuzzy variable using a membership function.

Fuzzy composition – A method of deriving fuzzy control output from given fuzzy control inputs.

Fuzzy control – A process control that is based on fuzzy logic and is normally characterized by “IF ... THEN...” rules.

Fuzzy system – A system that uses fuzzy reasoning to map an input space to an output space.

Fuzzy implication – The process of shaping the fuzzy set in the consequent part based on the antecedent part of a rule.

Fuzzy rule – IF/THEN rule relating input (conditions) fuzzy variables to output (actions) fuzzy variables.

Fuzzy set (or fuzzy subset) – A set consisting of elements having degrees of membership varying between 0 (nonmember) to 1 (full member). It is usually characterized by a membership function and associated with linguistic values, such as SMALL, MEDIUM, etc.

Fuzzy set theory – A set theory based on fuzzy logic.

Fuzzy variable – A variable that can be defined by fuzzy sets.

Height defuzzification – A method of calculating a crisp output from a composed fuzzy value by performing a weighted average of individual fuzzy sets. The heights of each fuzzy set are used as weighting factors in the procedure.

Linguistic variable – A variable (such as temperature, speed, etc.) whose values are defined by language, such as LARGE, SMALL, etc.

Membership function (MF) – A function that defines a fuzzy set by associating every element in the set with a number between 0 and 1.

MOM (mean of maxima) defuzzification – A method of calculating crisp output from an output membership function where the highest MF component is considered only.

Singleton – A special fuzzy set that has the membership value of 1 at a particular point and 0 elsewhere.

SUP-MIN composition – A composition (or inference) method for constructing the output membership function by using the maximum and minimum principle.

Universe of discourse – The range of values associated with a fuzzy variable.

11.9 SUMMARY

This chapter broadly reviewed FL principles and their application in power electronic systems. FL, as a part of AI, and its difference from an ES, was explained. The analogy and the differences between FL and Boolean logic were explained, particularly relating to operations on fuzzy sets. The principal steps of a fuzzy inference system were explained and illustrated with the restaurant tipping example for clarity of understanding. Several implication and defuzzification methods were explained. The chapter particularly emphasized the control applications of FL. After

explaining the basic control principles, a number of applications in power electronic systems including a vector-controlled induction motor drive, an efficiency improvement control by an on-line search of flux, a wind electric generation system, slip gain tuning by a fuzzy MRAC, a stator resistance estimation, and an estimation of a distorted wave, were reviewed from the literature. The fuzzy speed control algorithm was described in detail with a numerical example for clarity.

Numerous software and hardware tools are available for the development and implementation of a fuzzy system. The salient features of a popular tool, MATLAB/Fuzzy Logic Toolbox, were reviewed, and finally, an example application using the Toolbox was discussed. A Glossary was added to the end of this chapter. Hopefully, the knowledge gained in this chapter will help the reader use FL in new applications.

REFERENCES

1. B. K. Bose, "Expert system, fuzzy logic and neural network applications in power electronics and motion control", *Proc. IEEE*, vol. 82, pp. 1303-1323, Aug. 1994.
2. L. H. Tsoukalas and R. E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, Wiley, NY , 1997.
3. Math Works, *Fuzzy Logic Toolbox User's Guide*, Jan., 1998.
4. T. Takagi and M. Sugeno, "Fuzzy identification of a system and its applications to modeling and control", *IEEE Trans. Syst. Man and Cybern.*, vol. 15, pp. 116-132, Jan./Feb. 1985.
5. G. C. D. Sousa and B. K. Bose, "A fuzzy set theory based control of a phase controlled converter dc drive", *IEEE Trans. of Ind. Appl.*, vol. 30, pp. 34-44, Jan./Feb. 1994.
6. Y. F. Li and C. C. Lau, "Development of fuzzy algorithm for servo systems", *IEEE Control Syst. Magazine*, Apr. 1989.
7. G. C. D. Sousa, B. K. Bose, and J. G. Cleland, "Fuzzy logic based on-line efficiency optimization control of an indirect vector controlled induction motor drive", *IEEE Trans. of Ind. Elec.*, vol. 42, pp. 192-198, Apr. 1995.
8. M. G. Simoes, B. K. Bose, and R. J. Spiegcl, "Design and performance evaluation of a fuzzy logic based variable speed wind generation system", *IEEE Trans. of Ind. Appl.*, vol. 33, pp. 956-965, July/Aug. 1997.
9. T. M. Rowan, R. J. Kerkman, and D. Leggate, "A simple on-line adaptation for indirect field orientation of an induction machine", *IEEE Trans. on Ind. Appl.*, vol. 42, pp. 129-132, Apr. 1995.
10. G. C. D. Sousa, B. K. Bose, and K. S. Kim, "Fuzzy logic based on-line tuning of slip gain for an indirect vector controlled induction motor drive", *IEEE IECON Conf. Rec.*, pp. 1003-1008, 1993.
11. H. Akagi, Y. Kanazawa, and A. Nabae, "Instantaneous reactive power compensators comprising switching devices without energy storage components", *IEEE Trans. on Ind. Appl.*, vol. 20, pp. 625-630, May/June 1984.
12. B. K. Bose and N. R. Patel, "Quasi-fuzzy estimation of stator resistance of induction motor", *IEEE Trans. of Pow. Elec.*, vol. 13, pp. 401-409, May 1998.
13. M. G. Simoes and B. K. Bose, "Application of fuzzy logic in the estimation of power electronic waveforms", *IEEE IAS Annu. Meet. Conf. Rec.*, pp. 853-861, 1993.
14. I. Miki, N. Nagai, S. Nishigama, and T. Yamada, "Vector control of induction motor with fuzzy PI controller", *IEEE IAS Annu. Meet. Conf. Rec.*, pp. 342-346, 1991.