ICS 25.040.40
N 18

中华人民共和国国家标准

GB/T 15969.7—2008/IEC 61131-7:2000
代替 GB/T 17165.3—2001

可编程序控制器
第7部分:模糊控制编程

Programmable controllers—Part 7:Fuzzy control programming

(IEC 61131-7:2000,IDT)

2008-06-30 发布　　2009-01-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目　次

前言 …… Ⅲ
1　范围 …… 1
2　规范性引用文件 …… 1
3　术语和定义 …… 1
4　模糊控制在可编程序控制器中的集成 …… 3
5　模糊控制语言(FCL) …… 3
5.1　模糊控制程序的交换 …… 3
5.2　模糊控制语言元素 …… 4
5.3　模糊控制语言(FCL)示例 …… 11
5.4　模糊控制语言(FCL)的产生式规则和关键字 …… 12
6　相符性 …… 15
6.1　模糊控制语言(FCL)的相符性等级 …… 15
6.2　数据检查清单 …… 17
附录A(资料性附录)　理论 …… 18
附录B(资料性附录)　示例 …… 29
附录C(资料性附录)　工业实例:集装箱吊车 …… 31
附录D(资料性附录)　在规则块中使用变量的例子 …… 39
附录E(资料性附录)　符号、缩写、同义词 …… 42

图1　用功能块图FBD表示的模糊控制功能块示例 …… 3
图2　模糊控制语言(FCL)程序的数据交换 …… 4
图3　使用ST语言和FBD语言的功能块接口说明示例 …… 5
图4　上(下)界型示例 …… 6
图5　隶属函数的变量用法示例 …… 6
图6　单点集语言项的示例 …… 7
图7　模糊功能块示例 …… 11
图8　相符性等级 …… 15
图A.1　语言项“法定年龄”和“成年人”的隶属函数 …… 18
图A.2　用语言形描述语言变量“年纪”和其在时间尺度(年龄)上的对应关系 …… 19
图A.3　常用的隶属函数的形状 …… 19
图A.4　两个隶属函数之间运算的算法 …… 21
图A.5　模糊控制的结构和功能元素 …… 22
图A.6　模糊化原理(示例) …… 22
图A.7　用控制规则形式表示的知识库 …… 22
图A.8　两个变量的矩阵表示 …… 23
图A.9　推理元素 …… 23
图A.10a)　聚集原理(举例说明) …… 24
图A.10b)　激活原理(举例说明) …… 24
图A.10c)　综合原理(举例说明) …… 25
图A.11a)　清晰化方法 …… 25

图 A.11b) 左取大和右取大的区别 …… 26
图 A.11c) 面积中心线和重心的区别 …… 26
图 A.11d) 清晰化方法 …… 27
图 A.12a) 模糊控制器:基本结构 …… 27
图 A.12b) 模糊控制器示例 …… 28
图 B.1 预先控制示例 …… 29
图 B.2 参数自适应调整举例 …… 29
图 B.3 直接模糊控制示例 …… 30
图 C.1 工业实例:集装箱吊车 …… 31
图 C.2 吊车头与目的地之间的“距离”语言变量 …… 31
图 C.3 集装箱和吊车头形成的“角度”语言变量 …… 32
图 C.4 “功率”语言变量 …… 32
图 C.5 规则库 …… 32
图 C.6 “距离”语言变量的模糊化 …… 33
图 C.7 “角度”语言变量的模糊化 …… 33
图 C.8 三个规则的子集 …… 34
图 C.9 聚集的元素 …… 34
图 C.10 聚集原理 …… 34
图 C.11 激活的元素 …… 35
图 C.12 激活原理 …… 35
图 C.13 综合的元素 …… 36
图 C.14 综合原理 …… 36
图 C.15 清晰化 …… 37
图 C.16 FCL 应用实例 …… 37
图 D.1 受控系统原理 …… 39
图 D.2 烤炉的模糊控制原理 …… 39
图 D.3 规则块 …… 39
图 D.4 FCL 示例 …… 40

表 1 清晰化方法 …… 7
表 2 清晰化方法的公式 …… 7
表 3 OR 和 AND 对偶算法 …… 8
表 4 激活方法 …… 9
表 5 综合方法 …… 9
表 6 运算符优先级 …… 10
表 7 保留关键字 …… 14
表 8 FCL 基本级语言元素(必备) …… 16
表 9 FCL 扩展级语言元素(可选) …… 16
表 10 开放级语言元素清单示例 …… 17
表 11 数据检查清单 …… 17
表 A.1 推理步骤和常用算法 …… 24
表 C.1 推理步骤和指定的运算符 …… 32
表 E.1 符号、缩写 …… 42
表 E.2 同义词 …… 42

前　言

GB/T 15969《可编程序控制器》分为以下几部分：

——第1部分：通用信息；

——第2部分：设备特性；

——第3部分：编程语言；

——第4部分：用户导则；

——第5部分：通信；

——第6部分：基于现场总线的可编程控制器通信；

——第7部分：模糊控制编程。

本部分为GB/T 15969的第7部分。

本部分等同采用IEC 61131-7:2000《可编程序控制器　第7部分：模糊控制编程》(英文版)。

根据GB/T 1.1—2000《标准化工作导则　第1部分：标准的结构和编写规则》，对IEC 61131-7:2000进行了下列编辑性修改：

a) “本标准”改为了“本部分”；

b) 删除了IEC 61131-7:2000的前言；

c) 凡有“IEC 61131”的地方改为“GB/T 15969”；

d) 用小数点“.”代替作小数点的逗号“,”；

本部分代替GB/T 17165.3—2001《模糊控制装置和系统　第3部分：可编程控制器　模糊控制编程》。

本部分与GB/T 17165.3-2001相比，主要变化如下：

——将“模糊控制装置和系统　第3部分：可编程控制器　模糊控制编程”改为了“可编程序控制器　第7部分：模糊控制编程”；

——将标准号由“GB/T 17165.3—2001”改为“GB/T 15969.7—2008”；

——“定义”改为“术语和定义”(GB/T 17165.3—2001的第3章；本部分的第3章)；

——将第3章中术语的排列顺序，按照IEC 61131-7:2000中术语的排列顺序进行了调整(GB/T 17165.3—2001的第3章；本部分的第3章)；

——根据GB/T 1.1—2000，进行了重新排版。

本部分的附录A、附录B、附录C、附录D、附录E都是资料性附录。

本部分由中国机械工业联合会提出。

本部分由全国工业过程测量和控制标准化技术委员会第五分技术委员会归口。

本部分负责起草单位：西南大学、中国四联仪器仪表集团。

本部分参加起草单位：机械工业仪器仪表综合技术经济研究所、北京机械工业自动化研究所。

本部分主要起草人：唐　雁、陈映萍、黄巧莉、刘进。

本部分参加起草人：冯晓升、谢兵兵。

本部分所代替标准的历次发布情况为：

——GB/T 17165.3—2001。

可编程序控制器
第7部分:模糊控制编程

1 范围

GB/T 15969的本部分定义了在可编程控制器中应用模糊控制的编程语言。

本部分规定了制造商和用户将模糊控制应用集成于GB/T 15969.3—2005规定的可编程序控制器语言中的基本方法,以及在不同编程系统之间交换可移植模糊控制程序的可能性。

为了帮助读者理解本部分内容,附录A简单介绍了模糊控制和模糊逻辑的最基本内容。

2 规范性引用文件

下列文件中的条款通过GB/T 15969的本部分的引用而成为本部分的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分,然而,鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本部分。

GB/T 2900.56—2002 电工术语 自动控制(IEC 60050-351:1998,IDT)

GB/T 15969.3—2005 可编程序控制器 第3部分:编程语言(IEC 61131-3:2002,IDT)

3 术语和定义

下列术语和定义适用于本部分。

其他有关编程语言的定义由GB/T 15969.3—2005确定。

注:本章定义的术语在定义主体中以黑体字出现。

3.1

综合 accumulation

把各条控制规则推理的结果汇总成一个总的推理结果。

注:即推理结果的聚集。

3.2

聚集 aggregation

由一条规则的多个子条件的隶属度计算该条规则条件(前件部分)的满足程度。

注:即确定一条规则的激活程度。

3.3

激活 activation

规则条件的满足程度作用于一个输出模糊集的过程。

3.4

结论 conclusion

简单或多维模糊条件语句"若……,则……"中的"则……"部分。

注:控制规则的输出,也称后件(consequent)。

3.5

条件 condition

简单或多维模糊条件语句"若……,则……"中的"若……"部分。

注:也称前件(antecedent),由模糊算子AND,OR,NOT结合各子条件组成的一个表达式。

3.6

清晰集 crisp set

模糊集的一种特例，其隶属函数仅取两个值，通常规定为0和1。

3.7

清晰化 defuzzification

将模糊(逻辑)推理后得到的模糊集转变成为用作控制的清晰值的过程。

3.8

隶属度 degree of membership

隶属函数的函数值。表示指定元素隶属于一个模糊集合的程度，其取值范围为[0,1]区间。

3.9

模糊化 fuzzification

将输入量的清晰值转变为可用于模糊(逻辑)推理的模糊集合的过程。

注：在本部分中具体指求出清晰的输入值对于相应的输入语言变量的每一个语言项的隶属度。

3.10

模糊控制 fuzzy control

用模糊(逻辑)推理方法，模拟人的操作技能、控制经验和知识的一种控制方法。

3.11

模糊逻辑 fuzzy logic

应用模糊集合理论对模糊的概念、判断和推理进行量化处理和分析的一种非经典逻辑。

注：模糊逻辑的狭义理解是指真值在[0,1]区间上取值的无限多值逻辑。

3.12

模糊算子 fuzzy operator

在模糊逻辑中所采用的算子。

注：例如AND,OR,NOT等。

3.13

模糊集 fuzzy set

带有隶属程度(以[0,1]区间上的数表示)的事物(物体、对象或概念)的整体。

推理 inference

以已知的模糊命题为前提(包括大前提和小前提)，提出新的模糊命题作为结论的过程。

3.14

语言规则 linguistic rule

“IF-THEN”规则“if-then”rule，表征模糊控制策略的模糊条件语句。

注：规则是一个形如“若<条件部分>，则<结论部分>”的语句，其中条件部分和结论部分二者之一或全部是语言项。

3.15

语言项 linguistic term

语言变量的取值。语言项是以模糊集来定义的。

注：即语言值(linguistic value)。

3.16

语言变量 linguistic variable

以人工或自然语言的词、词组或句子(语言项)作为值的变量。

3.17

隶属函数 membership function

表征论域中每一元素隶属于一个模糊集合的程度的函数。

3.18

单点集 singleton

隶属函数仅在一点为1而在其余点为0的模糊集。

3.19

子条件 subcondition

形式为一个变量或"语言变量 IS 语言项"的基本表达式。

3.20

规则库 rule base

为实现某些目标而建立的控制规则的总和。

3.21

加权因子 weighting factor

描述控制规则的重要程度、可信程度和置信程度的、其值介于0～1之间的一个数。

4 模糊控制在可编程序控制器中的集成

模糊控制应用参照本部分第5章规定的模糊控制语言(FCL)进行编程时,应封装于GB/T 15969.3—2005所定义的功能块(或程序)内。GB/T 15969.3—2005给出的功能块类型和功能块实例均适用于本部分。

以模糊控制语言(FCL)定义的功能块类型将对输入输出参数、模糊控制具体规则和说明给出规定。

相应的功能块实例应包含该模糊控制应用的具体数据。

以模糊控制语言(FCL)定义的功能块可以用于按GB/T 15969.3给出的任一种语言(如梯形图、指令表等)所编写的程序或功能块。用FCL编写的功能块或程序的输入输出参数的数据类型应与相应的"调用环境"的参数的数据类型相匹配,如图1所示。

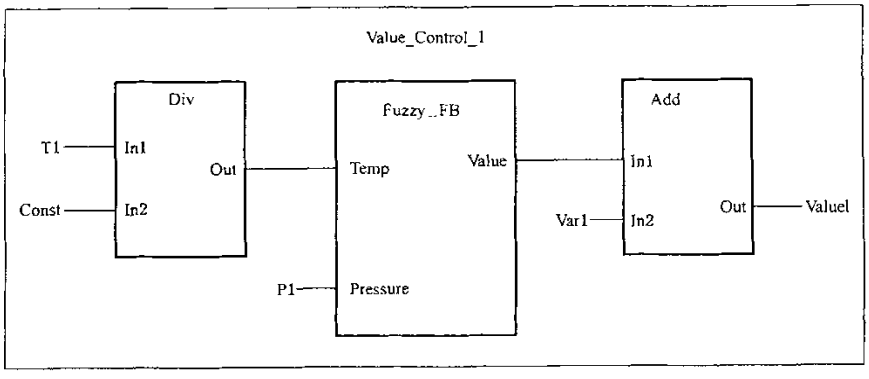


图1 用功能块图FBD表示的模糊控制功能块示例

图1中Valve_Control_1是功能块类型Fuzzy_FB的一个用户定义的功能块实例,功能块类型Fuzzy_FB可以用本部分第5章所规定的模糊控制语言(FCL)编制。本例所示功能块Fuzzy_FB用在以GB/T 15969.3—2005的图形语言FBD(功能块图)表示的程序或功能块中。

5 模糊控制语言(FCL)

5.1 模糊控制程序的交换

模糊控制语言(FCL)以GB/T 15969.3—2005规定的编程语言的定义为基础。模糊控制算法与程序环境的交互使它在程序中"隐藏"起来。所以,模糊控制算法外部表示成GB/T 15969.3—2005规定

的模块形式。描述模糊控制功能块内部的语言部分所必需的元素，如隶属函数、规则、算子和推理方法均必须按照本章来定义。

FCL 的语言元素把不同厂家的模糊控制组态工具之间的数据交换标准化成通用形式，如图 2 所示。用这种通用形式，每个生产可编程序控制器的厂家均可保持自己的硬件、软件编辑器和编译器不变。厂家只需在其编辑器中支持数据接口，用户的模糊控制项目就可以在不同厂家的产品之间交换。

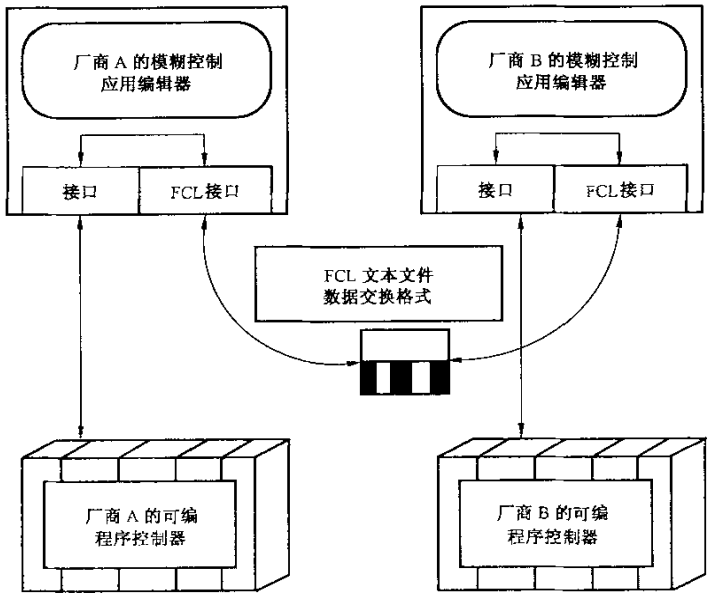


图 2 模糊控制语言(FCL)程序的数据交换

5.2 模糊控制语言元素

本条通过示例给出模糊控制语言元素，详细的产生式规则将在 5.4 中给出。

5.2.1 功能块接口

按照第 4 章，模糊功能块的格式要求使用 GB/T 15969.3—2005 给出的下列标准语言元素。

```
FUNCTION_BLOOK function_block_name          (*功能块说明*)
VAR_INPUT                                   (*输入变量说明*)
Variable_name:data_type;
...
END_VAR
VAR_OUTPUT                                  (*输出变量说明*)
Variable_name:data_type;
...
END_VAR
VAR                                         (*局部变量说明*)
...
variable_name:data_type;
END_VAR
END_FUNCTION_BLOCK
```

用这些语言元素便可以描述一个功能块的接口。功能块接口由输入输出功能块的参数来定义。这些参数的数据类型应按 GB/T 15969.3—2005 来定义。

图 3 是用结构化文本(ST)语言和功能块图(FBD)语言来描述功能块说明的示例。

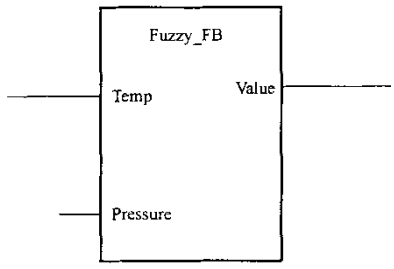
结构化文本(ST)	功能块图(FBD)
FUNCTION_BLOCK Fuzzy_FB VAR_INPUT Temp: REAL; Pressure: REAL; END_VAR VAR_OUTPUT Valve: REAL; END_VAR END_FUNCTION_BLOCK	

图 3 使用 ST 语言和 FBD 语言的功能块接口说明示例

5.2.2 模糊化

输入变量的清晰值必须变换成定义于该变量论域上的隶属函数的隶属度，这一变换在关键字 FUZZIFY 和 END_FUZZIFY 之间说明。

```
FUZZIFY  variable_name
    TERM          term_name:=membership_function;
    ……
END_FUZZIFY
```

要进行模糊化的变量名称应在关键字 FUZZIFY 之后列出。该名称应是在前面 VAR_INPUT 段已定义过的变量名。对该语言变量应赋予一个或多个语言项。由关键字 TERM 引导的语言项由隶属函数来描述，以用于计算输入变量的清晰量的隶属度。隶属函数一般是分段线形函数，由含有多个点的一张表来定义。

```
Membership_function::=(point i),(point j),…
```

每个点由一个数对，即该变量的值及相应的隶属度的值来表示，其间用逗号分开。每个数对均用括号括起来，其间也用逗号隔开。

```
point i::=value of input i | variable_name of input i, value i of membership degree
```

通过这一定义，所有简单形式的隶属函数如上(下)界型(如图 4 所示)、三角型等都易于定义。这些点应按输入变量值递升的顺序给出，隶属函数在相邻点之间约定为直线。相对每一语言项的隶属度都可以通过相邻隶属函数点之间的线性插值计算出来。

点的数目最小为 2，其最大值应受第 6 章规定的相符性等级所限制。

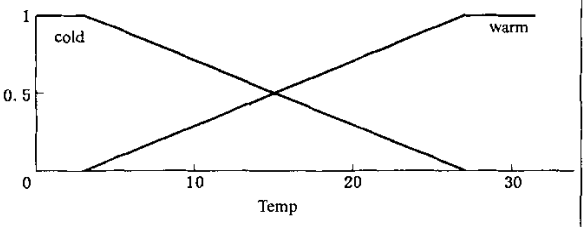
用三个点表示的语言项“warm”的隶属函数示例如下：

```
TERM warm:=(17.5,0.0),(20.0,1.0),(22.5,0.0);
```

小于第一个点的全部输入变量值的隶属度均取与第一个点同样的隶属度值。

大于最后一个点的全部输入变量的隶属度均取与最后一个点同样的隶属度值。

```
上(下)界定义:
FUZZIFY temp;
    TERM  cold:=(3,1),(27,0);
    TERM  warm:=(3,0),(27,1);
END_FUZZIFY
```



注:隶属函数的点的数据类型未加定义,厂商必须提供适应任何变换需要的编译器。

图 4 上(下)界型示例

为使模糊控制能实现在线自适应调整,隶属函数的基点应可以调整。这种调整可以通过输入到功能块的变量来实现。这些变量必须在功能块的 VAR_INPUT 部分中加以说明。用变量来定义隶属函数某些点的示例见图 5。

注:隶属函数的点的值在运行时,次序可能变化。

```
VAR_INPUT
    temp:REAL;                          (*该输入应被模糊化*)
    pressure:REAL;                      (*该输入应被模糊化*)
    bp_warm1,bp_warm2:REAL;             (*这些输入用于在线调整*)
END_VAR
FUZZIFY temp
    TERM warm:=(bp_warm1,0.0),(21.0,1.0),(bp_warm2,0.0);
    ……
END_FUZZIFY
```

图 5 隶属函数的变量用法示例

5.2.3 清晰化

输出变量的推理结果一般为一个模糊集,必须变换成一个清晰值,这种变换在关键字 DEFUZZIFY 和 END_DEFUZZIFY 之间描述。

要清晰化的变量须在关键字 DEFUZZIFY 之后列出其名称,它应是在前面 VAR_OUTPUT 部分中已定义过的变量名。

```
DEFUZZIFY variable_name
    RANGE (min..max);
    TERM term_namemembership_function;
    defuzzification_method;
    default_value;
END_DEFUZZIFY
```

语言项的定义见 5.2.2。

为了简化清晰化的过程,输出隶属函数常用单点集,它的每一个语言项仅用一个单值来描述。图 6 给出了示例。

单点集定义：

```
DEFUZZIFY    valve
    RANGE(-100,100)
    TERM drainage:=-100;
    TERM closed:=0;
    TERM inletl=100;

END_DEFUZZIFY
```

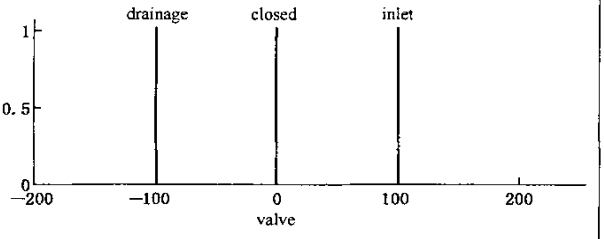


图 6　单点集语言项的示例

清晰化方法用语言元素 METHOD 来定义。

```
METHOD:defuzzification_method;
```

允许使用下列清晰化方法(见表 1 和表 2)。

表 1　清晰化方法

关　键　字	解　　释
CoG CoGS	重心(注 1)法
CoA	面积中心(注 2 和注 3)
LM	左取大(注 4)
RM	右取大(注 4)

注 1：重心法等同于求面积重心。

注 2：面积中心法等同与求面积平分线的坐标值

注 3：单点集不能使用 CoA。

注 4：LM 和 RM 清晰化方法对 0 点是非对称的。

表 2　清晰化方法的公式

CoG	$U=\dfrac{\int_{Min}^{Max}u\mu(u)\mathrm{d}u}{\int_{Min}^{Max}\mu(u)\mathrm{d}u}$
CoGS	$U=\dfrac{\sum_{i=1}^{p}[u_i\mu_i]}{\sum_{i=1}^{p}[\mu_i]}$
CoA	$U=u',\int_{Min}^{u'}\mu(u)\mathrm{d}u=\int_{u'}^{Max}\mu(u)\mathrm{d}u$
RM	$U=\sup(u'),\mu(u')=u\in[Min,\overset{\sup}{Max}]^{\mu(u)}$
LM	$U=\inf(u'),\mu(u')=u\in[Min,\overset{\sup}{Max}]^{\mu(u)}$

表 2（续）

式中：

U——清晰化结果

u——输入变量

p——单点集的个数

μ——模糊集综合后的隶属函数

i——下标

Min——在 RANGE 中给定清晰化时用到的变量最小值，在单点集的情况下 $Min=-$

Max——在 RANGE 中给定清晰化时用到的变量最大值，在单点集的情况下 $Min=+$

sup——最大值

inf——最小值

如果对一个输出变量所有语言项的隶属度都为 0，则表示对该变量无规则被激发。在这种情况下，清晰化不能生成一个有效的输出。此时，可以为输出指定一个缺省值。只有在无任何规则激发时，输出值才取缺省值。

```
DEFAULT:=value|NC
```

缺省值应在关键字 DEFAULT 之后给定。亦可用关键字 NC(不改变)来表明无规则激发时输出值保持上一步推理结果不变。

范围是指定的一个最小值和一个最大值，其间用两点分开。

```
RANGE:=(minimum value..maximum value);
```

RANGE 是用来限定输出变量的每一个隶属函数均在输出的范围之内。如果输出隶属函数采用单点集，则 RANGE 不起作用。

如果未定义其范围，则缺省范围为 GB/T 15969.3—2005 中所规定的该变量数据类型的范围。

5.2.4 规则块

模糊推理算法应在一个或多个规则块中加以定义。为适应将规则库分成几个不同模块的可能情况，可以使用若干个规则块。每个规则块有一个唯一的名称。

规则应定义在关键字 RULEBLOCK 和 END_RULEBLOCK 之间。

```
RULEBLOCK ruleblock_name
    operator_difinition;
    [activation_method;]
    accumulation_method;
    rules;
END_RULEBLOCK
```

模糊算子用于规则块中。

```
operator_difinition: :=operator:algorithm
```

根据德·摩根定律，运算符 AND 和 OR 是对偶的，例如若 MIN 用于 AND，则 MAX 用于 OR。

表 3 OR 和 AND 对偶算法

OR 算子		AND 算子	
算法关键字	算　法	算法关键字	算　法
MAX	$Max(\mu_1(x),\mu_2(x))$	MIN	$Min(\mu_1(x),\mu_2(x))$
ASUM	$\mu_1(x)+\mu_2(x)-\mu_1(x)*\mu_2(x)$	PROD	$\mu_1(x)*\mu_2(x)$
BSUM	$Min(1,\mu_1(x)+\mu_2(x))$	BDIF	$Max(0,\mu_1(x)+\mu_2(x)-1)$

规则块示例：

```
RULEBLOCK first
  AND:MIN;
  ...
END_RULEBLOCK
RULEBLOCK second
  AND:PROD;
  ...
END_RULEBLOCK
```

下面的语言元素定义了激活方法：

```
ACT:activation_method;
```

表 4 列出了可选用的激活方法。

表 4 激活方法

名　称	关 键 字	算　法
乘积	PROD	$\mu_1(x) * \mu_2(x)$
取小运算	MIN	$Min(\mu_1(x),\mu_2(x))$
注：激活方法与是否用单点集无关。		

下面的语言元素定义了综合方法：

```
ACCU:accumulation_method;
```

表 5 列出了可用的综合方法。

表 5 综合方法

名　称	关 键 字	公　式
取大算法	MAX	$Max(\mu_1(x),\mu_2(x))$
归一化求和	NSUM	$\frac{\mu_1(x)+\mu_2(x)}{Max(1,Max_{x'\in X}(\mu_1(x')+\mu_2(x')))}$
有界和	BSUM	$Min(1,\mu_1(x)+\mu_2(x))$

规则块的输入是带有各自语言项集合的语言变量，每一个语言项赋有一个隶属函数。

规则定义在规则块内，每条规则以关键字 RULE 开始，后面是规则序号，并以分号结束。在规则块内的每一规则只能有唯一的序号。

```
RULE numbers:IF condition THEN conclusion [WITH weighting factor];
```

规则本身以关键字 IF 开始，紧接着为条件(condition)，条件后面是以关键字 THEN 开头的结论(conclusion)。

可以把几个子条件(subcondition)和输入变量(variable)组合在一条规则的条件(condition)之中。变量的作用是将模糊隶属度导入模糊功能块。所有的输入项应定义在关键字 IF 和 THEN 之间，并通过以关键字 AND、OR 或 NOT 表示的运算符组合在一起。

运算符的优先级(见表 6)按表 3 所示布尔代数的运算规律。

表 6 运算符优先级

优 先 级	运 算 符
1	()括号
2	NOT
3	AND
4	OR

规则的简化示例：

```
RULE1:IF subcondition1 AND variable1 OR variable2 THEN conclusion;
```

在相符性的基本级中，OR 操作可以通过定义成两条规则来实现：

```
RULE3:IF subcondition1 OR subcondition2 THEN conclusion;
替代为：
RULE3a:IF condition1 THEN conclusion;
RULE3b:IF condition2 THEN conclusion;
```

子条件以语言变量名开始，后面是关键字 IS 带一可选项 NOT 和条件中用到的语言变量的一个语言项。

```
subcondition:=linguistic_variable IS [NOT] linguistic_term
```

条件中的语言项应与该条件中的语言变量相匹配，用到的语言项必须预先用关键字 TERM 定义。

子条件示例：

```
temp IS hot
temp IS NOT hot
```

关键字 NOT 也可以用在子条件之前，此时可使用括号将子条件括起。

```
IF NOT temp IS hot THEN…    或 IF NOT (temp IS hot) THEN…
```

结论可以分为几个子结论和输出变量，各子结论之间用逗号隔开。

子结论以语言变量开始，随后是关键字 IS 和该语言变量的一个语言项。

```
subconclusion:=linguistic_variable IS linguistic_term
```

有几个子结论(写成一行或多行)的示例：

```
IF temp IS cold AND pressure IS low THEN var1,valve1 IS inlet,valve2 IS closed;
或写在多行上：
IF temp IS cold AND pressure IS low
THEN var1,
    valve1 IS inlet,
    valve2 IS closed;
```

可以选择给任一个子结论指定一个介于 0.0 和 1.0 之间的实数作为加权因子。这可以通过关键字 WITH 后跟加权因子来实现。

加权因子与子结论的结果相乘使子结论的隶属度(隶属函数)减小。

为了在外部操纵模糊控制应用的参数，加权因子允许是一个变量。这种情况下，该变量必须在 VAR_INPUT 部分加以说明。这样，便可以在运行时通过调整加权因子使模糊控制程序适应过程控制的要求。

如果子结论后面没有 WITH 语句，则视加权因子为缺省值 1.0。

```
IF condition THEN subconclusion [WITH weighting_factor],subconclusion;
```

常数加权因子的示例：

```
IF temp IS cold AND pressure IS low THEN valve1 IS inlet WITH 0.5.
                                        valve2 IS closed;
```

变量加权因子的示例：

```
VAR_INPUT
    w_myrule1:REAL:=0.8;
END_VAR
RULEBLOCK temp_rule
    RULE1:IF temp IS cold AND pressure IS low
          THEN valve IS inlet WITH w_myrule1;
...
END_RULEBLOCK
```

5.2.5 可选参数

为在不同的目标系统上实现模糊控制应用，可能需要给出附加信息，使这些应用在不同的系统之间可能取得最佳的转换。这些附加信息用语言元素写在关键字 OPTIONS 和 END_OPTIONS 之间。

```
OPTIONS
    application_specific_parameters
END_OPTIONS
```

这些语言元素将用于按第 6 章规定的相符性开放级的特征描述。

5.3 模糊控制语言(FCL)示例

图 7 是模糊控制语言示例。

```
FUNCTION_BLOCK Fuzzy_FB
VAR_INPUT
        Temp:       REAL;
        Pressure:   REAL;
END_VAR
VAR_OUTPUT
        Valve:      REAL;
END_VAR
FUZZIFY temp
        TERM cold     :=(3,1),(27,0);
        TERM hot      :=(3,0),(27,1);
END_FUZZIFY
FUZZIFY pressure
        TERM low      :=(55,1),(95,0);
        TERM high     :=(55,0),(95,1),
END_FUZZIFY
DEFUZZIFY valve
        TERM drainage   :=-100;
        TERM closed     :=0;
        TERM inlet      :=100;
        METHOD:     CoGS;
        DEFAULT       :=0;
END_DEFUZZIFY
```

图 7 模糊功能块示例

```
RULEBLOCK No1
        AND:        MIN;
        ACCU:       MAX;
        RULE1:      IF temp IS cold AND pressure IS low THEN valve IS inlet;
        RULE2:      IF temp IS cold AND pressure IS high THEN valve IS closed WITH 0.8;
        RULE3:      IF temp IS hot AND pressure IS low THEN valve IS closed;
        RULE4:      IF temp IS hot AND pressure IS high THEN valve IS drainage;
END_RULEBLOCK
END_FUNCTION_BLOCK
```

图 7（续）

5.4 模糊控制语言（FCL）的产生式规则和关键字

GB/T 15969.3—2005 的附录 A 定义了可编程序控制器文本语言的说明方法，这里，FCL 要用到这种说明方法。

GB/T 15969.3—2005 的附录 B 有文本编程语言的语言元素的正式规定。FCL 要用到下列 GB/T 15969.3—2005 的语言元素的子集：

B1.1 字母、数字、标识符

B1.2 常数

B1.3 数据类型

B1.4 变量

5.4.1 产生式规则

除了上面列出的 GB/T 15969.3—2005 的语言元素外，还要用到下列的语言元素：

```
function_block_declaration::='FUNCTION_BLOCK'function_block_name
                                {fb_io_var_declarations}
                                {other_var_declarations}
                                Function_block_body
                                'END_FUNCTION_BLOCK'
fb_io_var_declarations::=input_declarations | output_declarations
other_var_dedlarations::= var_declarations
funcition_block_body::={fuzzify_block}
                       {defuzzify_block}
                       {rule_block}
                       {option_block}
fuzzify_block::=       'FUZZIFY'variable_name
                         {linguistic_term}
                       'END_FUZZY'
defuzzify_block::=     'DEFUZZIFY'f_variable_name
                         [range]
                         {linguistic_term}
                         defuzzification_method
                         default_value
                       'END_DEFUZZIFY'
rule_block::=          'RULE_BLOCK'rule_block_name
                         operator_definition
```

```
                    [activation_method]
                    accumulation_method
                    {rule}
                  'END_RULEBLOCK'
option_block::=   'OPTION'
                    any manufacturer specific parameter
                  'END_OPTION'
linguistic_term::= 'TERM'term_name':='membership_function';'
membership_function::=singleton | points
```

注：单点集，请参阅 5.2.3。

```
singleton::=      numeric_literal | variable_name
points::=         {'('numeric_literal|varable_name','
                      numeric_literal')'}
```

注：关于允许的点数，请参阅 5.2.2。

```
defuzzification_method::= 'METHOD'':''CoG'|'CoGS'|'CoA'|'RM'|'LM'
default value::=          'DEFAULT'' :='numeric_literal |'NC'';'
range::=                  'RANGE('numeric_literal '..'numeric_literal ')'';'
operator_defintion::=     [('OR''MAX'|'ASUM'|'BSUM')]
```

注：关于对偶的算法，见表 3。

```
activation_method::= 'ACT'':''PROD'|'MIN'';'
accumulation_method::='ACCU' ':' 'MAX'| 'BSUM'| 'NSUM' ';'
rule::=              'RULE'integer_literal ';'
                     'IF' condition 'THEN' conclusion ['WITH'weighting_factor] ';'
condition::=         x{('AND'x)| ('OR'x)}
x::=                 ['NOT'](subcondition|('('condition ')'))
subcondition::=      variable_name|(variable_name 'IS'[NET]term_name)
conclusion::=        {(variable_name|(variable_name 'IS'term_name)) ','}
                       (variable_name|variable_name 'IS'term_name)
weighting_factor::=  variable|numeric_literal
function_block_name::= identifier
rule_block_name::=   identifier
term_name::=         identifier
f_variable_name::=   identifier
variable_literal::=  identifier
numeric_literal::=   identifier_literal|real_literal
input_declarations::= 见 GB/T 15969.3—2005 附录 B
output_declarations::= 见 GB/T 15969.3—2005 附录 B
var_declarations::=  见 GB/T 15969.3—2005 附录 B
identifier::=        见 GB/T 15969.3—2005 附录 B
```

5.4.2 关键字

表 7 保留关键字

关 键 字	意 义	参见
()	条件、语言项、范围中用到的符号	5.2.4
ACCU	综合方法	5.2.4
ACT	激活方法	5.2.4
AND	AND 算子	5.2.4
ASUM	OR 算子、代数和	5.2.4
BDIF	AND 算子、有界差	5.2.4
BSUM	综合方法,OR 算子,有界和	5.2.4
CoA	面积中心清晰化方法	5.2.3
CoG	重心清晰化方法	5.2.3
CoGS	单点集重心清晰化方法	5.2.3
DEFAULT	没有规则触发时的默认输出值	5.2.3
DEFUZZIFY	输出变量的清晰化	5.2.3
END_DEFUZZIFY	清晰化说明结束	5.2.3
END_FUNCITION_BLOCK	功能块说明结束	5.2.1
END_FUZZIFY	模糊化说明结束	5.2.2
END_OPTIONS	可选项说明结束	5.2.5
END_RULEBLOCK	规则块说明结束	5.2.4
END_VAR	输入输出变量定义结束	5.2.1
FUNCTION_BLOCK	功能块	5.2.1
FUZZIFY	输入变量模糊化	5.2.2
IF	规则的开始,后跟条件	5.2.4
IS	用于条件和结论中,跟在语言变量之后	5.2.4
LM	左取大清晰化方法	5.2.3
MAX	最大值综合方法,OR 算子	5.2.4
METHOD	清晰化方法	5.2.3
MIN	作为 AND 算子取小,激发方法	5.2.4
NC	没有规则激发时输出值维持不变	5.2.3
NOT	NOT 算子	5.2.4
NSUM	归一化代数和综合方法	5.2.4
OPTIONS	可选参数定义	5.2.5
OR	OR 算子	5.2.4
PROD	作为 AND 算子求积,激活方法	5.2.4
RANGE	输出变量范围的限定	5.2.3
RM	右取大清晰化方法	5.2.3
RULE	模糊规则说明的开始	5.2.4
RULEBLOCK	规则块说明的开始	5.2.4
TERM	为语言变量定义其一个语言项(隶属函数)	5.2.2
THEN	将条件和结论隔开	5.2.4
VAR	局部变量定义	5.2.1
VAR_INPUT	输入变量定义	5.2.1
VAR_OUTPUT	输出变量定义	5.2.1
WITH	加权因子定义	5.2.4

6 相符性

6.1 模糊控制语言(FCL)的相符性等级

应用模糊控制语言(FCL)的控制系统相符性等级划分如图 8 所示。这个级别体系包括以下三级：

基本级　包含 GB/T 15969.3—2005 中规定的功能块和数据类型的定义。

扩展级　允许选择表 9 中所列出的特性。

开放级　允许使用本部分中未包括的附加特性，由厂家用表列出。

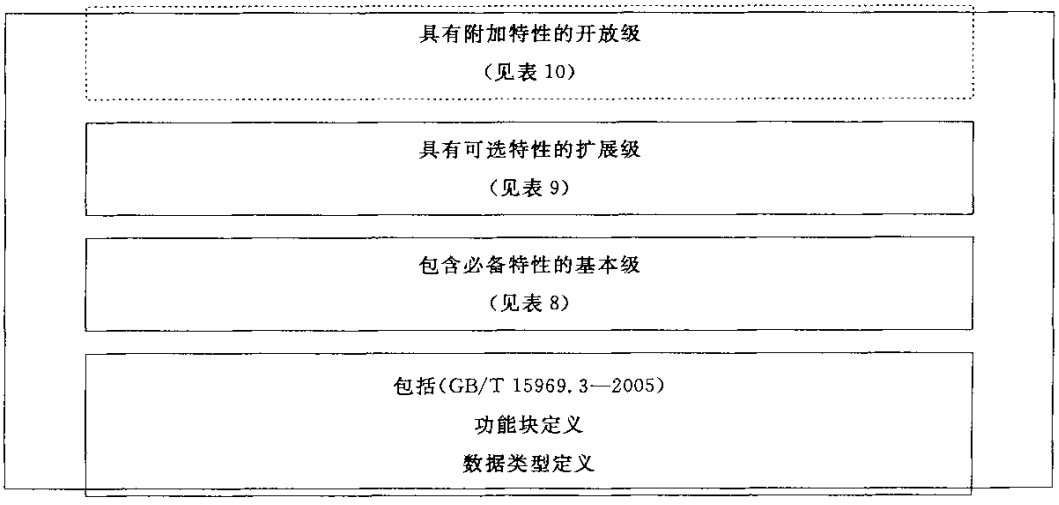


图 8　相符性等级

采用模糊控制语言(FCL)且符合本部分规定的控制系统，应遵循以下规定：

a) 应使用 GB/T 15969.3—2005 规定的功能块特性来实现模糊控制功能。因此，功能块的定义及模糊控制功能块需要的输入输出参数的数据类型的定义均应符合 GB/T 15969.3—2005 的规定。

b) 表 8 所定义的模糊控制功能的所有特性均应依据本部分的各条予以实现。该表定义了所有遵循本部分的控制系统均应具备的基本级元素的集合。

c) 在基本特性之外，表 9 所定义的扩展级的元素子集是可选择实现的附加元素。其实现应准确地根据本部分各条的定义。应将这些特性注明为标准的扩展，并用如表 9 的形式，把所实现的特性列成表作为系统文件的一部分。

d) 可以有基本级和扩展级之外的特性，只要这些特性没有与标准特性相同或相似的功能或表示，以避免任何可能的混淆。这些特性应注明为开放级特性，并且用如表 10 的形式列出来作为系统文件的一部分。

e) 在不同模糊控制系统之间应用程序的交换，应以本部分规定的 FCL 文本形式进行。输入输出的格式要符合本部分，以有效地保证交换。

f) 为了使用户最方便且不影响其进一步的改善，模糊控制应用程序的设计、输入和测试等外部表示可采用任何的图形或文本的方式来实现。

表 8 中的元素是所有遵循本部分的模糊控制系统均应实现的基本特性的集合。

表 8 FCL 基本级语言元素(必备)

语言元素	关键字	详细说明
功能块说明	VAR_INPUT VAR_OUTPUT	包括输入、输出变量
隶属函数	输入变量:TERM	最多三个常数点 (隶属度取值 0 或 1)
	输出变量:TERM	仅用常数单点集
条件聚集	算子:AND	算法:MIN
激活	—	无关,因仅用单点集
综合(结论的聚集)	算子:ACCU	算法:MAX
清晰化	METHOD	算法:CoGS
缺省值	DEFAULT	NC,具体值
规则块	RULEBLOCK	仅有一个规则块
条件	IF…IS…	n 个子条件
结论	THEN	仅有一个子结论

表 9 中的元素是在标准模糊控制系统中可选择实现的扩展特性(如对 AND 算子,可选取算法 PROD 或 BDIF,或两者都选)。

表 9 FCL 扩展级语言元素(可选)

语言元素	关键字	详细说明
功能块说明	VAR	包含局部变量
隶属函数	输入变量:TERM	最多四个点(隶属度取值 0 或 1)
	输出变量:TERM	最多四个点(隶属度取值 0 或 1)
条件聚集	算子:AND	算法:PROD,BDIF
	算子:OR	算法:ASUM,BSUM
	算子:NOT	算法:1——(变元)
	括号	()
激活	算子:ACT	算法:MIN,PROD
综合	算子:ACCU	算法:BSUM,NSUM
清晰化范围	算子:RANGE	RANGE(最小值,最大值)对输出变量的范围的限定
清晰化方法	算子:METHOD	算法:CoG,CoA,LM,RM
规则块	算子:RULEBLOCK	n 个规则块
条件	IF	n 个子条件,n 个输入变量
结论	THEN	n 个子结论,n 个输出变量
加权因子	WITH	常数或赋给在说明部分 VAR_INPUT…END_VAR 中定义的变量的值

表 10 给出了一个开放级语言元素清单的例子，该清单应是系统文件的一个组成部分。

表 10 开放级语言元素清单示例

自由定义输入输出隶属函数(如：高斯函数，指数函数)
多于四个点的隶属函数
隶属度取值可以从 0 到 1
隶属度的值可取变量

6.2 数据检查清单

数据检查清单(见表 11)应随技术文档一同提交。在该清单中，可编程序控制器、模糊编程工具和应用软件的厂商应描述其模糊控制系统特有的性能特征。为了方便模糊控制应用在不同的厂商系统之间移植，下面的数据检查清单可用来检查程序移植的可能性。表 11 不可能全面，只是作为一个例子而已，厂商可予以补充。

表 11 数据检查清单

技 术 数 据	制造商声明(例)
功能块输入输出数据类型	REAL,INT
FCL 程序中的注释行	YES,NO
执行时间(ms)	20,30
所需存储器(KB)	3,4
将加权因子和隶属度的变化从 0.0 到 1.0 映射到整数值范围	0～200,0～400
标识符长度(如变量名，规则块名，语言项名)	6,8
需模糊化的输入变量的最大数目	6,8
每个输入变量隶属函数语言项的最大数目	5,7
所有输入变量隶属函数语言项总的最大数目	30,56
每个输入变量语言项对应的隶属函数的最大点数	3,4,10
所有输入变量语言项对应的隶属函数总的最大点数	90,224
需清晰化的输出变量的最大数目	6,8
每个输出变量隶属函数语言项的最大数目	5,7
所有输出变量隶属函数语言项总的最大数目	30,56
与每一个输出变量语言项相应的隶属函数的最大点数	1,4,10
与所有输出变量语言项相应的隶属函数总的最大点数	90,224
规则块的最大数目	1,10
每个规则块中规则的最大数目	10
每个规则子条件的最大数目	4,10
所有规则总数的最大数目	60
每条规则子结论的最大数目	4
()的嵌套深度	1,3

附 录 A
（资料性附录）
理论

本附录为对第3章中所做定义的解释。

A.1 模糊逻辑

在传统开环、闭环控制系统中，物理变量用相应的名称或数字（通常为实数）来描述，而在模糊逻辑中，则用语言值和表达式来描述。词“低”或“大开”可以用作物理变量“温度”或“加热阀开度”的语言项。如果某输入变量用语言项来描述，它就被视为一个语言值。

每个语言项用一个模糊集 M 来描述。这样它就从数学上被基本集（论域）G 和隶属函数 μ 明确地加以定义。隶属函数用一个0和1之间的数值来表示论域 G 上的每一个元素（如时间尺度的数值[以岁数表示年龄]）在集合 M（如“年轻”）中的隶属度。如果某个特定值的隶属函数为1，则表示对应于该语言项的语句完全成立（如“年轻”对于20岁）；相反，如其值为0，则表示完全不成立（如“非常老”对于20岁）。

模糊集可以用下面的符号来描述：

对有限集：采用递增形式的数值对集合：

$$M=\{(x_1,\mu_M(x_1)),(x_2,\mu_M(x_2)),\cdots,(x_n,\mu_M(x_n))\}\quad x_i\in G,i=1,2,\cdots,n \tag{A.1}$$

这里 $\mu_M(x_i)$ 以其数字取值列出。

对无限集：

$$M=\{(x,\mu_M(x))\}\quad x\in G \tag{A.2}$$

为了更清楚地说明清晰词和模糊词的差别，语言变量“年龄”的语言项“成年人”表示如图A.1。“法定年龄”由法律明确规定，因此对应隶属函数呈现不连续的跳变，而对“成年人”就不可能给出一个清晰的年龄界限。

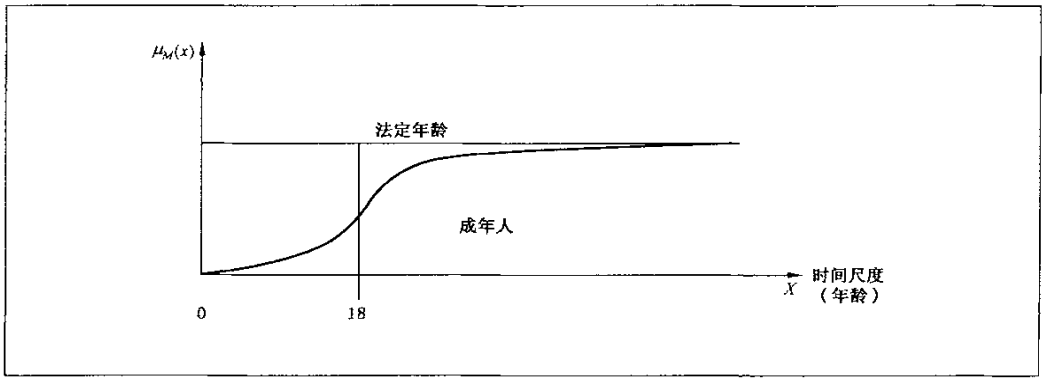


图 A.1 语言项“法定年龄”和“成年人”的隶属函数

作为示例，图A.2示出用语言项描述语言变量“年纪”以及用隶属度的方法描述在时间尺度“年龄”上的对应关系。

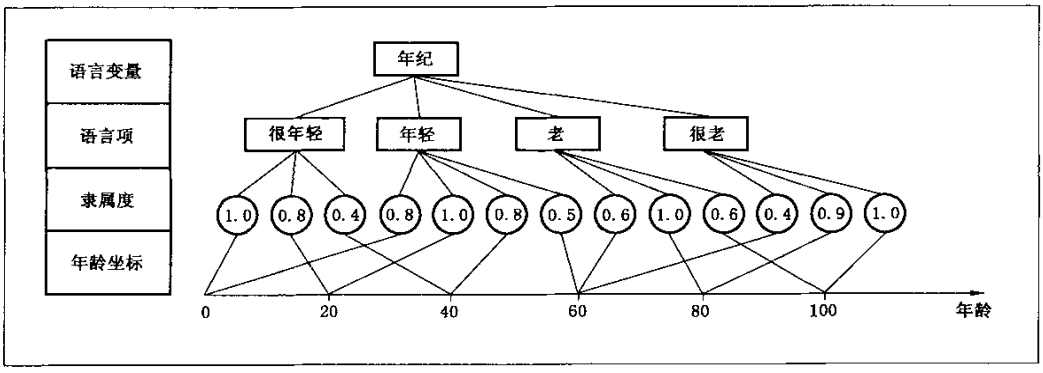


图 A.2 用语言形描述语言变量"年纪"和其在时间尺度(年龄)上的对应关系

隶属函数的典型形式如图 A.3 所示,下面是几个特殊形式:

——用矩形(即区间)来定义,以描述多个值;

——"单点集",作为输出变量的另一种表示。

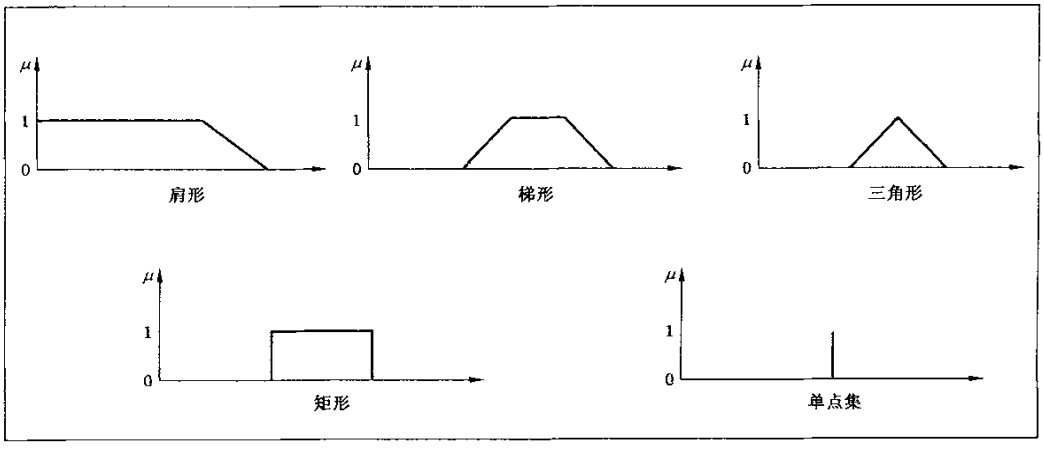


图 A.3 常用的隶属函数的形状

一个表达式,其中一些语言变量和一些语言项相联系,在模糊逻辑中就表示一个语句。像"温度是高"或"温度是低"这样具有以下简单的基本结构的表达式就被视为语句:

Linguistic variable-Symbol of comparsion-Linguistic term (A.3)
(Temperature IS low)

与经典逻辑假定命题仅仅取"真"和"假"两个真值不同,模糊逻辑中的语句有一个 0~1 之间的隶属度。

经验知识可以表示成规则,规则 R_K 的形式如下:

R_K :IF condition P_K THEN conclusion C_K (A.4)

每条规则的条件由单条或多条语句(多输入变量)组合而成。而结论则是输出变量的动作指令。

$P_K = A$ AND B OR (NOT C) (A.5)

如果条件和/或结论由语句描述，则该规则也被视为语言规则。规则库由若干条规则组成。与经典的基于规则的系统不同，通常，在同一时刻它有几条规则同时响应(被激发)。因此，多条规则的结果必须通过相应的数学算子组合起来。

模糊集与模糊集之间的运算关系用隶属函数来定义：

下面的关系适用于模糊集 A 和 B，其中的元素 X 来自基本集(论域)G：

相等 $A=B$：

为真，如果 $\mu_A(x)=\mu_B(x)$，对所有 $x\in G$ (A.6)

包含 $A\subseteq B$：

为真，如果 $\mu_A(x)\leqslant\mu_B(x)$，对所有 $x\in G$ (A.7)

真包含 $A\subset B$：

为真，如果 $\mu_A(x)\leqslant\mu_B(x)$，对所有 $x\in G$

并且 $\mu_A(x)<\mu_B(x)$，至少有一个 $x\in G$ (A.8)

下面规定两个模糊集 A 和 B 之间的运算，其元素 x 来自基本集(论域)G：

交：$A\cap B$ 定义为 $\mu_{A\cap B}(x)=\cap(\mu_A(x),\mu_B(x))$，其中称为∩交算子 (A.9)

并：$A\cup B$ 定义为 $\mu_{A\cup B}(x)=\cup(\mu_A(x),\mu_B(x))$，其中称为∪交算子 (A.10)

补：定义为 $\mu_{\overline{A}}(x)=1-\mu_A(x)$ (A.11)

如同经典清晰集一样，下面的说明也适用于模糊集：

交相当于模糊算子 AND

并相当于模糊算子 OR

补相当于模糊算子 NOT

交、并、补在数学上的基本运算如下(参见图 A.4)：

对于交，取最小值：

$\mu_{A\cap B}(x)=Min(\mu_A(x),\mu_B(x))\quad x\in G$ (A.12)

对于并，取最大值：

$\mu_{A\cup B}(x)=Max(\mu_A(x),\mu_B(x))\quad x\in G$ (A.13)

对于补，用 1 减去该值：

$\mu_{\overline{A}}(x)=1-\mu_A(x)\quad x\in G$ (A.14)

对于模糊逻辑算子 AND 和 OR，还有许多可能的算法。但应指出，AND 和 OR 算子不能随便地选用。

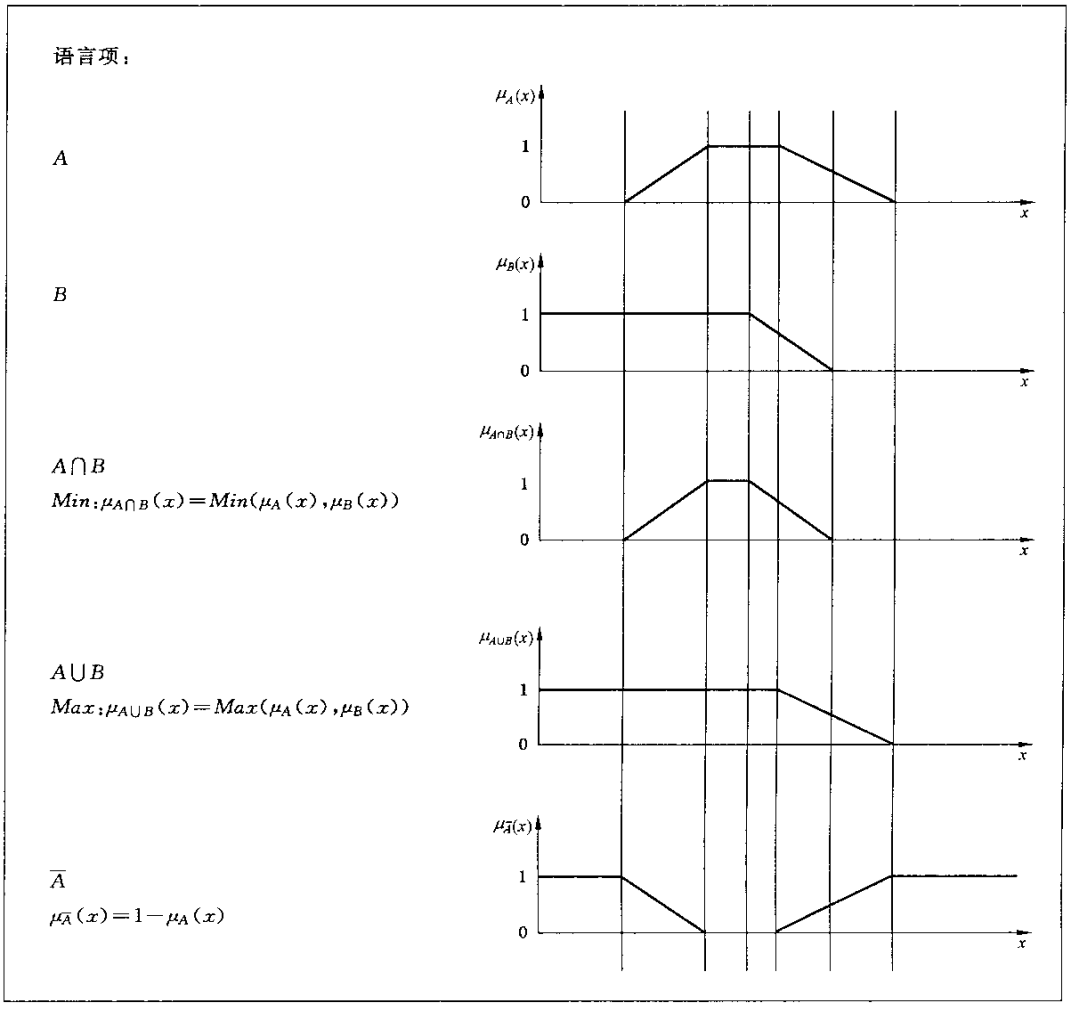


图 A.4 两个隶属函数之间运算的算法

A.2 模糊控制

模糊控制是指对测量值的处理是基于模糊规则并用模糊逻辑算法对这些规则进行处理的生产工艺过程的开环或闭环控制。

输入信息包括可测量的过程变量的实变量、导出变量和设定值。输出变量是校正变量的实变量。需要在过程的输入输出变量和模糊集合之间进行变换(模糊化,清晰化)。模糊控制的核心部分是规则库中的控制规则和推理。

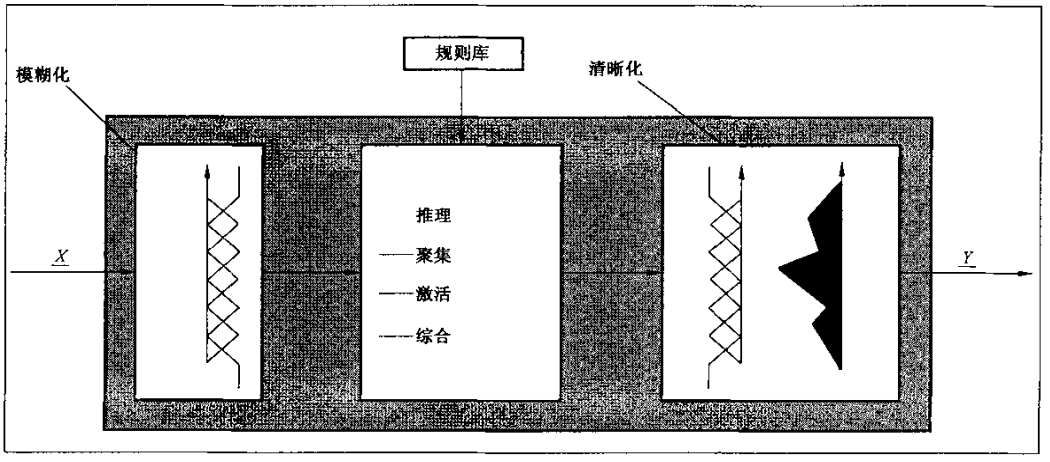


图 A.5 模糊控制的结构和功能元素

上述示于图 A.5 的模糊控制的功能元素，将解释如下。

A.2.1 模糊化

确定输入量与各语言项相匹配的程度称为模糊化。为此，要确定输入变量对语言变量的各个语言项的实际隶属度。图 A.6 是模糊化的示例。

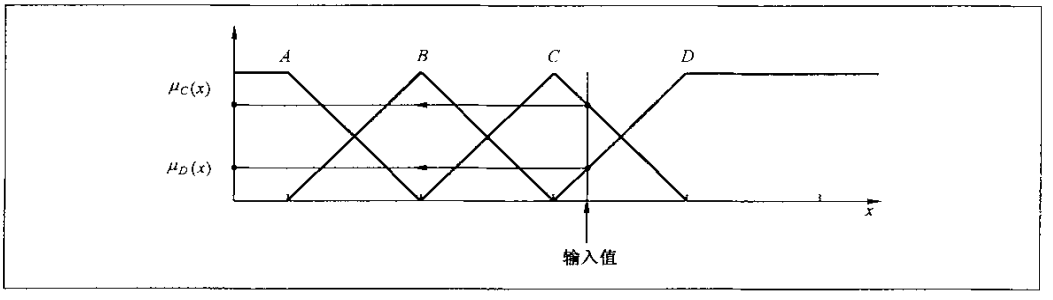


图 A.6 模糊化原理(示例)

A.2.2 规则库

规则库包含与所考虑的特定过程操作有关的经验知识。控制规则就是用来描述表示这些经验的。只要模糊算子 AND 取 MIN、OR 取 MAX，当一条模糊规则 R_j 的前件由 OR 结合的 m 条语句组成时，可以将其拆分成前件仅包含 AND 语句的 m 条规则来替代。图 A.7 和图 A.8 是不同规则表示的例子。

规则库

——以文本形式定义的规则库

R1：IF 条件 P_1 THEN 结论 C_1

R2：IF 条件 P_2 THEN 结论 C_2

R3：IF 条件 P_3 THEN 结论 C_3

…

Rn：IF 条件 P_n THEN 结论 C_n

其中条件可以是多条语句的结合，如对于规则 K

$P_K = P_{K1}$ AND P_{K2} OR (NOT P_{K3})

结论也可能不止一个，如

$C_K = C_{K1}, C_{K2}$

图 A.7 用控制规则形式表示的知识库

如果有两个输入和一个输出，且这两个输入变量仅以 AND 结合起来，则规则库可以用矩阵形式给出，其中，输入变量的值排成矩阵的列和(或)行，矩阵内的元素则是输出变量的值。

输入变量 X_2 \ 输入变量 X_1	NB	NS	Z	PS	PS
NB	NB	NS	Z	PS	PS
NS	NS	NS	PS	PS	PB
Z	Z	Z	PS	PB	PB
PS	Z	Z	PS	PB	PB
PB	Z	PS	PB	PB	PB

NB=Ngeative Big　NS=Negative Small　Z=Zero

PB=Positive Big　PS=Positive Small

图 A.8　两个变量的矩阵表示

A.2.3　推理

模糊规则库的更一般和数学上证明过的定义乃是基于广义取式和模糊蕴涵原理。这里所使用的原理和定义是用于模糊规则库的简单情况，它基本上是取之于广泛使用的 Mamdani 推理方法。更复杂的推理方法此处不予讨论。

推理包括三个子功能：聚集、激活和综合，如图 A.9 所示。参见图 A.10。

规则库的推理可以分成以下的步骤：

- 聚集：

  从一条规则前件的子条件的隶属函数确定前件条件的满足程度

  $$P_{K1} \quad \text{AND} \quad P_{K2} \quad \text{OR} \quad (\text{NOT} \quad P_{K3}) = P_K$$

- 激活：

  对所有 n 条规则，由 IF 条件部分的满足程度激活 THEN 部分的结论

  IF　条件　P_K　THEN　结论　C_K

  当规则有加权因子时，还应考虑每条规则的加权因子

- 综合：

  由各条规则的加权结果组合成一个总的结果

图 A.9　推理元素

——聚集：

如果条件只含有一个子条件，条件的确认与子条件的确认乃一回事，即条件的满足程度就是该子条件的满足程度。但如果条件含有几个子条件，则满足程度必须由各个子条件的满足程度通过聚集来确定。如果条件是由子条件用 AND 组合起来的，那么满足程度就用模糊算子 AND 计算。

——激活：

在结论中，各子结论对应输出变量。以通过聚集确定的条件的满足程度为基础来确定结论或子结论的隶属度。通常，MIN 或 PROD 算子用于激活。

如果规则库包含有带加权因子 W_K 的规则，其中 $W_K \in [0,1]$，此时用以激活结论的条件满足程度取：

$$C_K^* = W_K \times C_K \qquad (A.15)$$

注：用乘法实现，其中 C_K 为通过聚集确定的条件的满足条件。

——综合：

各条规则的结果组合到一起得到一个总结果，综合通常用取大算法。表 A.1 给出了各个推理步骤通常应用的算子。

各个步骤中算子的不同结合导致不同的推理策略。其中最有名的是 MaxMin 推理和 MaxProd 推理，二者都用取大算子作综合，但用取小或代数积来激活。在 MaxMin 推理中，结论模糊集的隶属函数用条件的满足程度来限制，然后，用取大来合成一个模糊集。在 MaxProd 推理中则不同，结论模糊集合的隶属函数同条件的满足程度相乘，然后用取大来合成。

表 A.1 推理步骤和常用算法

推理步骤	算子	算法
聚集(Aggregation)		
对于 AND	Minimum	$a_K = Min\{a_{K1}(x_1), a_{K2}(x_2)\}$
对于 OR	Maximum	$a_K = Max\{a_{K1}(x_1), a_{K2}(x_2)\}$
激活(Activation)		
IF…THEN 结论的变换		
	Minimum	$c'_K = Min\{a_K, \mu_\infty(u)\}$
每条规则的加权因子		
	Multiplication	$c_K = Mult\{\omega_K, c'_K\} = \omega_K \times c'_K$
综合(Accumulation)	Maximum	$\mu = Max\{c_i(u)\}$

$$P_K = \underbrace{P_{K1} \quad \text{AND} \quad P_{K2}}_{Min\{a_{K1}, a_{K2}\}} \quad \underbrace{OR \quad (\text{NOT} \quad P_{K3}}_{(1-a_{K3})}$$

$$a_K = Max[Min\{a_{K1}, a_{K2}\}, (1-a_{K3})]$$

图 A.10a) 聚集原理(举例说明)

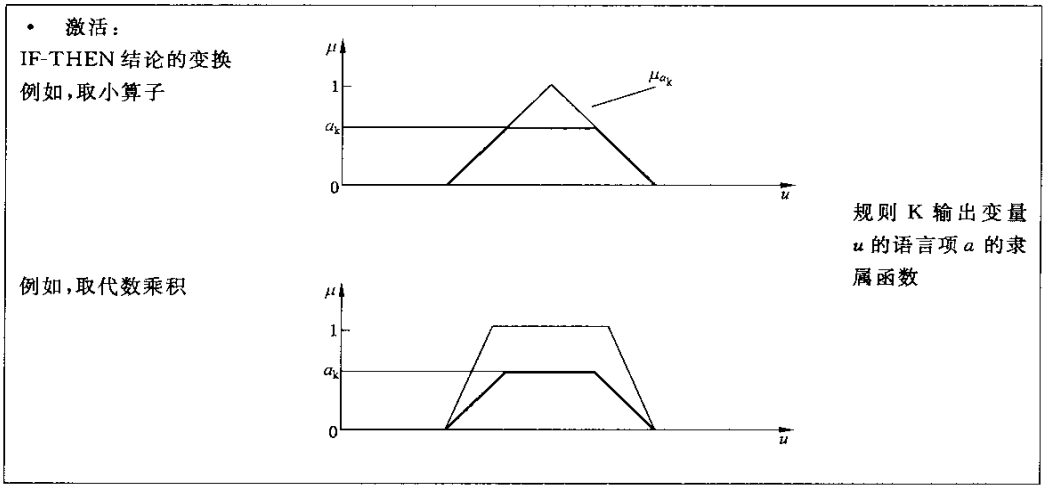


图 A.10b) 激活原理(举例说明)

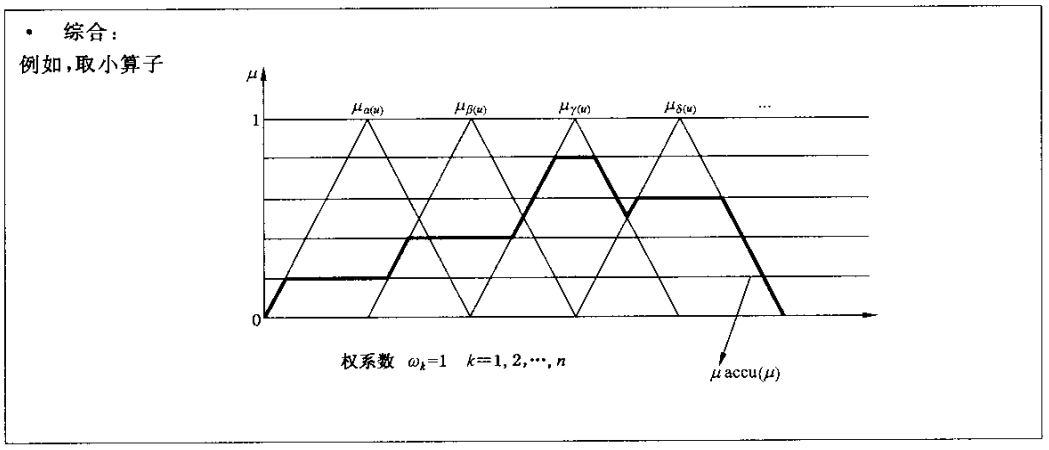


图 A.10c) 综合原理(举例说明)

A.2.4 清晰化

推理导出的结果一般是一个模糊集或它的隶属函数。控制机构不能直接地处理这样的模糊信息，所以，推理的结果必须转化成清晰的数字量(见图 A.11)。从这个意义上说，待定的清晰数(通常是一个实数)应该很好地代表模糊集中所包含的信息。

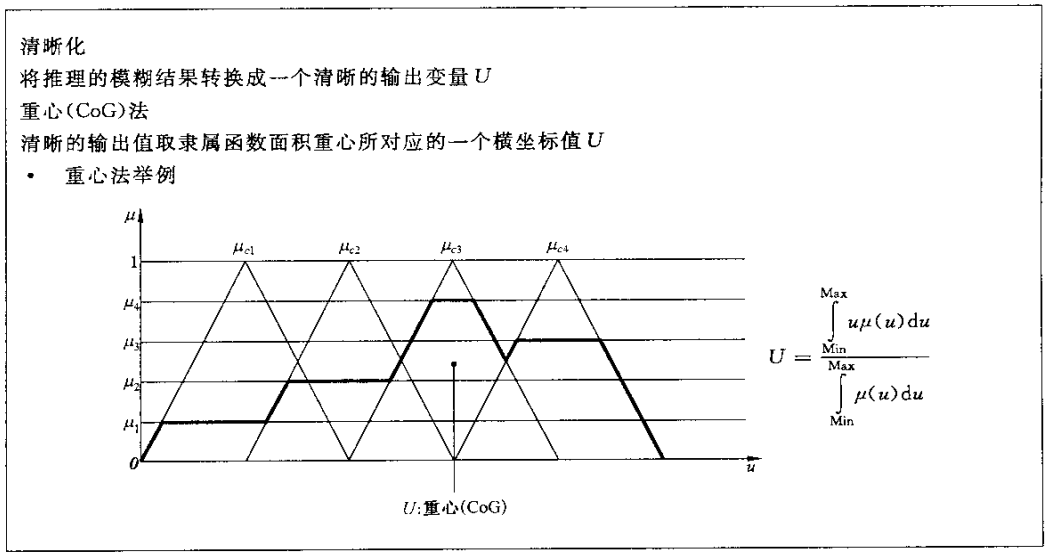


图 A.11a) 清晰化方法

除此之外，常用的方法还有：

左取大法(LM)：

输出变量值取输出隶属函数达到最左极大值所对应的值。

右取大法(RM)：

输出变量值取输出隶属函数达到最右极大值所对应的值。

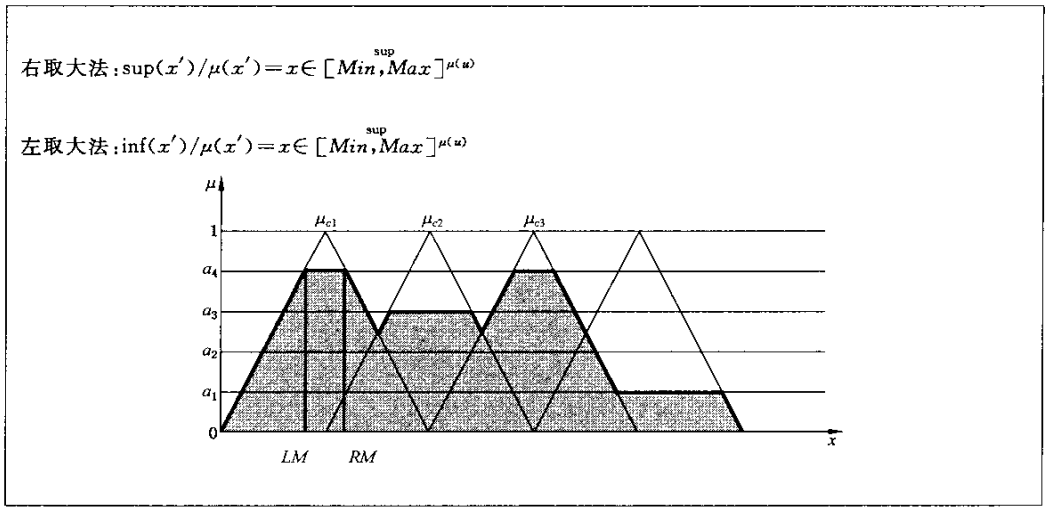


图 A.11b) 左取大和右取大的区别

面积中心线方法

此法，将等分隶属函数面积的垂直线所对应的横坐标值作为输出值。

注 1：重心等价于面积重心(centroid)。

注 2：面积中心线等价面积平分。

注 3：面积中心线(CoA)不适用于单点集。

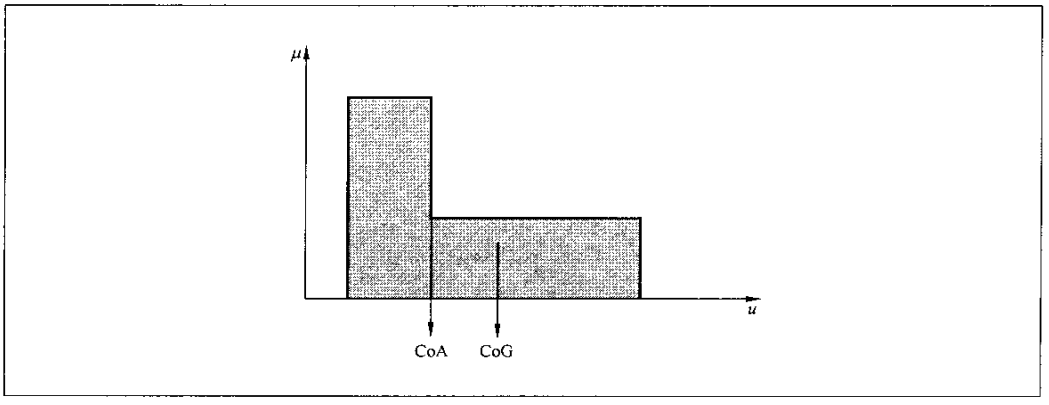


图 A.11c) 面积中心线和重心的区别

注：清晰化方法 CoA 和 CoG 由表 2 给出。

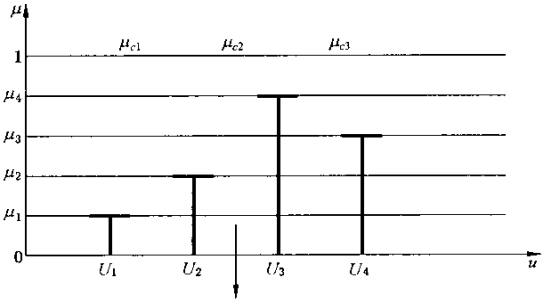
清晰化

假如输出值的隶属函数是单点集，其计算给定如下：

- 单点集重心法(CoGS)

$$U=\frac{\sum_{i=1}^{p}[U_i \cdot \mu_i]}{\sum_{i}^{p}[\mu_i]}$$

其中 μ 是综合的结果



面积中心线方法不适应于单点集

上图所示的四个单点集示例如下：

$$U=\frac{[U_1 \cdot \mu_1+U_2 \cdot \mu_2+U_3 \cdot \mu_3+U_4 \cdot \mu_4]}{[\mu_1+\mu_2+\mu_3+\mu_4]}$$

图 A.11d) 清晰化方法

A.3 模糊控制的性能

从信息技术的观点来看，模糊控制是一种基于语言规则的专家系统。从控制系统技术的观点来看，它是属于非线性特征领域控制器。除了没有规则被激发又没有定义缺省值的情况之外，输出变量的当前值只取决于输入变量的当前值，而与过去的值无关。如果控制器要具备动态特点，则必须在模糊功能块之外给出相关的动态函数。

这些函数一般是一阶微分和积分元件。它们的输出变量是模糊控制的附加输入变量。这对于控制偏差也同样适用，也要在模糊控制功能块以外形成。反之，模糊控制的输出变量可以传给对校正变量进一步处理的运算器，例如，给速度算法一个积分元件，或分配到不同的控制元件中。

图 A.12a)给出了模糊控制器的基本结构，而图 A.12b)给出一个例子：过程变量和设定值之差形成了控制偏差，该偏差和其时间导数作为两个输入变量送给模糊控制系统。将输出变量对时间积分可以得出校正变量。如果规则库设计成带有一个速度算法，则模糊控制器表现出与 PI 控制器相似的动态特性。

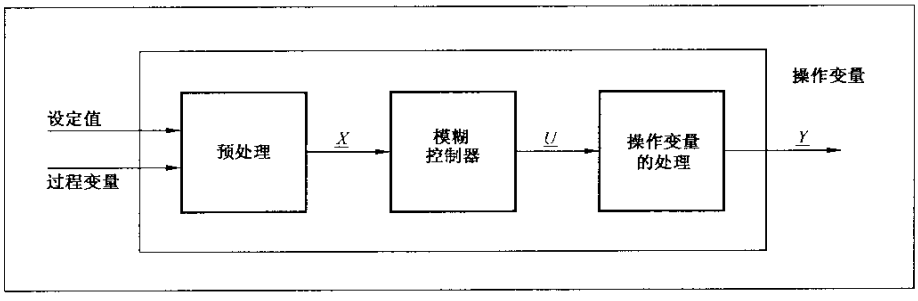


图 A.12a) 模糊控制器：基本结构

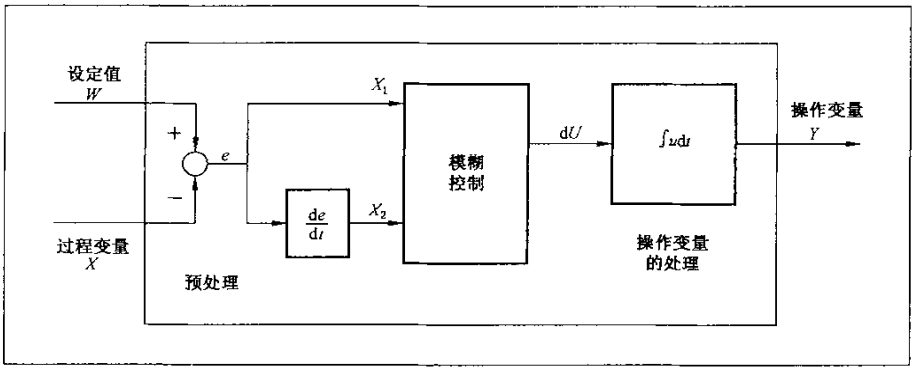


图 A.12b) 模糊控制器示例

附 录 B
（资料性附录）
示例

在可编程控制器范围内模糊控制的主要应用领域之一是与传统 PID 控制器相结合以便改进 PID 控制器的控制品质。下面的这些例子提供了一些可能应用模糊控制的地方。

B.1 预先控制

模糊控制器通过对被操作值提供一个校正信号来补充传统闭环控制器(见图 B.1)。

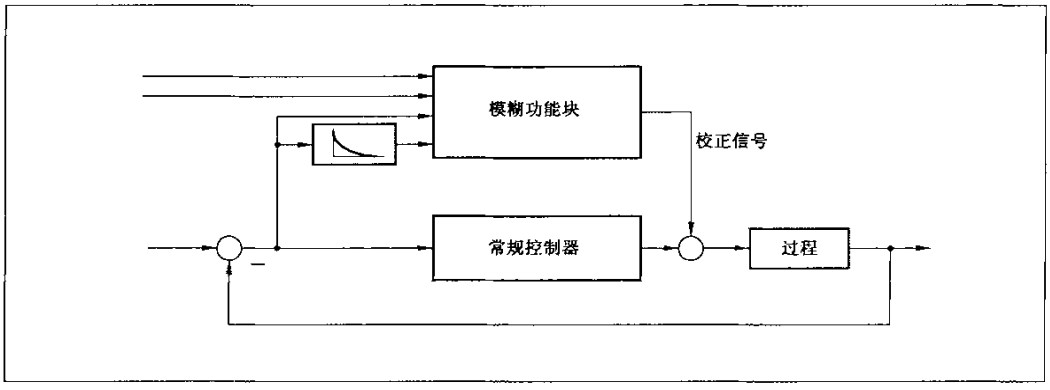


图 B.1 预先控制示例

B.2 常规 PID 控制器的参数自调整

模糊控制器可用来自适应调整 PID 控制器参数(见图 B.2)。

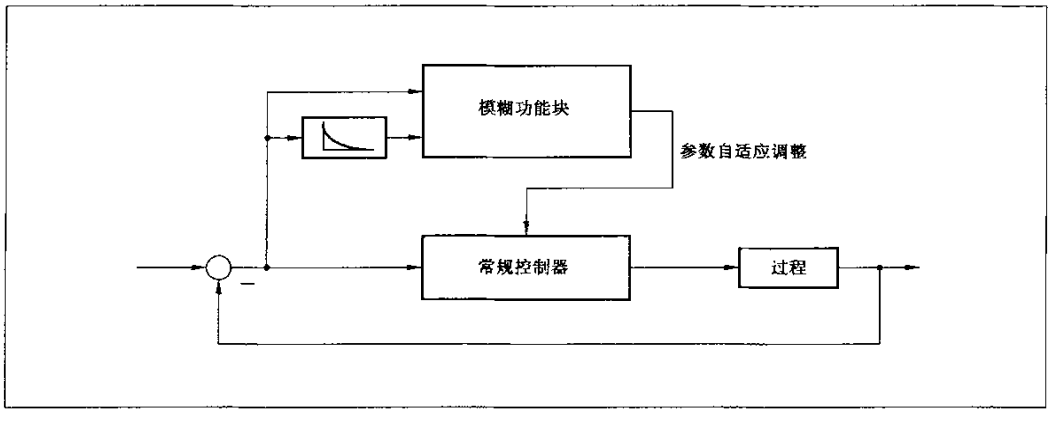


图 B.2 参数自适应调整举例

B.3 过程的直接模糊控制

模糊控制的另一个应用领域是把生产过程的经验知识和语言控制策略直接用于工业自动化中。它可应用于需要操作员干预的许多生产过程(见图 B.3)。

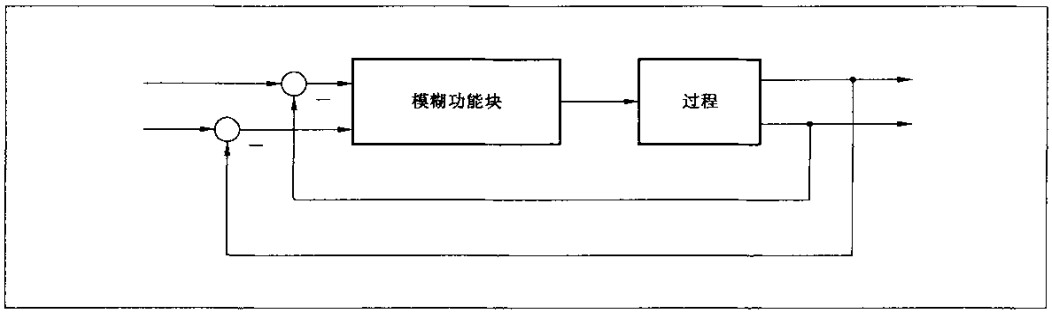


图 B.3 直接模糊控制示例

附 录 C
（资料性附录）
工业实例：集装箱吊车

大多数港口集装箱的装卸都采用集装箱吊车。它用安装在吊车头上的缆绳吊起集装箱。吊车头在水平轨道上移动。一个集装箱被吊起后吊车头就开始移动。集装箱开始如图 C.1 所示那样摇摆。在运送过程中出现摇摆不是什么大问题，但摇摆着的集装箱放不下来。为使集装箱快速精确定位，操作者凭经验来控制吊车头的运动策略。

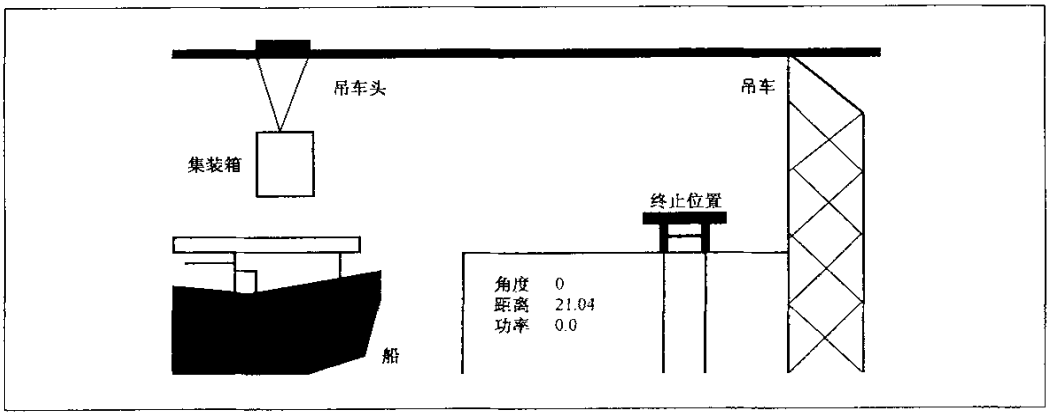


图 C.1 工业实例：集装箱吊车

操作者的动作分析表明，操作者运用的“经验规律”可描述成如下的控制策略：

a) 开始用中等的功率。

b) 如果开始后还离目的地很远，调整电动机功率使集装箱稍落后于吊车头。

c) 靠近目的地时，降低功率而使集装箱稍前于吊车头。

d) 当集装箱靠近目的地时，根据摇摆方向，使用+/−中等功率。

e) 集装箱在目的地上方，摇摆为零时，停止电动机运行。

为自动控制该吊车，使用了吊车头位置（“距离”）和集装箱摇摆角度（“角度”）传感器。输出是电动机功率。首先，对所有变量须定义其语言变量。语言变量距离、角度和电动机功率（见图 C.2、图 C.3、图 C.4）均划分成五个语言项。隶属函数使用的形式有上（下）界型、三角型和单点集。语言变量和语言项如下图所示。

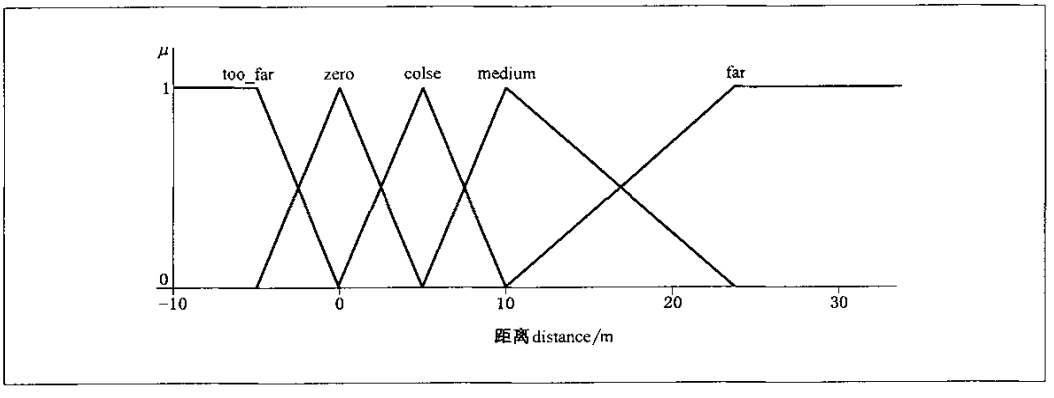


图 C.2 吊车头与目的地之间的“距离”语言变量

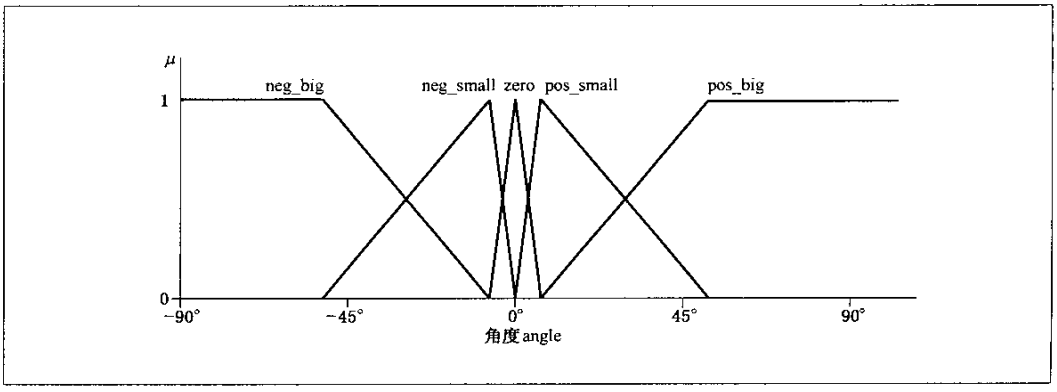


图 C.3 集装箱和吊车头形成的“角度”语言变量

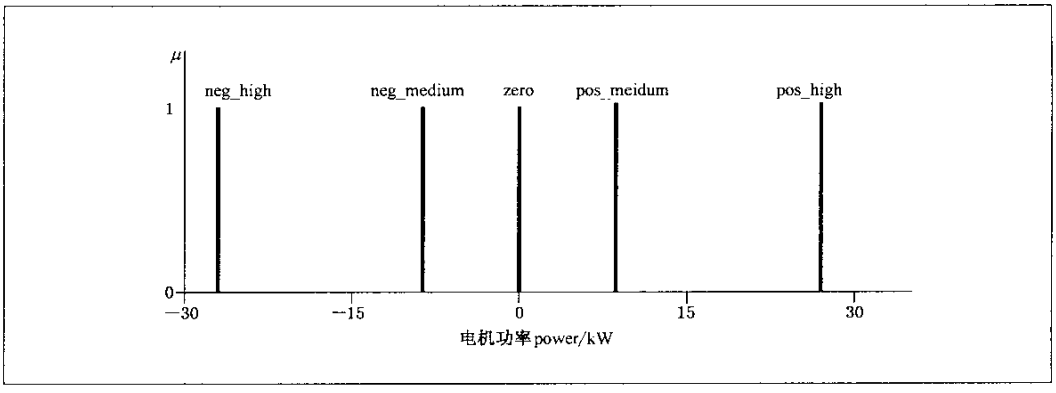


图 C.4 “功率”语言变量

用这些语言项来描述当前吊车的情况，规则库由 5 条经验规则组成。图 C.5 示出了用 FCL 概念定义的规则库。

RULE1：IF distance IS far	AND angle IS zero	THEN power IS pos_medium
RULE2：IF distance IS far	AND angle IS neg_small	THEN power IS pos_big
RULE3：IF distance IS far	AND angle IS neg_big	THEN power IS pos_medium
RULE4：IF distance IS medium	AND angle IS pos_small	THEN power IS neg_medium
RULE5：IF distance IS close	AND angle IS pos_small	THEN power IS pos_medium
RULE6：IF distance IS zero	AND angle IS zero	THEN power IS zero

图 C.5 规则库

表 C.1 所示分别为推理步骤和应用的算子。

表 C.1 推理步骤和指定的运算符

推理步骤	算子
聚集	
AND	Minimum
激活	
IF…THEN 结论的变换	
	Minimum
综合	Maximum

考虑吊车当前的一种情况，这时吊车头与目的地的距离是 12m，集装箱的角度是+4°。为了说明情况，假定子集由三个规则组成。

图 C.6 和图 C.7 给出了这种情况下模糊化如何进行。

12 m 距离对应下述语言项模糊集的隶属度：

far　　对应隶属度为 0.1

medium　　对应隶属度为 0.9

close　　对应隶属度为 0

zero　　对应隶属度为 0

too_far　　对应隶属度为 0

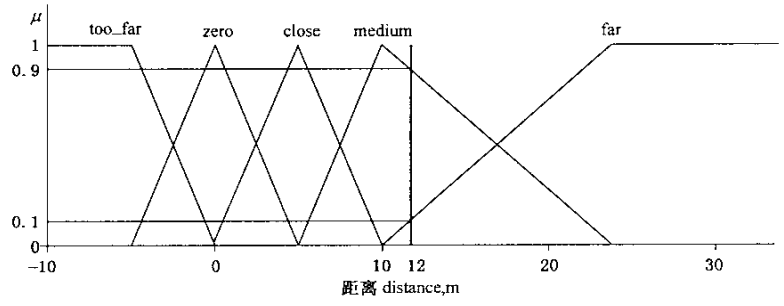


图 C.6 “距离”语言变量的模糊化

角度+4°对应下述语言项模糊集的隶属度：

neg_big　　对应隶属度为 0

neg_small　　对应隶属度为 0

zero　　对应隶属度为 0.2

pos_small　　对应隶属度为 0.8

pos_big　　对应隶属度为 0

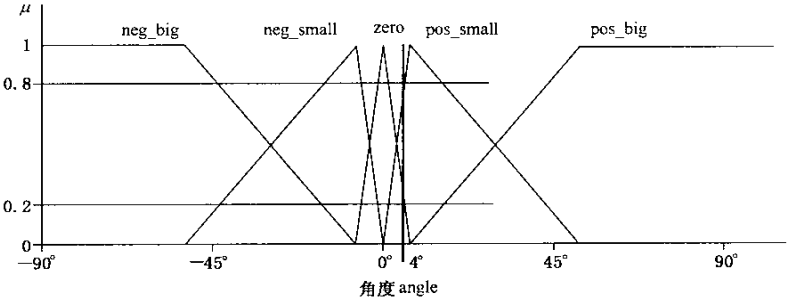


图 C.7 “角度”语言变量的模糊化

12 m 的距离被转化成语言变量值{0.1,0.9,0,0,0}，它可以说成是“中等，稍远”；4°的角度被转化成语言变量值{0,0,0.2,0.8,0}，可视为“正小，接近于零”。

推理

所有的输入变量都已转换成语言变量值之后，模糊推理步骤能够识别适用于当前状态的规则并计算出输出语言变量的值。图 C.8 给出了三条规则的子集来加以说明：

RULE1:IF distance IS medium	AND angle IS pos_small	THEN power IS pos_medium
RULE2:IF distance IS medium	AND angle IS zero	THEN power IS zero
RULE3:IF distance IS far	AND angle IS zero	THEN power IS pos_medium

图 C.8 三个规则的子集

推理包括三个部分——聚集、激活和综合。

——聚集(见图 C.10):

由基本语句的隶属度确定前件的符合程度。例子见图 C.9。

RULE 1:	distance = medium	AND	angle = pos_small	=P1
RULE 2:	distance = medium	AND	angle = zero	=P2
RULE 3:	distance = far	AND	angle = zero	=P3
	对所有前件			

图 C.9 聚集的元素

Min 算子对应于 AND 聚集。图 C.10 表明了这种情况下聚集是如何进行的。

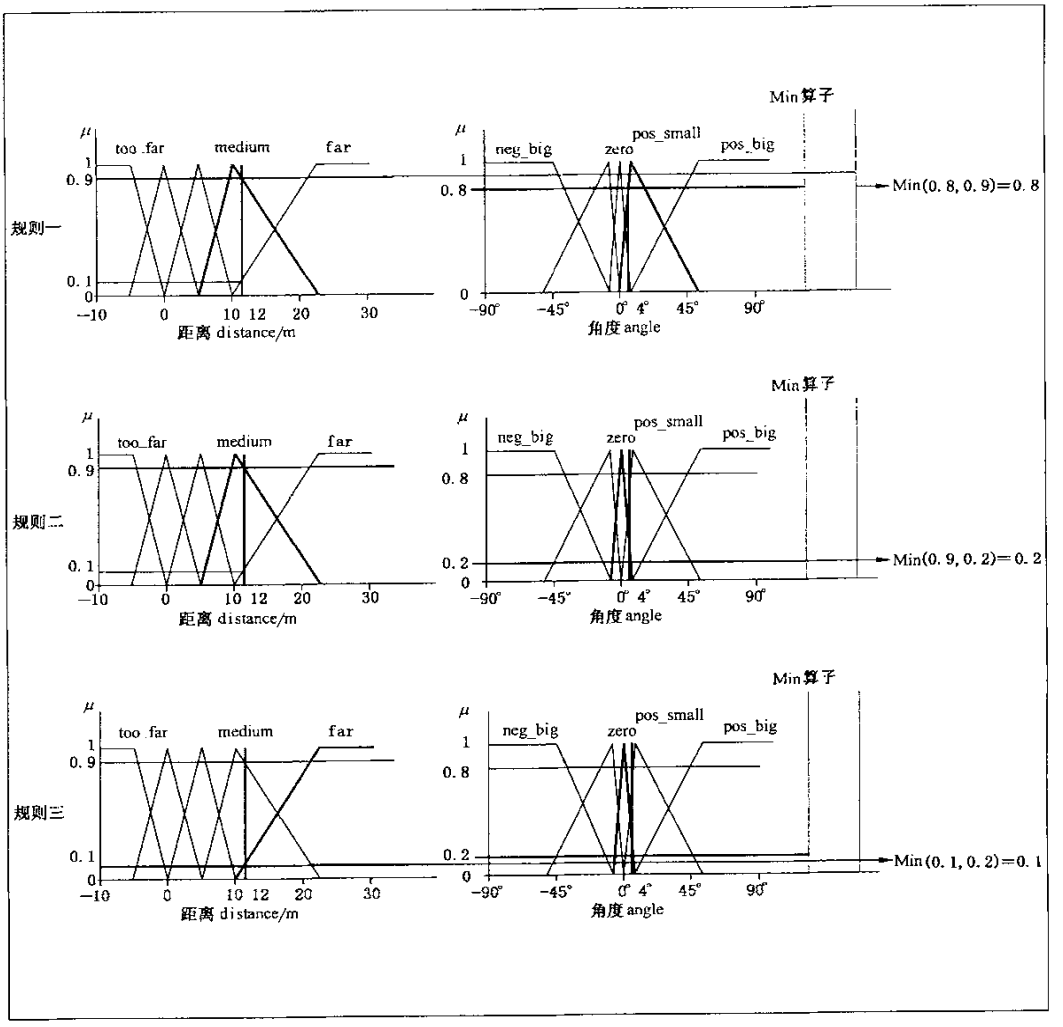


图 C.10 聚集原理

——激活(见图 C.12):

IF-THEN 结论的转换。例子见图 C.11。

图 C.11 激活的元素

图 C.12 表明了这种情况下激活是如何进行的。聚集的结果表示在左边,激活的结果表示在右边。

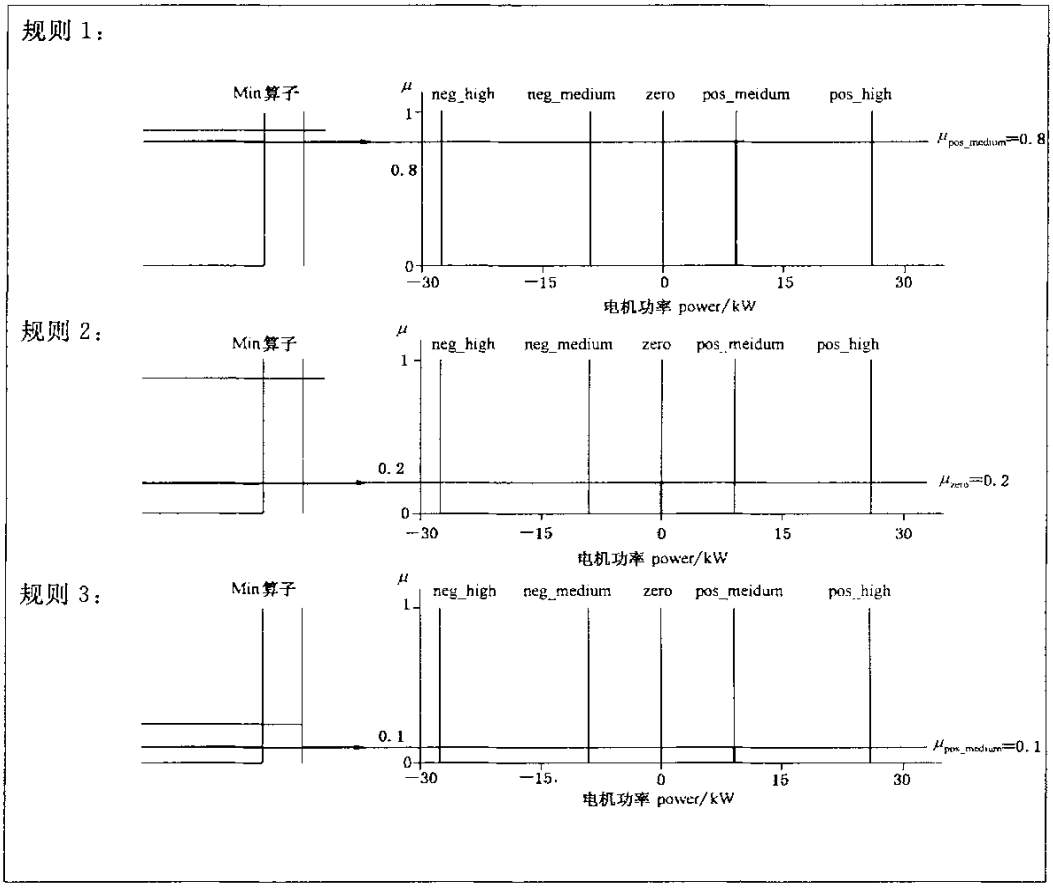


图 C.12 激活原理

——综合(见图 C.14):

规则的加权结果综合成一个总结果。例子见图 C.13。

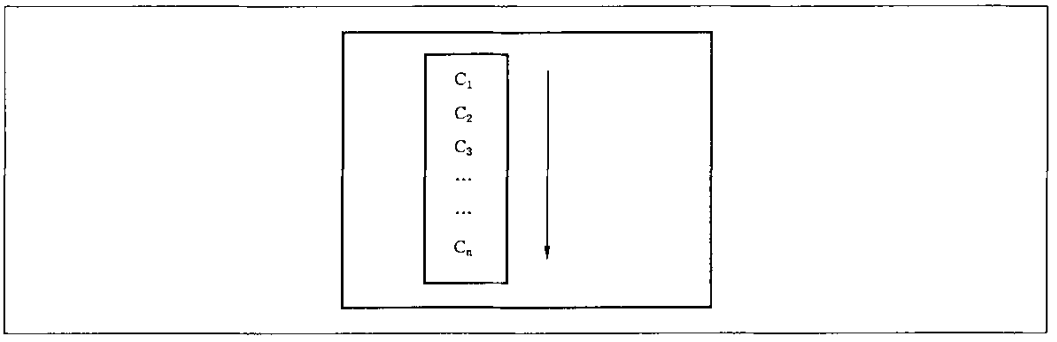


图 C.13 综合的元素

规则 1～3 的综合结果表示在图 C.14 的底部。例如：单点集“pos_medium”的结果计算成：Max (0.8,0.1)=0.8

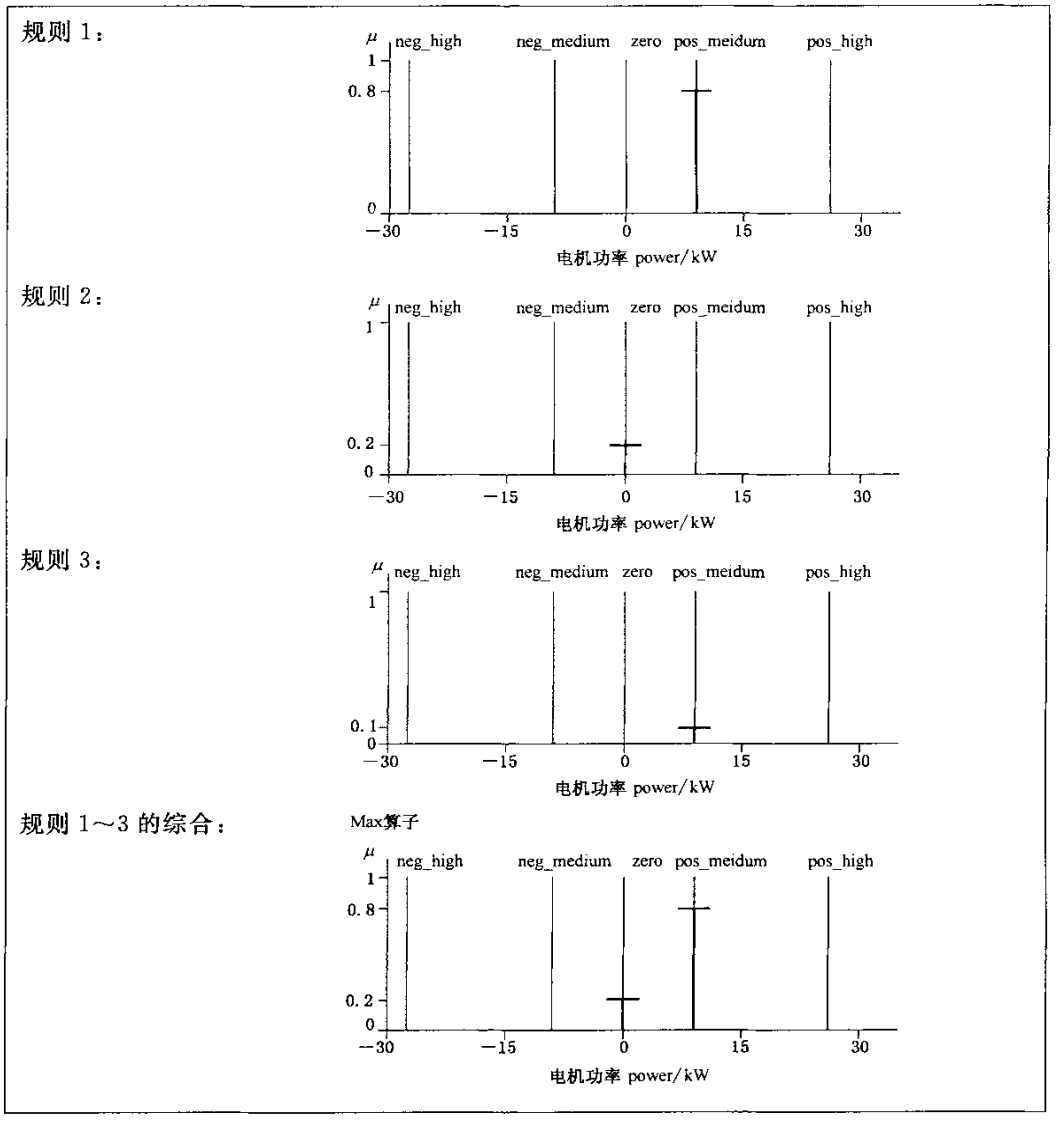
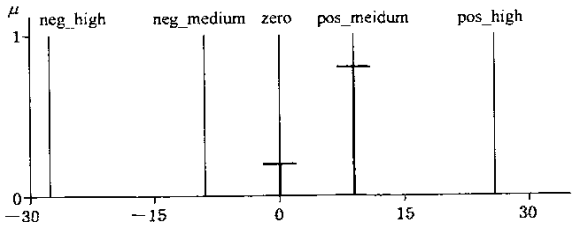


图 C.14 综合原理

清晰化(见图 C.15)

推理产生一个模糊集或它的隶属函数作为结果。要用它来设定电机功率,必须把它变换为清晰数值。从这个意义上说,待定值(通常为实数)应尽可能好地表征已求得的模糊集中所包含的信息。运用单点集重心法,输出变量清晰值——电机功率的计算如下:

规则 1~3 的综合:



清晰化:单点集重心法(CoGS)

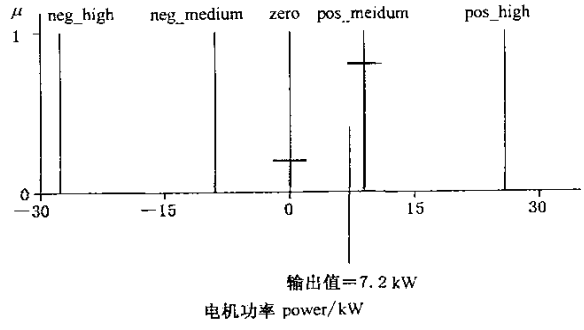


图 C.15 清晰化

用 FCL 语言描述的集装箱吊车模糊控制的例子(见图 C.16):

```
FUNCTION_BLOCK container_crane

VAR_INPUT
    distance:       REAL;
    angle:          REAL;
END_VAR

VAR_OUTPUT
    power:      REAL;
END_VAR

FUZZIFY distance
TERM        too_far         :=      (-5,1),(0,0);
TERM        zero            :=      (-5,0),(0,1),(5,0);
TERM        close           :=      (0,0),(5,1),(10,0);
TERM        medium          :=      (5,0),(10,1),(22,0);
TERM        far             :=      (10,0),(22,1);
END_FUZZIFY
```

图 C.16 FCL 应用实例

```
FUZZIFY angle
    TERM    neg_big      :=   (-50,1),(-5,1);
    TERM    neg_small    :=   (-50,0),(-5,1),(0,0);
    TERM    zero         :=   (-5,0),(0,1),(5,0);
    TERM    pos_small    :=   (0,0),(5,1),(50,0);
    TERM    pos_big      :=   (5,0),(50,1);
END_FUZZIFY

DEFUZZIFY power
    TERM    neg_high     :=    -27;
    TERM    neg_medium   :=    -9;
    TERM    zero         :=    0;
    TERM    pos_medium   :=    9;
    TERM    pos_high     :=    27;
    METHOD:CcGS;
    DEFAULT:=0;
END_DEFUZZIFY

RULEBLOCK No1
    AND:MIN;
    ACCU:MAX;
    RULE1:IF distance IS far      AND angle IS zero      THEN power IS pos_medium;
    RULE2:IF distance IS far      AND angle IS neg_small THEN power IS pos_big;
    RULE3:IF distance IS far      AND angle IS neg_big   THEN power IS pos_medium;
    RULE4:IF distance IS medium   AND angle IS pos_small THEN power IS neg_medium;
    RULE5:IF distance IS close    AND angle IS pos_small THEN power IS pos_medium;
    RULE6:IF distance IS zero     AND angle IS zero      THEN power IS zero;
END_RULEBLOCK

END_FUNCTION_BLOCK
```

图 C.16（续）

FCL 功能块应按 GB/T 15969.3—2005 规定的程序调用方式调用。其所以如此是因为模糊块仅能通过变量来与外部接口。

根据 GB/T 15969.3—2005 编程语言，上述例子的调用如下所示：

```
container_crane(distance:=INP_DIS, angle:=INP_ANG);
A:=container_crane.power;
```

变量 INP_DIS 和 INP_ANG 的值可以直接在控制器的输入语句中给出或由其他值计算得出。

附 录 D
（资料性附录）
在规则块中使用变量的例子

饼干在管状炉中烤制，饼干的颜色由安放在炉子末端的颜色传感器检测。颜色是三维测量值，用模糊分级方法来评定所测颜色归属于褐、亮、暗3挡的隶属度。湿度也在炉内测量。炉子由两个温度回路控制，一个用于前半部，另一个用于后半部。

图D.1和D.2给出了控制原理。

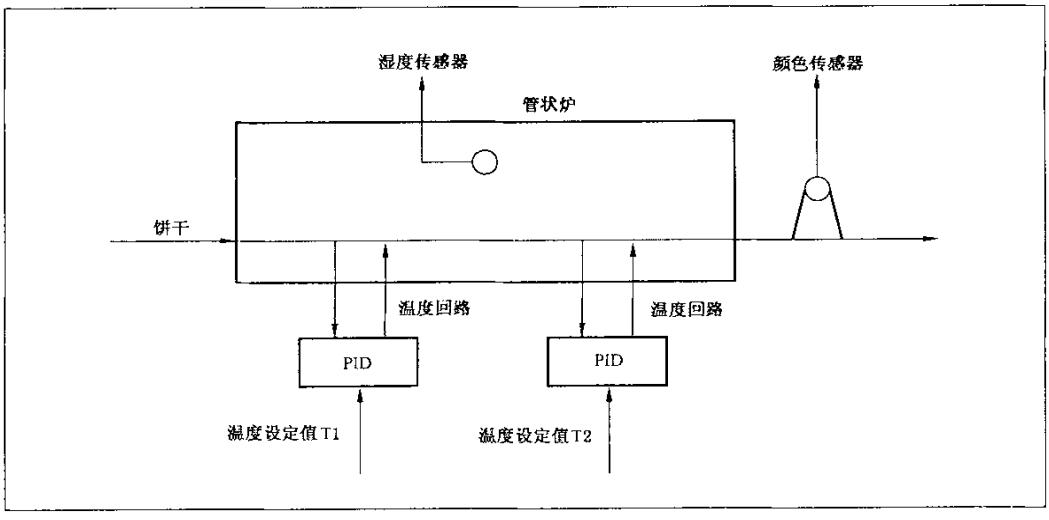


图 D.1 受控系统原理

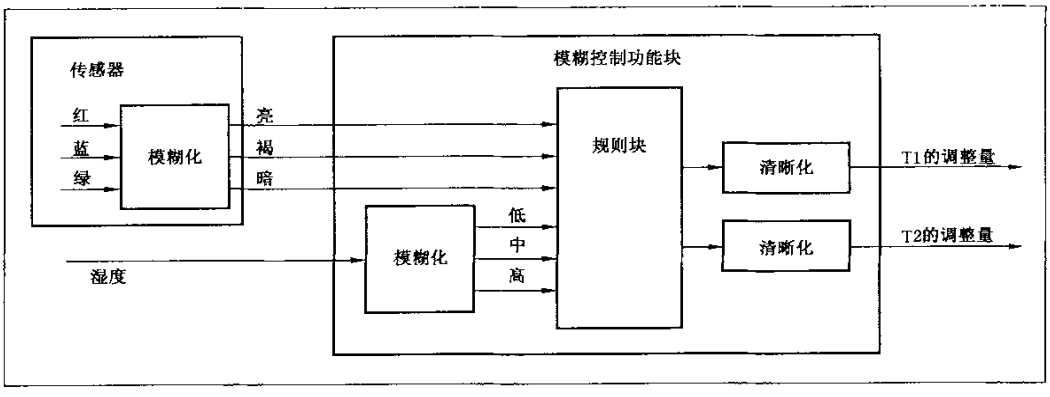


图 D.2 烤炉的模糊控制原理

规则块包括以下5条规则（见图D.3）：

IF humidity IS middle AND color IS brown	THEN dT1	IS zero AND dT2 IS zero
IF humidity IS high	TEHN dT1	IS positive
IF humidity IS low	THEN dT1	IS negative
IF humidity IS middle AND color IS light	THEN dT2	IS positive
IF humidity IS middle AND color IS dark	THEN dT2	IS negative

图 D.3 规则块

图 D.4 给出了此例 FCL 语法规则。

注：取代亮、褐、暗 3 变量，而用 GB/T 15969.3—2005 中规定的列举数据类型 color 一个变量也可以。在此例中，若 humidity 只取 high 或 low，则 dT2 无定义。

```
TYPE
STRUCT color_type
      brown:    REAL;
      light:    REAL;
      dark:     REAL;
END_STRUCT
END_TYPE

FUNCTION_BLOCK oven_control
VAR_INPUT
      humidity: REAL;
      color:    color_type;
END_VAR

VAR_OUTPUT
      dT1:      REAL;
      dT2:      REAL;
END_VAR

FUZZIFY humidity
    TERM low    :=(30,1),(50,0);
    TERM middle :=(30,0),(50,1),(70,1),(80,0);
    TERM high   :=(70,0),(80,1);
END_FUZZIFY

DEFUZZIFY dT1
    TERM negative:=-5;
    TERM zero    :=0;
    TERM positive:=5;
    METHOD:CoGS;
    DEFAULT:=0;
END_DEFUZZIFY

DEFUZZIFY dT2
    TERM negative  :=-3;
    TERM zero      :=0;
    TERM positive  :=3;
    METHOD:CoGS;
    DEFAULT:=0;
END_DEFUZZIFY
```

图 D.4 FCL 示例

```
RULEBLOCK inference
    AND:MIN;
    ACCU:MAX;
    RULE1:IF humidity IS middle AND color=brown    THEN dT1 IS zero AND dT2 IS zero;
    RULE2:IF humidity IS high                      THEN dT1 IS positive;
    RULE3:IF humidity IS low                       THEN dT1 IS negative;
    RULE4:IF humidity IS middle AND color=light    THEN dT2 IS positive;
    RULE5:IF humidity IS middle AND color=dark     THEN dT2 IS negative;
END_RULEBLOCK
END_FUNCTION_BLOCK
```

图 D.4（续）

附 录 E
（资料性附录）
符号、缩写、同义词

表 E.1 符号、缩写

CoA	面积中心线
CoG	重心法
CoGS	单点集重心法
FB	功能块
FBD	功能块图
FCL	模糊控制语言
IL	指令表
ISO	国际标准化组织
MAX	取大算子
MIN	取小算子
PROD	乘法算子
ST	结构化文本
μ	隶属度
ω	加权因子

表 E.2 同义词

Conclusion	Consequent
Accumulation	Result aggregation
Activation	Composition
Condition	Antecedent
Centre of Gravity	Centroid of Area
Centre of Area	Bisector of Area