Chapter 10 Fuzzy Control and Fuzzy Expert Systems

The fuzzy logic controller (FLC) is introduced in this chapter. After introducing the architecture of the FLC, we study its components step by step and suggest a design procedure of the FLC. An example of the design procedure is also given. The structure and function of the fuzzy expert systems are similar to those of the FLC, and thus the design procedure of FLC can be used in the fuzzy expert systems.

10.1 Fuzzy logic controller

10.1.1 Advantage of fuzzy logic controller

Fuzzy logic is much closer in spirit to human thinking and natural language than the traditional (classical) logical systems. Basically, it provides an effective means of capturing the approximate, inexact nature of the real world. Therefore, the essential part of the fuzzy logic controller (FLC) is a set of linguistic control strategy based on expert knowledge into an automatic control strategy.

The FLC is considered as a good methodology because it yields results superior to those obtained by conventional control algorithms. In particular the FLC is useful in two cases.

(1) The control processes are too complex to analyze by conventional quantitative techniques.

(2) The available sources of information are interpreted qualitatively, inexact, or uncertainly.

Indeed, the advantage of FLC can be summarized as follows.

(1) Parallel or distributed control: in the conventional control system, a control action is determined by single control strategy like \( \mu = f(x_1, x_2, \ldots, x_n) \). But in FLC, the control strategy is represented by multiple fuzzy rules, and thus it is easy to represent complex systems and nonlinear systems.

(2) Linguistic control: the control strategy is modeled by linguistic terms and thus it is easy to represent the human knowledge.

(3) Robust control: there are more than one control rule and thus, in general, one error of a rule is not fatal for the whole system.

10.1.2 Configuration of fuzzy logic controller

There is no systematic procedure for the design of an FLC. However we can present here a basic configuration of FLC as shown in Fig 10.1. The configuration consists of four main components: fuzzification interface, knowledge base, decision-making logic, and defuzzification interface.

![Configuration of FLC](image-url)
(1) The fuzzification interface transforms input crisp values into fuzzy values and it involves the following functions.
- receives the input values
- transforms the range of values of input variable into corresponding universe of discourse
- converts input data into suitable linguistic values (fuzzy sets).

This component is necessary when input data are fuzzy sets in the fuzzy inference.

(2) The knowledge base contains a knowledge of the application domain and the control goals. It consists of a data base and a linguistic rule base.
- The data base contains necessary definitions which are used in control rules and data manipulation.
- The linguistic rule base defines the control strategy and goals by means of linguistic control rules.

(3) The decision-making logic performs the following functions
- simulates the human decision-making procedure based on fuzzy concepts
- infers fuzzy control actions employing fuzzy implication and linguistic rules.

(4) The defuzzification interface the functions
- a scale mapping which converts the range of output values into corresponding universe of discourse
- defuzzification which yields a nonfuzzy control action from an inferred fuzzy control action.

In the following sections, the control components will be developed in detail.

10.1.3 Choice of state variables and control variables
Before starting the detailed procedure of the FLC design, we have to choose the variables. A fuzzy control system is designed to control a process, and thus it is needed to determine state variables and control variables of the process. The state variables become input variables of the fuzzy control system, and the control variables become output variables. Selection of the variables depends on expert knowledge on the process. In particular, variables such as state, state error, state error deviation, and state error integral are often used.

10.2 Fuzzification interface component
In the fuzzification component, there are three main issues to be considered: scale mapping of input data, strategy for noise and selection of fuzzification functions.

(1) Scale mapping of input data: We have to decide a strategy to convert the range of values of input variables into corresponding universe of discourse. When an input value is come through a measuring system, the values must be located in the range of input variables. For example, if the range of input variables was normalized between −1 and +1, a procedure is needed which maps the observed input value into the normalized range.

(2) Strategy for noise: When observed data are measured, we may often think that the data were disturbed by random noise. In this case, a fuzzification operator should convert the probabilistic data into fuzzy numbers. In this way, computational efficiency is enhanced since fuzzy numbers are much easier to manipulate than random variables. Otherwise, we assume that the observed data do not contain vagueness, and then we consider the observed data as a fuzzy singleton. A fuzzy singleton is a precise value and hence no fuzziness is introduced by
fuzzification in this case. In control applications, the observed data are usually crisp and used as fuzzy singleton inputs in the fuzzy reasoning.

(3) Selection of fuzzification function: A fuzzification operator has the effect of transforming crisp data into fuzzy sets.

\[ x = \text{fuzzifier}(x_0) \]

Where \( x_0 \) is a observed crisp value and \( x \) is a fuzzy set, and fuzzifier represents a fuzzification operator.

Fig 10.1 shows a fuzzification function which transforms crisp data into a fuzzy singleton value.

![Fuzzification function for fuzzy singleton](image)

Fig 10.1  Fuzzification function for fuzzy singleton

Fig 10.2 shows a fuzzification function transforming a crisp value into a triangular fuzzy number. The peak point of this triangle corresponds to the mean value of a data set, while the base is twice the standard deviation of the data set.

![Fuzzification function for fuzzy triangular number](image)

Fig 10.2  Fuzzification function for fuzzy triangular number
10.3 Knowledge base component

10.3.1 Data base

The knowledge base of an FLC is comprised of two parts: a data base and a fuzzy control rule base. We will discuss some issues relating to the data base in this section and the rule base in the next section.

In the data base part, there are four principal design parameters for an FLC: discretization and normalization of universe of discourse, fuzzy partition of input and output spaces, and membership function of primary fuzzy set.

1) Discretization and normalization of universe of discourse

The modeling of uncertain information with fuzzy sets raises the problem of quantifying such information for digital computers. A universe of discourse in an FLC is either discrete or continuous. If the universe is continuous, a discrete universe may be formed by a discretization procedure. A data set may be also normalized into a certain range of data.

1) Discretization of a universe of discourse: It is often referred to as quantization. The quantization discretizes a universe into a certain number of segments. Each segment is labeled as a generic element and forms a discrete universe. A fuzzy set is then defined on the discrete universe of discourse. The number of quantization levels affects an important influence on the control performance, and thus it should be large enough to give adequate approximation. That number should be determined in considering both the control quality and the memory storage in computer.

For the discretization, we need a scale mapping, which serves to transform measured variables into values in the discretized universe. The mapping can be uniform (linear), nonuniform (nonlinear), or both.

Table 10.1 shows an example of discretization, where a universe of discourse is discretized into 13 levels (-6, -5, -4, …, 0, 1, …, 5, 6).

<table>
<thead>
<tr>
<th>Range</th>
<th>Level No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x \leq -2.4 )</td>
<td>-6</td>
</tr>
<tr>
<td>(-2.4 &lt; x \leq -2.0 )</td>
<td>-5</td>
</tr>
<tr>
<td>(-1.6 &lt; x \leq -0.8 )</td>
<td>-4</td>
</tr>
<tr>
<td>(-0.8 &lt; x \leq -0.4 )</td>
<td>-3</td>
</tr>
<tr>
<td>(-0.4 &lt; x \leq -0.2 )</td>
<td>-2</td>
</tr>
<tr>
<td>(-0.2 &lt; x \leq -0.1 )</td>
<td>-1</td>
</tr>
<tr>
<td>(-0.1 &lt; x \leq +0.1 )</td>
<td>0</td>
</tr>
<tr>
<td>(+0.1 &lt; x \leq +0.2 )</td>
<td>1</td>
</tr>
<tr>
<td>(+0.2 &lt; x \leq +0.4 )</td>
<td>2</td>
</tr>
<tr>
<td>(+0.4 &lt; x \leq +0.8 )</td>
<td>3</td>
</tr>
<tr>
<td>(+0.8 &lt; x \leq +1.1 )</td>
<td>4</td>
</tr>
</tbody>
</table>
2) Normalization of a universe of discourse: It is a discretization into a normalized universe. The normalized universe consists of finite number of segments. The scale mapping can be uniform, nonuniform, or both. Table 10.2 shows an example, where the universe of discourse [-6.9, +4.5] is transformed into the normalized closed interval [-1, 1].

<table>
<thead>
<tr>
<th>Range</th>
<th>Normalized segments</th>
<th>Normalized universe</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-6.9, -4.1]</td>
<td>[-1.0, -0.5]</td>
<td>[-1.0, +1.0]</td>
</tr>
<tr>
<td>[-4.1, -2.2]</td>
<td>[-0.5, -0.3]</td>
<td></td>
</tr>
<tr>
<td>[-2.2, 0.0]</td>
<td>[-0.3, 0.0]</td>
<td></td>
</tr>
<tr>
<td>[-0.0, +1.0]</td>
<td>[0.0, +0.2]</td>
<td></td>
</tr>
<tr>
<td>[+1.0, +2.5]</td>
<td>[+0.2, +0.6]</td>
<td></td>
</tr>
<tr>
<td>[+2.5, +4.5]</td>
<td>[+0.6, +1.0]</td>
<td></td>
</tr>
</tbody>
</table>

(2) Fuzzy partition of input and output spaces

A linguistic variable in the antecedent of a rule forms a fuzzy input space, while that in the consequent of the rule forms a fuzzy output space. In general, a linguistic variable is associated with a term set. A fuzzy partition of the space determines how many terms should exist in a term set. This is the same problem to find the number of primary fuzzy sets (linguistic terms).

There are seven linguistic terms often used in the fuzzy inference:

NB: negative big
NM: negative medium
NS: negative small
ZE: zero
PS: positive small
PM: positive medium
PB: positive big

A typical example is given in Fig 10.3 representing two fuzzy partition in the same normalized universe [-1, +1]. Membership functions with triangle and trapezoid shapes are used here.
Fig 10.3 Example of fuzzy partition with linguistic terms

The number of fuzzy terms in a input space determines the maximum number of fuzzy control rules. Suppose a fuzzy control system with two input and one output variables. If the input variables have 5 and 4 terms, the maximum number of control rules that we can construct is 20 ($5 \times 4$) as shown in Fig 10.4. Fig 10.5 shows an example of system having 3 fuzzy rules.

Fig 10.4 A fuzzy partition in 2-dimension input space
A fuzzy control system could always infer a proper control action for every state of process. This property is concerned with the supports on which primary fuzzy sets are defined. The union of these supports should cover the related universe of discourse in relation to some level. For example, in Fig 10.3, any input value is included to at least one linguistic term with membership value greater than 0.5.

(3) Membership function of primary fuzzy set

There are various types of membership functions such as triangular, trapezoid, and bell shapes. Table 10.3 shows an example defining triangular membership functions on the discretized universe of discourse in Table 10.1. For example, term NM is defined such as:

\[
\begin{align*}
\mu_{NM}(-6) &= 0.3 \\
\mu_{NM}(-5) &= 0.7 \\
\mu_{NM}(-4) &= 1.0 \\
\mu_{NM}(-3) &= 0.7 \\
\mu_{NM}(-2) &= 0.3
\end{align*}
\]

Table 10.3 Definition of triangular membership function

<table>
<thead>
<tr>
<th>Level No.</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>−6</td>
<td>1.0</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>−5</td>
<td>0.7</td>
<td>0.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>−4</td>
<td>0.3</td>
<td>1.0</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>−3</td>
<td>0.0</td>
<td>0.7</td>
<td>0.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>−2</td>
<td>0.0</td>
<td>0.3</td>
<td>1.0</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>−1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>0.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.3</td>
<td>1.0</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
An example of bell shaped membership function is given in Table 10.4 and Fig 10.6, where fuzzy sets are defined on the normalized universe of discourse \([-1, +1]\) given in Table 10.2. They have the shapes of function of parameter mean \(m_f\) and standard deviation \(\sigma_f\).

\[
\mu_f(x) = \exp\left\{ -\frac{(x - m_f)^2}{2\sigma_f^2} \right\}
\]

Table 10.4 Definition of bell-shaped membership function

<table>
<thead>
<tr>
<th>Normalized universe</th>
<th>Normalized segments</th>
<th>(m_f)</th>
<th>(\sigma_f)</th>
<th>Fuzzy sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>([-1.0, +1.0])</td>
<td>([-1.0, -0.5]]</td>
<td>-1.0</td>
<td>0.4</td>
<td>NB</td>
</tr>
<tr>
<td></td>
<td>([-0.5, -0.3]]</td>
<td>-0.5</td>
<td>0.2</td>
<td>NM</td>
</tr>
<tr>
<td></td>
<td>([-0.3, -0.0]]</td>
<td>-0.2</td>
<td>0.2</td>
<td>NM</td>
</tr>
<tr>
<td></td>
<td>([-0.0, +0.2]]</td>
<td>0.0</td>
<td>0.2</td>
<td>ZE</td>
</tr>
<tr>
<td></td>
<td>([+0.2, +0.6]]</td>
<td>0.2</td>
<td>0.2</td>
<td>PS</td>
</tr>
<tr>
<td></td>
<td>([+0.6, +0.8]]</td>
<td>0.5</td>
<td>0.2</td>
<td>PM</td>
</tr>
<tr>
<td></td>
<td>([+0.8, +1.0]]</td>
<td>1.0</td>
<td>0.4</td>
<td>PB</td>
</tr>
</tbody>
</table>

Fig 10.6 Example of bell-shaped membership function
10.3.2 Rule base
A fuzzy system is characterized by a set of linguistic statements usually represented by in the form of “if-then” rules. In this section, we examine several topics related to fuzzy control rules.

1) Source of fuzzy control rules

There are two principal approaches to the derivation of fuzzy control rules. The first is a heuristic method in which rules are formed by analyzing the behavior of a controlled process. The derivation relies on the qualitative knowledge of process behavior. The second approach is basically a deterministic method which can systematically determine the linguistic structure of rules.

We can use four modes of derivation of fuzzy control rules. These four modes are not mutually exclusive, and it is necessary to combine them to obtain an effective system.

- Expert experience and control engineering knowledge: operating manual and questionnaire.
- Based on operators’ control actions: observation of human controller’s actions in terms of input-output operating data.
- Based on the fuzzy model of a process: linguistic description of the dynamic characteristics of a process.
- Based on learning: ability to modify control rules such as self-organizing controller.

2) Types of fuzzy control rules

There are two types of control rules: state evaluation control rules and object evaluation fuzzy control rules.

(1) State evaluation fuzzy control rules: State variables are in the antecedent part of rules and control variables are in the consequent part. In the case of MISO (multiple input single output), they are characterized as a collection of rules of the form.

\[ R_1: \text{if } x \text{ is } A_1, \ldots \text{ and } y \text{ is } B_1 \text{ then } z \text{ is } C_1 \]
\[ R_2: \text{if } x \text{ is } A_2, \ldots \text{ and } y \text{ is } B_2 \text{ then } z \text{ is } C_2 \]
\[ \vdots \]
\[ R_n: \text{if } x \text{ is } A_n, \ldots \text{ and } y \text{ is } B_n \text{ then } z \text{ is } C_n \]

where \( x, \ldots, y \) and \( z \) are linguistic variables representing the process state variable and the control variable. \( A_i, \ldots, B_i, \text{ and } C_i \) are linguistic values of the variables \( x, \ldots, y \) and \( z \) in the universe of discourse \( U, \ldots, V \) and \( W \), respectively \( i = 1, 2, \ldots, n \). That is,

\[ x \in U, A_i \subset U \]
\[ \ldots \]
\[ y \in V, B_i \subset V \]
\[ z \in W, C_i \subset W \]

In a more general version, the consequent part is represented as a function of the state variable \( x, \ldots, y \).

\[ R_i: \text{if } x \text{ is } A_i, \ldots \text{ and } y \text{ is } B_i \text{ then } z = f(x, \ldots, y) \]

The state evaluation rules evaluate the process state (e.g. state, state error, change of error) at
time $t$ and compute a fuzzy control action at time $t$.

In the previous section concerned with the fuzzy partition of input space, we said that the maximum number of control rules is defined by the partition. In the input variable space, the combination of input linguistic term may give a fuzzy rule. When there is a set of fuzzy rules as follows

$$R_i: \text{if } x \text{ is } A_i \text{ and } y \text{ is } B_i \text{ then } z \text{ is } C_i$$

$i = 1, 2, \ldots, n$

The rules can be represented as the form of table in Fig 10.7.

![Fig 10.7 Fuzzy rules represented by a rule table](image)

(2) Object evaluation fuzzy control rules: It is also called predictive fuzzy control. They predict present and future control actions, and evaluate control objectives. A typical rule is described as

$$R_1: \text{if } (z \text{ is } C_1 \rightarrow (x \text{ is } A_1 \text{ and } y \text{ is } B_1)) \text{ then } z \text{ is } C_1.$$  

$$R_2: \text{if } (z \text{ is } C_2 \rightarrow (x \text{ is } A_2 \text{ and } y \text{ is } B_2)) \text{ then } z \text{ is } C_2.$$  

$$\ldots$$

$$R_n: \text{if } (z \text{ is } C_n \rightarrow (x \text{ is } A_n \text{ and } y \text{ is } B_n)) \text{ then } z \text{ is } C_n.$$  

A control action is determined by an objective evaluation that satisfies the desired states and objectives. $x$ and $y$ are performance indices for the evaluation and $z$ is control command. $A_i$ and $B_i$ are fuzzy values such as NM and PS. The most likely control rule is selected through predicting the results $(x, y)$ corresponding to every control command $C_i$, $i = 1, 2, \ldots, n$.

In linguistic terms, the rule is interpreted as: if the performance index $x$ is $A_i$ and index $y$ is $B_i$ when a control command $z_i$ is $C_i$, then this rule is selected, and the control command $C_i$ is taken to be the output of the controller.

**10.4 Inference (Decision making logic)**

In general, in decision making logic part, we use four inference methods described in the previous
chapter: Mandani method, Larsen method, Tsukamoto method, and TSK method.

10.4.1 Mandani method

This method uses minimum operator as a fuzzy implication operator, and max-min operator for the composition as shown in section 9.4. Suppose fuzzy rules are given in the following form.

\[ R_i: \text{if } x \text{ is } A_i \text{ and } y \text{ is } B_i \text{ then } z \text{ is } C_i, \]

\[ i = 1, 2, \ldots, n \]

\[ x \in U, A_i \subseteq U \]

\[ y \in V, B_i \subseteq V \]

\[ z \in W, C_i \subseteq W \]

1) When input data are singleton such as \( x = x_0 \) and \( y = y_0 \) (in this case, the input data are not fuzzified.), the matching degrees (firing strength) of \( A_i \) and \( B_i \) are \( \mu_{A_i}(x_0) \) and \( \mu_{B_i}(y_0) \), respectively. Therefore the matching degree of rule \( R_i \) is

\[ \alpha_i = \mu_{A_i}(x_0) \land \mu_{B_i}(y_0) \]

Then \( \mu_{C'_i}(z) = \alpha_i \land \mu_{C_i}(z) \) where \( C'_i \) is the result of rule \( R_i \). The aggregated result \( C' \) derived from individual control rules is defined as follows:

\[ \mu_{C'}(z) = \bigvee_{i=1}^{n} [\alpha_i \land \mu_{C_i}(z)] \]

\[ C' = \bigcup_{i=1}^{n} C'_i \]

2) When input data are fuzzy sets, \( A' \) and \( B' \)

\[ \alpha_i = \min\{\max(\mu_{A_i}(x) \land \mu_{A}(x)), \max(\mu_{B_i}(y) \land \mu_{B}(y))\} \]

\[ \mu_{C'_i}(z) = \alpha_i \land \mu_{C_i}(z) \]

\[ i = 1, 2, \ldots, n \]

The aggregate result \( C' \) is defined by

\[ \mu_{C'}(z) = \bigvee_{i=1}^{n} [\alpha_i \land \mu_{C_i}(z)] \]

\[ C' = \bigcup_{i=1}^{n} C'_i \]

10.4.2 Larsen method

This method uses the product operator (\( \cdot \)) for the fuzzy implication, and the max-product operator for the composition. Suppose fuzzy rules are given in the following form.

\[ R_i: \text{if } x \text{ is } A_i \text{ and } y \text{ is } B_i \text{ then } z \text{ is } C_i, \]

\[ i = 1, 2, \ldots, n \]

1) When input data are singleton, \( x = x_0 \) and \( y = y_0 \). The matching degrees is
\[ \alpha_i = \mu_{A_i}(x_i) \land \mu_{B_i}(y_i) \]

The result \( C'_i \) of rule \( R_i \) is defined by
\[ \mu_{C'_i}(z) = \alpha_i \cdot \mu_{C_i}(z) \]

The aggregated result \( C' \) is
\[ \mu_{C'}(z) = \bigvee_{i=1}^{n} [\alpha_i \cdot \mu_{C_i}(z)] \]

or
\[ C' = U_{i=1}^{n} C'_i \]

2) When input data are given as the form of fuzzy sets, \( A' \) and \( B' \), we have matching degrees as
\[ \alpha_i = \min[\max(\mu_{A_i}(x) \land \mu_{A_i}(x)), \max(\mu_{B_i}(y) \land \mu_{B_i}(y))] \]

The result \( C'_i \) of rule \( R_i \) is defined by
\[ \mu_{C'_i}(z) = \alpha_i \cdot \mu_{C_i}(z) \]

The aggregate result \( C' \) is
\[ \mu_{C'}(z) = \bigvee_{i=1}^{n} [\alpha_i \land \mu_{C_i}(z)] \]

or
\[ C' = U_{i=1}^{n} C'_i \]

**10.4.3 Tsukamoto method**

This method is used when the consequent part of each rule is represented by fuzzy set with a monotonic membership function. The inferred output of each rule is defined as a crisp value induced by the rule’s matching degree (firing strength).

We suppose fuzzy rules are given in the following form and the set \( C_i \) has a monotonic membership function \( \mu_{C_i}(z) \)

\[ R_i: \text{if } x \text{ is } A_i \text{ and } y \text{ is } B_i \text{ then } z \text{ is } C_i, \]

\[ i = 1, 2, \ldots, n \]

The matching degree \( \alpha \) of each rule is defined like in the previous methods in the cases of both singleton input and fuzzy set input.

The result of \( z_i \) rule \( R_i \) is obtained by (Fig 10.8)
\[ z_i = \mu_{C_i}^{-1}(\alpha_i) \]
The aggregated result $z'$ is taken as the weighted average of each rule’s output

$$z' = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2}$$

This method gives a crisp value as an aggregated result and thus there is no need to defuzzify it.

If $x_1$, $x_2$ then $y$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$:</td>
<td>5 7</td>
<td>1.0x_1 + 0.5x_2 + 1.0</td>
</tr>
<tr>
<td>$R_2$:</td>
<td>3 9</td>
<td>-0.1x_1 + 4.0x_2 + 1.2</td>
</tr>
<tr>
<td>$R_3$:</td>
<td>3 9 11 18</td>
<td>0.9x_1 + 0.7x_2 + 9.0</td>
</tr>
<tr>
<td>$R_4$:</td>
<td>11 18</td>
<td>0.2x_1 + 0.1x_2 + 0.2</td>
</tr>
</tbody>
</table>

Fig 10.8  Example of Tsukamoto control rules

Fig 10.9  Example of TSK fuzzy control rules
10.4.4 TSK method
This method is used when the consequent part is given as a function of input variables.

\[ R_i: \text{if } x \text{ is } A_i \text{ and } y \text{ is } B_i \text{ then } z \text{ is } f_i(x, y) \]

Where \( z = f(x, y) \) is a crisp function of input variables \( x \) and \( y \). Usually \( f(x, y) \) has a polynomial form (Fig 10.9).

Suppose input data are singleton \( x_0 \) and \( y_0 \), then the inferred result of rule \( R_i \) is \( f_i(x_0, y_0) \). The matching degree \( \alpha_i \) of \( R_i \) is same with the previous one. Therefore the final aggregated result \( z' \) is the weighted average using the matching degree \( \alpha_i \):

\[ z' = \frac{\alpha_i f_1(x_0, y_0) + \alpha_2 f_2(x_0, y_0)}{\alpha_1 + \alpha_2} \]

The final result is a crisp value and thus there is no need to defuzzify it.

10.5 Defuzzification
In many practical applications, a control command is given as a crisp value. Therefore it is needed to defuzzify the result of the fuzzy inference. A defuzzification is a process to get a non-fuzzy control action that best represents the possibility distribution of an inferred fuzzy control action. Unfortunately, we have no systematic procedure for choosing a good defuzzification strategy, and thus we have to select one in considering the properties of application case. The three commonly used strategies are described in this section.

10.5.1 Mean of maximum method (MOM)
The MOM strategy generates a control action which represents the mean value of all control actions, whose membership functions reach the maximum (Fig 10.10). In the case of a discrete universe, the control action may be expressed as

\[ z_0 = \frac{1}{k} \sum_{j=1}^{k} z_j \]

\( z_j \): control action whose membership functions reach the maximum.

\( k \): number of such control actions.

![Mean of maximum (MOM)](Fig 10.10)

10.5.2 Center of area method (COA)
The widely used COA strategy generates the center of gravity of the possibility distribution of a fuzzy set \( C \) (Fig 10.11). In the case of a discrete universe, thus method gives
\[ z_0 = \frac{\sum_{j=1}^{n} \mu_c(z_j) \cdot z_j}{\sum_{j=1}^{n} \mu_c(z_j)} \]

Where \( n \) is the number of quantization levels of the output, \( C \) is a fuzzy set defined on the output dimension \( (z) \).

![Center of area (COA)](image1)

**10.5.3 Bisector of area (BOA)**

The BOA generates the action \((z_0)\) which partitions the area into two regions with the same area (Fig 10.12).

\[
\int_{\alpha}^{\beta} \mu_c(z) \, dz = \int_{\alpha}^{\beta} \mu_c(z) \, dz
\]

where \( \alpha = \min\{z \mid z \in W\} \)

\( \beta = \max\{z \mid z \in W\} \)

![Bisector of area (BOA)](image2)
10.5.4 Lookup table

Even with the many advantages, it is pointed out that the FLC has the problem of time complexity. It takes much time to compute the fuzzy inference and defuzzification. Therefore a lookup table is often used which simply shows relationships between input variables and control output actions. But the lookup table can be constructed after making the FLC and identifying the relationships between the input and output variables. In general, it is extremely difficult to get an acceptable lookup table of a nonlinear control system without constructing a corresponding FLC.

Example 10.1 Table 10.5 shows an example of lookup table for the two input variables error (e) and change of error (ce), and control variable (v). The variables are all discretized and normalized in the range \([-1, +1]\). For example, when \(e = -1.0\) and \(ce = -0.5\), we can obtain \(v = -0.5\) by using the lookup table instead of by executing the full fuzzy controller.

Table 10.5 Example of lookup table

<table>
<thead>
<tr>
<th>ce</th>
<th>-1.0</th>
<th>-0.5</th>
<th>0</th>
<th>0.5</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>-1.0</td>
<td>-0.5</td>
<td>-0.5</td>
<td>-0.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>-0.5</td>
<td>-1.0</td>
<td>-1.0</td>
<td>0.5</td>
<td>-0.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>0</td>
<td>-1.0</td>
<td>-1.0</td>
<td>0.5</td>
<td>-0.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

10.6 Design procedure of fuzzy logic controller

When we decided to design a fuzzy logic controller, we can follow the following design procedure

1) Determination of state variables and control variables

In general, the control variable is determined depending on the property of process to be controlled. But we have to select the state variables. In general, state, state error and error difference are often used. The state variables are input variables, and the control variables are output of our controller to be developed.

2) Determination of inference method

We select one method among four inference methods described in the previous section. The decision is dependent upon the properties of process to be studied.

3) Determination of fuzzification method

It is necessary to study the property of measured data of state variables. If there is uncertainty in the data, the fuzzification is necessary, and we have to select a fuzzification method and membership functions of fuzzy sets. If there is no uncertainty, we can use singleton state variables.

4) Discretization and normalization of state variable space

In general, it is useful to use discretized and normalized universe of discourse. We have to decide whether it is necessary and how we can do.

5) Partition of variable space.
The state variables are input variables of our controller and thus the partition is important for the structure of fuzzy rules. At this step, partition of control space (output space of the controller) is also necessary.

6) Determination of the shapes of fuzzy sets

It is necessary to determine the shapes of fuzzy sets and their membership functions for the partitioned input spaces and output spaces.

7) Construction of fuzzy rule base

Now, we can build control rules. We determined the variables and corresponding linguistic terms in antecedent part and consequent part of each rule. The architecture of rules is dependent upon the inference method determined in step 2).

8) Determination of defuzzification strategy

In general, we use singleton control values and thus we have to determine the method.

9) Test and tuning

It is almost impossible to obtain a satisfactory fuzzy controller without tuning. In general it is necessary to verify the controller and tune it until when we get satisfactory results.

10) Construction of lookup table

If the controller shows satisfactory performance, we have to decide whether we use a lookup table instead of using the inference system. The lookup table is often used to save computing the time of the inference and defuzzification. The lookup table shows the relationships between a combination of input variables and control actions.

10.7 Application example of FLC design

Servomotors are used in many automatic system including drivers for printers, floppy disks, tape recorders, and robot manipulations. The control of such servomotors is an important issue. The servomotor process shows nonlinear properties, and thus we apply the fuzzy logic control to the motor control. The task of the control is to rotate the shaft of the motor to a set point without overshoot. The set point and process output in measured in degree.

1) Determination of state variables and control variable

   (1) State variables (input variable of controller):
   - Error equals the set point minus the process output (e).
   - Change of error (ce) equals the error from the process output minus the error from the last process output.

   (2) Control variable (output variable of the controller):
   - Control input (v) equals the voltage applied to the process.

2) Determination of inference method

   The Mandani inference method is selected because it is simple to explain.

3) Determination of fuzzification method

   We can measure the state variables without uncertainty and thus we use the measured singleton for the fuzzy inference.
4) Discretization and normalization

The shaft encoder of the motor has a resolution of 1000. The universes of discourse are as follows:

\[-1000 \leq e \leq 1000\]
\[-100 \leq ce \leq 100\]

The servo amplifier has an output range of 30 V and thus the control variables (v) are in the range

\[-30 \leq v \leq 30\]

We discretize and normalize the input variables in the range $[-1, +1]$ as shown in Table 10.6. The control variable $v$ is normalized in the range $[-1, +1]$ with the equation.

$$v' = \frac{1}{30} v$$

<table>
<thead>
<tr>
<th>error (e)</th>
<th>error change (ce)</th>
<th>quantized level</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-1000 \leq e \leq -800$</td>
<td>$-100 \leq ce \leq -80$</td>
<td>$-1.0$</td>
</tr>
<tr>
<td>$-800 &lt; e \leq -600$</td>
<td>$-80 &lt; ce \leq -60$</td>
<td>$-0.8$</td>
</tr>
<tr>
<td>$-600 &lt; e \leq -400$</td>
<td>$-60 &lt; ce \leq -40$</td>
<td>$-0.6$</td>
</tr>
<tr>
<td>$-400 \leq e \leq -200$</td>
<td>$-40 \leq ce \leq -20$</td>
<td>$-0.4$</td>
</tr>
<tr>
<td>$-200 &lt; e \leq -100$</td>
<td>$-20 \leq ce \leq -10$</td>
<td>$-0.2$</td>
</tr>
<tr>
<td>$-100 &lt; e \leq 100$</td>
<td>$-10 \leq ce \leq 10$</td>
<td>$0$</td>
</tr>
<tr>
<td>$100 &lt; e \leq 200$</td>
<td>$10 \leq ce \leq 20$</td>
<td>$0.2$</td>
</tr>
<tr>
<td>$200 &lt; e \leq 400$</td>
<td>$20 \leq ce \leq 40$</td>
<td>$0.4$</td>
</tr>
<tr>
<td>$400 &lt; e \leq 600$</td>
<td>$40 \leq ce \leq 60$</td>
<td>$0.6$</td>
</tr>
<tr>
<td>$600 &lt; e \leq 800$</td>
<td>$60 \leq ce \leq 80$</td>
<td>$0.8$</td>
</tr>
<tr>
<td>$800 &lt; e \leq 1000$</td>
<td>$80 \leq ce \leq 100$</td>
<td>$1.0$</td>
</tr>
</tbody>
</table>

5) Partition of input space and output space

We partition space of each input and output variable into seven regions, and each region is associated with linguistic term as shown in Fig 10.13. Now we know the maximum number of possible fuzzy rules is 49.
6) Determination of the shapes of fuzzy sets

We normalized the input and output variables on the same interval $[-1, +1]$ and partitioned the region into seven subregions, and thus we define the primary triangular fuzzy sets for all variables as shown in Table 10.7 and Fig 10.14.

Table 10.7 Definition of primary fuzzy sets

<table>
<thead>
<tr>
<th>Level</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-1$</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$-0.8$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$-0.6$</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$-0.4$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$-0.2$</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>0.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>0.8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Fig 10.14 Graphical representation of primary fuzzy sets
7) Construction of fuzzy rules

We interviewed with an expert of the servomotor control, and we collect knowledge such as:

If the error is zero and the error change is positive small, then the control input is negative small.

This type of rules are rewritten in the following form

(1) If \( e \) is PB and \( ce \) is any, then \( v \) is PB.
(2) If \( e \) is PM and \( ce \) is NB, NM, or NS, then \( v \) is PS.
(3) If \( e \) is ZE and \( ce \) is ZE, PS, or PM, then \( v \) is ZE.
(4) If \( e \) is PS and \( ce \) is NS, ZE, or PS, then \( v \) is ZE.
(5) If \( e \) is NS and \( ce \) is NS, ZE, PS, or PM, then \( v \) is NS
(6) If \( e \) is NS or ZE and \( ce \) is PB, then \( v \) is PS.

The full set of fuzzy rules is summarized in the rule table in Fig 10.15

<table>
<thead>
<tr>
<th>( e )</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td></td>
<td></td>
<td></td>
<td>NB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NM</td>
<td>NM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NS</td>
<td></td>
<td>NS</td>
<td></td>
<td>PS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZE</td>
<td>NS</td>
<td></td>
<td>ZE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PS</td>
<td></td>
<td>PS</td>
<td></td>
<td></td>
<td>PM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB</td>
<td>PB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig 10.15  Fuzzy rule table

8) Determination of defuzzification strategy

We take the COA (center of area) method because it is most commonly used.

9) Test and tuning

We checked the performance of the developed controller and refined some fuzzy rules.

10) Construction of lookup table
After verifying the controller showing good performance, we decided to use a lookup table. We extended the inference for every combination of discretized input variables $c$ and $ce$. For example,

for $c = -0.2$ and $ce = 0$, $v$ is $-0.4$

for $c = -0.4$ and $ce = 0.4$, $v$ is $0.2$

The corresponding lookup table is given in Table 10.8. Now we can use this lookup table in order to save the inference time and defuzzification time.

Table 10.8 Lookup table

<table>
<thead>
<tr>
<th>$c$</th>
<th>$ce$</th>
<th>1.0</th>
<th>0.8</th>
<th>0.6</th>
<th>0.4</th>
<th>0.2</th>
<th>0.0</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.0</td>
<td>-1.0</td>
<td>-1.0</td>
<td>-1.0</td>
<td>-1.0</td>
<td>-0.8</td>
<td>-0.8</td>
<td>-0.8</td>
<td>-0.8</td>
<td>-0.8</td>
<td>-0.8</td>
<td>-0.6</td>
<td>-0.6</td>
</tr>
<tr>
<td>-0.8</td>
<td>-0.8</td>
<td>-0.8</td>
<td>-0.8</td>
<td>-0.8</td>
<td>-0.8</td>
<td>-0.8</td>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.6</td>
</tr>
<tr>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
</tr>
<tr>
<td>-0.4</td>
<td>-0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>-0.2</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
</tr>
<tr>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>0.8</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>1.0</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
</tbody>
</table>

10.8 Fuzzy expert systems

An expert system is a program which contains human expert’s knowledge and gives answers to the user’s query by using an inference method. The knowledge is often stored in the form of rule base, and the most popular form is that of “if-then”.

A fuzzy expert system is an expert system which can deal uncertain and fuzzy information. In our real world, a human expert has his knowledge in the form of linguistic terms. Therefore it is natural to represent the knowledge by fuzzy rules and thus to use fuzzy inference methods.

The structure of a fuzzy expert system is similar to that of the fuzzy logic controller. It’s configuration is shown in Fig 10.16. As in the fuzzy logic controller, there can be fuzzification interface, knowledge base, and inference engine (decision making logic). Instead of the defuzzification module, there is the linguistic approximation module.

10.8.1 fuzzification interface

This module deals user’s request, and thus we have to determine the fuzzification strategy. If we want to make the fuzzy expert system receive linguistic terms, this module has to have an ability to
handle such fuzzy information. The fuzzification strategy, if necessary, is similar to that of the fuzzy logic controller.

Contrary to the fuzzy logic controller, it is not needed to consider the discretization or normalization. But the fuzzy partition and assigning fuzzy linguistic terms to each subregion are necessary.

The expert’s knowledge may be represented in the form of “if-then” by using fuzzy linguistic terms. Each rule can have its certainty factor which represents the certainty level of the rule. This certainty factor is used in the aggregation of the results from each rule.

10.8.3 Inference engine (Decision making logic)

The fuzzy expert systems can use the inference methods of the fuzzy logic controller. The system does not deal with a machine or process, and thus it is difficult to have a fuzzy set with monotonic membership function in the consequent part of a rule. Therefore especially, Mamdani method and Larsen method are often used.

10.8.4 Linguistic approximation

As we stated before, a fuzzy expert system does not control a machine nor a process, and thus, in general, the defuzzification is not necessary. Instead of the defuzzification module, sometimes we need a linguistic approximation module.

This module finds a linguistic term which is closest to the obtained fuzzy set. To do it, we may use a measuring technique of distance between fuzzy sets.

10.8.5 Scheduler

This module controls all the processes in the fuzzy expert system. It determines the rules to be executed and sequence of their executions. It may also provide an explanation function for the result. For example, it can show the reason how the result was obtained.

Fig 10.16   Configuration of fuzzy expert system
[Summary]

- The fuzzy logic controller (FLC) is good when
  - the control process is too complex
  - the information is qualitative
- Advantage of the fuzzy logic controller
  - Parallel and distributed control
  - Linguistic control
  - Robust control
- Components of the FLC
  - Fuzzification interface
  - Knowledge base
  - Decision-making logic
  - Defuzzification interface
- Fuzzification interface components
  - Scale mapping of input data
  - Strategy of noise
  - Selection of fuzzification function
- Data base components
  - Discretization of universe of discourse
  - Normalization of universe of discourse
  - Fuzzy partition of input and output spaces
  - Membership function of primary fuzzy set
- Rule base
  - Choice of state variables and control variables
  - Source of fuzzy control rules
  - Type of fuzzy control rules: state evaluation rules and object evaluation rules.
- Decision making logic
  - Mamdani inference method
  - Larsen inference method
  - Tsukamoto inference method
  - TSK inference method
- Defuzzification methods
  - Mean of maximum method (MOM)
  - Center of area method (COA)
  - Bisector of area (BOA)

- Lookup table
  - Disadvantage of the FLC is the high time complexity.
  - Saving of computation time
  - Direct relationship between input variable and control output actions

- Design procedure of the FLC
  - Determination of variables
  - Determination of inference method
  - Determination of fuzzification method
  - Discretization and normalization of variables
  - Partition of space
  - Determination of fuzzy sets
  - Construction of fuzzy rule base
  - Determination of defuzzification strategy
  - Test and tuning
  - Lookup table
10.1 Describe advantage of the fuzzy logic controller (FLC).

10.2 In which case the FLC is superior to the conventional control algorithm?

10.3 Explain the followings:
- state variable
- control output
- control input
- control variable

10.4 Explain the following components in the FLC.
- fuzzification interface
- knowledge base
- data base
- decision-making logic
- defuzzification interface

10.5 Explain the three main issues in the fuzzification interface component.

10.6 Explain the three main issues in the data base.

10.7 What is the difference between the discretization and normalization of universe of discourse.

10.8 Explain the relationship between the fuzzy partition of input variable and the number of fuzzy rules.

10.9 Why and how membership functions can be defined in a table when the universe of discourse is discretized?

10.10 What is the objective of the normalization of universe of discourse?
10.11 What are the main criteria to determine the state variables and control variables?

10.12 Explain the two-types of control rules:
   - State evaluation fuzzy control rules
     \[ R_i: \text{if } x \text{ is } A_i, \ldots \text{ and } y \text{ is } B_i \text{ then } z \text{ is } C_i \]
   - Object evaluation fuzzy control rules
     \[ R_i: \text{if } (z \text{ is } C_i \rightarrow (x \text{ is } A_i \text{ and } y \text{ is } B_i)) \text{ then } z \text{ is } C_i \]

10.13 Explain the four inference methods:
   - Mamdani method
   - Larsen method
   - Tsukamoto method
   - TSK method

10.14 What is the property of the membership function in consequent part in Tsukamoto method?

10.15 Explain the three defuzzification methods:
   - Mean of maximum method
   - Center of area method
   - Bisector of area

10.16 What is the lookup table? Why is it often used?

10.17 Show the design procedure of the FLC.