

3.1 模糊控制的基本原理

3.1.1 模糊控制原理

模糊控制是以模糊集理论、模糊语言变量和模糊逻辑推理为基础的一种智能控制方法,它从行为上模仿人的模糊推理和决策过程。该方法首先将操作人员或专家经验编成模糊规则,然后将来自传感器的实时信号模糊化,将模糊化后的信号作为模糊规则的输入,完成模糊推理,将推理后得到的输出量加到执行器上。

模糊控制的基本原理框图如图 3.1 所示。它的核心部分为模糊控制器,如图中点画线框中部分所示,模糊控制器的控制规律由计算机的程序实现。实现一步模糊控制算法的过程描述如下:微型计算机经中断采样获取被控制量的精确值,然后将此量与给定值比较得到误差信号 E ,一般选误差信号 E 作为模糊控制器的一个输入量。把误差信号 E 的精确量进行模糊化,变成模糊量。误差 E 的模糊量可用相应的模糊语言表示,得到误差 E 的模糊语言集合的一个子集 \underline{e} (\underline{e} 是一个模糊矢量),再由 \underline{e} 和模糊控制规则 \underline{R} (模糊算子) 根据推理的合成规则进行模糊决策,得到模糊控制量 \underline{u}

$$\underline{u} = \underline{e} \circ \underline{R} \quad (3.1)$$

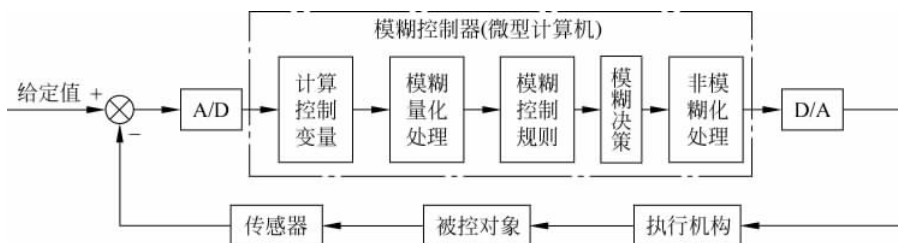


图 3.1 模糊控制原理框图

由图 3.1 可知,模糊控制系统与通常的计算机数字控制系统的主要差别是采用了模糊控制器。模糊控制器是模糊控制系统的核心,一个模糊控制系统的性能优劣,主要取决于模糊控制器的结构、所采用的模糊规则、合成推理算法以及模糊决策的方法等因素。

模糊控制器(Fuzzy Controller, FC)也称为模糊逻辑控制器(Fuzzy Logic Controller,

FLC), 由于所采用的模糊控制规则是由模糊理论中模糊条件语句来描述的, 因此模糊控制器是一种语言型控制器, 故也称为模糊语言控制器(Fuzzy Language Controller, FLC)。

3.1.2 模糊控制器的组成

模糊控制器的组成框图如图 3.2 所示。

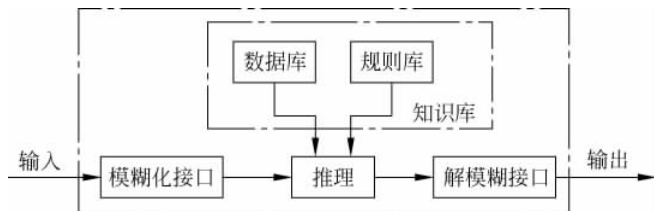


图 3.2 模糊控制器的组成框图

1. 模糊化接口(Fuzzy Interface)

模糊控制器的输入必须通过模糊化才能用于控制输出的求解, 因此它实际上是模糊控制器的输入接口。它的主要作用是将真实的确定量输入转换为一个模糊矢量。对于一个模糊输入变量 e , 其模糊子集通常可以做如下方式划分:

- (1) $e = \{\text{负大, 负小, 零, 正小, 正大}\} = \{\text{NB, NS, ZO, PS, PB}\}$ 。
- (2) $e = \{\text{负大, 负中, 负小, 零, 正小, 正中, 正大}\} = \{\text{NB, NM, NS, ZO, PS, PM, PB}\}$ 。
- (3) $e = \{\text{大, 负中, 负小, 零负, 零正, 正小, 正中, 正大}\} = \{\text{NB, NM, NS, NZ, PZ, PS, PM, PB}\}$ 。

用三角型隶属度函数表示如图 3.3 所示。

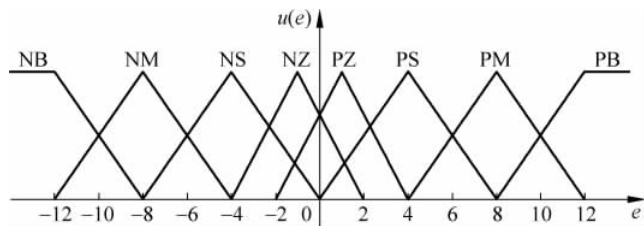


图 3.3 模糊子集和模糊化等级

2. 知识库(Knowledge Base, KB)

知识库由数据库和规则库两部分构成。

(1) 数据库(Data Base, DB)。数据库所存放的是所有输入、输出变量的全部模糊子集的隶属度矢量值(即经过论域等级离散化以后对应值的集合), 若论域为连续域则为隶属度函数。在规则推理的模糊关系方程求解过程中, 向推理机提供数据。

(2) 规则库(Rule Base, RB)。模糊控制器的规则是基于专家知识或手动操作人员长期积累的经验, 它是按人的直觉推理的一种语言表示形式。模糊规则通常由一系列的关系词连接而成, 如 if-then、else、also、end、or 等, 关系词必须经过“翻译”才能将模糊规则数值化。最常用的关系词为 if-then、also, 对于多变量模糊控制系统, 还有 and 等。例如, 某模糊控制系统输入变量为 e (误差) 和 ec (误差变化), 它们对应的语言变量为 E 和 EC , 可给出一组模糊规则:

R_1 : IF E is NB and EC is NB then U is PB

R_2 : IF E is NB and EC is NS then U is PM

通常把 if...部分称为“前提部”，而 then...部分称为“结论部”，其基本结构可归纳为 If A and B then C, 其中 A 为论域 U 上的一个模糊子集, B 是论域 V 上的一个模糊子集。根据人工控制经验,可离线组织其控制决策表 R , R 是笛卡儿乘积集 $U \times V$ 上的一个模糊子集, 则某一时刻其控制量由下式给出

$$C = (A \times B) \circ R \quad (3.2)$$

式中: \times ——模糊直积运算;

\circ ——模糊合成运算。

规则库是用来存放全部模糊控制规则的,在推理时为“推理机”提供控制规则。由上述可知,规则条数和模糊变量的模糊子集划分有关,划分越细,规则条数越多,但并不代表规则库的准确度越高,规则库的“准确性”还与专家知识的准确度有关。

3. 推理与解模糊接口(Inference and Defuzzy-interface)

推理是模糊控制器中,根据输入模糊量,由模糊控制规则完成模糊推理来求解模糊关系方程,并获得模糊控制量的功能部分。在模糊控制中,考虑到推理时间,通常采用运算较简单的推理方法。最基本的有 Zadeh 近似推理,它包含有正向推理和逆向推理两类。正向推理常被用于模糊控制中,而逆向推理一般用于知识工程学领域的专家系统中。

推理结果的获得,表示模糊控制的规则推理功能已经完成。但是,至此所获得的结果仍是一个模糊矢量,不能直接用来作为控制量,还必须做一次转换,求得清晰的控制量输出,即为解模糊。通常把输出端具有转换功能作用的部分称为解模糊接口。

综上所述,模糊控制器实际上就是依靠计算机(或单片机)来构成的,它的绝大部分功能都是由计算机程序来完成。随着专用模糊芯片的研究和开发,也可以由硬件逐步取代各组成单元的软件功能。

3.1.3 模糊控制系统的工作原理

以水位的模糊控制为例,如图 3.4 所示,设有一个水箱,通过调节阀可向内注水和向外抽水。设计一个模糊控制器,通过调节阀门将水位稳定在固定点附近。按照日常的操作经验,可以得到如下基本的控制规则:

- 若水位高于 O 点,则向外排水,差值越大,排水越快。
- 若水位低于 O 点,则向内注水,差值越大,注水越快。

根据上述经验,可按下列步骤设计一维模糊控制器。

1. 确定观测量和控制量

定义理想液位 O 点的水位为 h_0 , 实际测得的水位高度为 h , 选择液位差

$$e = \Delta h = h_0 - h$$

将当前水位对于 O 点的偏差 e 作为观测量。

2. 输入量和输出量的模糊化

将偏差 e 分为 5 级,分别为负大(NB)、负小(NS)、零(O)、正小(PS)和正大(PB),并根据偏差 e 的变化范围分为 7 个等

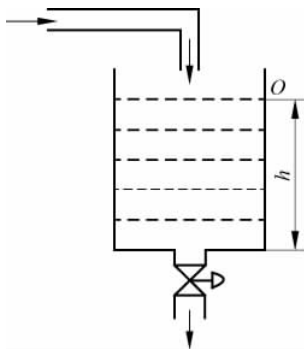


图 3.4 水箱液位控制

级,分别为-3、-2、-1、0、+1、+2 和+3,从而得到水位变化模糊表,如表 3.1 所示。

表 3.1 水位变化 e 划分表

隶 属 度		变 化 等 级						
		-3	-2	-1	0	1	2	3
模 糊 集	PB	0	0	0	0	0	0.5	1
	PS	0	0	0	0	1	0.5	0
	O	0	0	0.5	1	0.5	0	0
	NS	0	0.5	1	0	0	0	0
	NB	1	0.5	0	0	0	0	0

控制量 u 为调节阀门开度的变化,将其分为 5 级:负大(NB)、负小(NS)、零(O)、正小(PS)和正大(PB)。并根据 u 的变化范围分为 9 个等级:-4、-3、-2、-1、0、+1、+2、+3 和+4,从而得到控制量模糊划分表,如表 3.2 所示。

表 3.2 控制量 u 变化划分表

隶 属 度		变 化 等 级								
		-4	-3	-2	-1	0	1	2	3	4
模 糊 集	PB	0	0	0	0	0	0	0	0.5	1
	PS	0	0	0	0	0	0.5	1	0.5	0
	O	0	0	0	0.5	1	0.5	0	0	0
	NS	0	0.5	1	0.5	0	0	0	0	0
	NB	1	0.5	0	0	0	0	0	0	0

3. 模糊规则的描述

根据日常的经验,设计以下模糊规则

Rule1: 若 e 负大,则 u 正大

Rule2: 若 e 负小,则 u 正小

Rule3: 若 e 为 0,则 u 为 0

Rule4: 若 e 正小,则 u 负小

Rule5: 若 e 正大,则 u 负大

上述规则采用 IF A THEN B 的形式来描述

Rule1: if e =NB then u =NB

Rule2: if e =NS then u =NS

Rule3: if e =0 then u =0

Rule4: if e =PS then u =PS

Rule5: if e =PB then u =PB

根据上述经验规则,可得模糊控制表,如表 3.3 所示。

表 3.3 模糊控制规则表

若(IF)	NBe	NSe	Oe	PS e	PBe
则(THEN)	PBu	PSu	Ou	NSu	NBu

4. 求模糊关系

模糊控制规则是一个多条语句,它可以表示为 $U \times V$ 上的模糊子集,即模糊关系 R

$$R = (NBe \times NBu) \cup (NSE \times NSu) \cup (Oe \times Ou) \cup (PSe \times PSu) \cup (PBe \times PBu)$$

其中规则内的模糊集运算取交集,规则间的模糊集运算取并集。

$$\begin{aligned}
 NBe \times NBu &= \begin{bmatrix} 1 \\ 0.5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \times [1 \quad 0.5 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \\
 &= \begin{bmatrix} 1.0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 NSe \times NSu &= \begin{bmatrix} 0 \\ 0.5 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \times [0 \quad 0.5 \quad 1 \quad 0.5 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \\
 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 1.0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 Oe \times Ou &= \begin{bmatrix} 0 \\ 0 \\ 0.5 \\ 1.0 \\ 0.5 \\ 0 \\ 0 \\ 0 \end{bmatrix} \times [0 \quad 0 \quad 0 \quad 0.5 \quad 1 \quad 0.5 \quad 0 \quad 0 \quad 0]
 \end{aligned}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 1.0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$PSe \times PSu = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1.0 \\ 0.5 \\ 0 \end{bmatrix} \times [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0.5 \quad 1.0 \quad 0.5 \quad 0]$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 1.0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$PBe \times PBu = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.5 \\ 1.0 \end{bmatrix} \times [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0.5 \quad 1.0]$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1.0 \end{bmatrix}$$

由以上 5 个模糊矩阵求并集(即隶属函数最大值),得

$$R = \begin{bmatrix} 1.0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 1.0 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 1.0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 1.0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1.0 \end{bmatrix}$$

5. 模糊决策

模糊控制器的输出为误差向量和模糊关系的合成,即

$$u = e \circ R$$

当误差 e 为 NB 时, $e = [1.0 \ 0.5 \ 0 \ 0 \ 0 \ 0 \ 0]$, 控制器输出为

$$u = e \circ R = [1 \ 0.5 \ 0 \ 0 \ 0 \ 0 \ 0] \circ \begin{bmatrix} 1.0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 1.0 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 1.0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 1.0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1.0 \end{bmatrix}$$

$$= [1 \ 0.5 \ 0.5 \ 0.5 \ 0 \ 0 \ 0 \ 0 \ 0]$$

6. 控制量的反模糊化

由模糊决策可知,当误差为负大时,实际液位远高于理想液位, $e = \text{NB}$, 控制器的输出为一模糊向量,可表示为

$$u = \frac{1}{-4} + \frac{0.5}{-3} + \frac{0.5}{-2} + \frac{0.5}{-1} + \frac{0}{0} + \frac{0}{+1} + \frac{0}{+2} + \frac{0}{+3} + \frac{0}{+4}$$

如果按照“隶属度最大原则”进行反模糊化,选择控制量为 $u = -4$, 即阀门的开度应关小一些,减少进水量。

按上述步骤,设计水箱液位模糊控制的 MATLAB 仿真程序见 chap3_1.m。取 flag = 1, 可得到模糊系统的规则库并可实现模糊控制的动态仿真。模糊控制响应表如表 3.4 所示。取偏差 $e = -3$, 得 $u = -3.1481$ 。

表 3.4 模糊控制响应表

e	-3	-2	-1	0	1	2	3
u	-3	-2	-1	0	1	2	3

仿真程序: chap3_1.m

```
% Fuzzy Control for water tank
clear all;
close all;

a = newfis('fuzz_tank');
```

```

a = addvar(a, 'input', 'e', [-3, 3]); % Parameter e
a = addmf(a, 'input', 1, 'NB', 'zmf', [-3, -1]);
a = addmf(a, 'input', 1, 'NS', 'trimf', [-3, -1, 1]);
a = addmf(a, 'input', 1, 'Z', 'trimf', [-2, 0, 2]);
a = addmf(a, 'input', 1, 'PS', 'trimf', [-1, 1, 3]);
a = addmf(a, 'input', 1, 'PB', 'smf', [1, 3]);

a = addvar(a, 'output', 'u', [-4, 4]); % Parameter u
a = addmf(a, 'output', 1, 'NB', 'zmf', [-4, -1]);
a = addmf(a, 'output', 1, 'NS', 'trimf', [-4, -2, 1]);
a = addmf(a, 'output', 1, 'Z', 'trimf', [-2, 0, 2]);
a = addmf(a, 'output', 1, 'PS', 'trimf', [-1, 2, 4]);
a = addmf(a, 'output', 1, 'PB', 'smf', [1, 4]);

rulelist = [1 1 1 1; % Edit rule base
            2 2 1 1;
            3 3 1 1;
            4 4 1 1;
            5 5 1 1];

a = addrule(a, rulelist);

a1 = setfis(a, 'DefuzzMethod', 'mom'); % Defuzzy
writefis(a1, 'tank'); % Save to fuzzy file "tank.fis"
a2 = readfis('tank');

figure(1);
plotfis(a2);
figure(2);
plotmf(a, 'input', 1);
figure(3);
plotmf(a, 'output', 1);

flag = 1;
if flag == 1
    showrule(a) % Show fuzzy rule base
    ruleview('tank'); % Dynamic Simulation
end
disp(' ----- ');
disp(' fuzzy controller table:e = [-3, +3],u = [-4, +4] ');
disp(' ----- ');

for i = 1:1:7
    e(i) = i - 4;
    Ulist(i) = evalfis([e(i)], a2);
end
Ulist = round(Ulist)

e = -3; % Error
u = evalfis([e], a2) % Using fuzzy inference

```


3.1.4 模糊控制器结构

在确定性控制系统中,根据输入变量和输出变量的个数,可分为单变量控制系统和多变量控制系统。在模糊控制系统中也可类似地划分为单变量模糊控制和多变量模糊控制。

1. 单变量模糊控制器

在单变量模糊控制器(Single Variable Fuzzy Controller,SVFC)中,将其输入变量的个数定义为模糊控制的维数,如图 3.5 所示。

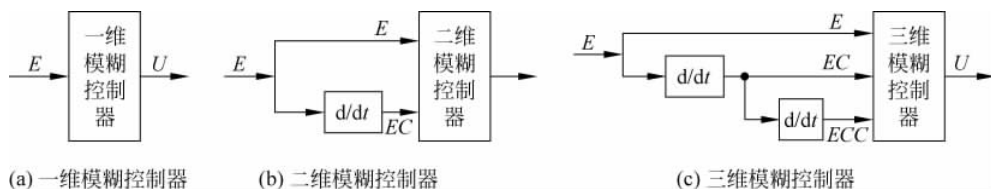


图 3.5 单变量模糊控制器

(1) 一维模糊控制器。

如图 3.5(a)所示,一维模糊控制器的输入变量往往选择为受控量和输入给定的偏差量 E 。由于仅仅采用偏差值,很难反映过程的动态特性品质,因此,所能获得的系统动态性能是不能令人满意的。这种一维模糊控制器往往被用于一阶被控对象。

(2) 二维模糊控制器。

如图 3.5(b)所示,二维模糊控制器的两个输入变量基本上都选用受控变量和输入给定的偏差 E 和偏差变化 EC ,由于它们能够较严格地反映受控过程中输出变量的动态特性,因此,在控制效果上要比一维控制器好得多,也是目前采用较广泛的一类模糊控制器。

(3) 三维模糊控制器。

如图 3.5(c)所示,三维模糊控制器的 3 个输入变量分别为系统偏差量 E 、偏差变化量 EC 和偏差变化的变化率 ECC 。由于这些模糊控制器结构较复杂,推理运算时间长,因此除非对动态特性的要求特别高的场合,一般较少选用三维模糊控制器。

上述三类模糊控制器的输出变量,均选择了受控变量的变化值。从理论上讲,模糊控制系统所选用的模糊控制器维数越高,系统的控制精度也就越高。但是维数选择太高,模糊控制规律就过于复杂,基于模糊合成推理的控制算法的计算机实现,也就更困难,这也许是人们在设计模糊控制系统时,多数采用二维控制器的原因。在需要时,为了获得较好的上升段特性和改善控制器的动态品质,也可以对模糊控制器的输出量做分段选择,即在偏差 E 为“大”时,以控制量的绝对值为输出,而当偏差 E 为“小”或“中等”时,则仍以控制量的增量为输出。

2. 多变量模糊控制器

一个多变量模糊控制器(Multiple Variable Fuzzy Controller)系统所采用的模糊控制器,往往具有多变量结构,称为多变量模糊控制器,如图 3.6 所示。

要直接设计一个多变量模糊控制器是相当困难的,可利用模糊控制器本身的解耦特点,通过模糊关系方程求解,在控制器结构上实现解耦,即将一个多输入-多输出(MIMO)的模

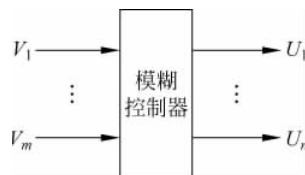


图 3.6 多变量模糊控制器

糊控制器,分解成若干个多输入-单输出(MISO)的模糊控制器,这样可采用单变量模糊控制器方法设计。

3.2 模糊控制系统分类

1. 按信号的时变特性分类

(1) 恒值模糊控制系统。

系统的指令信号为恒定值,通过模糊控制器消除外界对系统的扰动作用,使系统的输出跟踪输入的恒定值,也称为“自镇定模糊控制系统”,如温度模糊控制系统。

(2) 随动模糊控制系统。

系统的指令信号为时间函数,要求系统的输出高精度、快速地跟踪系统输入,也称为“模糊控制跟踪系统”或“模糊控制伺服系统”。

2. 按模糊控制的线性特性分类

对开环模糊控制系统 S , 设输入变量为 u , 输出变量为 v 。对任意输入偏差 Δu 和输出偏差 Δv , 满足 $\frac{\Delta v}{\Delta u} = k, u \in U, v \in V$ 。

定义线性度 δ , 用于衡量模糊控制系统的线性化程度

$$\delta = \frac{\Delta v_{\max}}{2\xi\Delta u_{\max}m} \quad (3.3)$$

其中, $\Delta v_{\max} = v_{\max} - v_{\min}$, $\Delta u_{\max} = u_{\max} - u_{\min}$, ξ 为线性化因子, m 为模糊子集 V 的个数。

设 k_0 为一经验值, 则定义模糊系统的线性特性为: (1) 当 $|k - k_0| \leq \delta$ 时, S 为线性模糊系统; (2) 当 $|k - k_0| > \delta$ 时, S 为非线性模糊系统。

3. 按静态误差是否存在分类

(1) 有差模糊控制系统。

将偏差的大小及其偏差变化率作为系统的输入为有差模糊控制系统。

(2) 无差模糊控制系统。

引入积分作用, 使系统的静差降至最小。

4. 按系统输入变量的多少分类

控制输入个数为 1 的系统为单变量模糊控制系统, 控制输入个数大于 1 的系统为多变量模糊控制系统。

3.3 模糊控制器的设计

3.3.1 模糊控制器的设计步骤

模糊控制器最简单的实现方法是將一系列模糊控制规则离线转化为一个查询表(又称为控制表), 存储在计算机中供在线控制时使用。这种模糊控制器结构简单, 使用方便, 是最基本的一种形式。本节以单变量二维模糊控制器为例, 介绍这种形式的模糊控制器的设计步骤, 其设计思想是设计其他模糊控制器的基础。具体设计步骤如下。

1. 模糊控制器的结构

单变量二维模糊控制器是最常见的结构形式。

2. 定义输入输出模糊集

例如,对误差 e 、误差变化 ec 及控制量 u 的模糊集及其论域定义如下:

e, ec 和 u 的模糊集均为: $\{NB, NM, NS, Z, PS, PM, PB\}$

e, ec 的论域均为: $\{-3, -2, -1, 0, 1, 2, 3\}$

u 的论域为: $\{-3.5, -3, -1.5, 0, 1, 3, 3.5\}$

3. 定义输入输出隶属函数

模糊变量误差 e 、误差变化 ec 及控制量 u 的模糊集和论域确定后,需对模糊语言变量确定隶属函数,即所谓对模糊变量赋值,就是确定论域内元素对模糊语言变量的隶属度。

4. 建立模糊控制规则

根据人的直觉思维推理,由系统输出的误差及误差的变化趋势来消除系统误差的模糊控制规则。模糊控制规则语句构成了描述众多被控过程的模糊模型。例如,卫星的姿态与作用的关系、飞机或舰船航向与舵偏角的关系和工业锅炉中的压力与加热的关系等。因此,在条件语句中,误差 e 、误差变化 ec 及控制量 u 对于不同的被控对象有着不同的意义。

5. 建立模糊控制表

上述描写的模糊控制规则可采用模糊规则表 3.5 来描述,共 49 条模糊规则,各个模糊语句之间是或的关系,由第 1 条语句所确定的控制规则可以计算出 u_1 。同理,可以由其余各条语句分别求出控制量 u_2, \dots, u_{49} ,则控制量为模糊集合 u ,可表示为

$$u = u_1 + u_2 + \dots + u_{49} \tag{3.4}$$

表 3.5 模糊规则表

u		e						
		NB	NM	NS	ZO	PS	PM	PB
ec	NB	PB	PB	PM	PM	PS	ZO	ZO
	NM	PB	PB	PM	PS	PS	ZO	NS
	NS	PM	PM	PM	PS	ZO	NS	NS
	ZO	PM	PM	PS	ZO	NS	NM	NM
	PS	PS	PS	ZO	NS	NS	NM	NM
	PM	PS	ZO	NS	NM	NM	NM	NB
	PB	ZO	ZO	NM	NM	NM	NB	NB

6. 模糊推理

模糊推理是模糊控制的核心,它利用某种模糊推理算法和模糊规则进行推理,得出最终的控制量。

7. 反模糊化

通过模糊推理得到的结果是一个模糊集合,但在实际模糊控制中,必须要有一个确定值才能控制或驱动执行机构。将模糊推理结果转化为精确值的过程称为反模糊化。常用的反模糊化有下面 3 种。

(1) 最大隶属度法。

选取推理结果模糊集合中隶属度最大的元素作为输出值,即 $v_0 = \max \mu_v(v), v \in V$ 。

如果在输出论域 V 中,其最大隶属度对应的输出值多于一个,则取所有具有最大隶属度输出的平均值,即

$$v_0 = \frac{1}{N} \sum_{i=1}^N v_i, \quad v_i = \max_{v \in V} (\mu_v(v)) \quad (3.5)$$

其中, N 为具有相同最大隶属度输出的总数。

最大隶属度法不考虑输出隶属度函数的形状,只考虑最大隶属度处的输出值。因此,难免会丢失许多信息。它的突出优点是计算简单,在一些控制要求不高的场合,可采用最大隶属度法。

(2) 重心法。

为了获得准确的控制量,就要求模糊方法能够很好地表达输出隶属度函数的计算结果。重心法是取隶属度函数曲线与横坐标围成面积的重心为模糊推理的最终输出值,即

$$v_0 = \frac{\int_V v \mu_v(v) dv}{\int_V \mu_v(v) dv} \quad (3.6)$$

对于具有 m 个输出量化级数的离散域情况

$$v_0 = \frac{\sum_{k=1}^m v_k \mu_v(v_k)}{\sum_{k=1}^m \mu_v(v_k)} \quad (3.7)$$

与最大隶属度法相比较,重心法具有更平滑的输出推理控制,即使输入信号有微小变化,输出也会发生变化。

(3) 加权平均法。

工业控制中广泛使用的反模糊方法为加权平均法,输出值由下式决定

$$v_0 = \frac{\sum_{i=1}^m v_i k_i}{\sum_{i=1}^m k_i} \quad (3.8)$$

其中,系数 k_i 的选择根据实际情况而定,不同的系数决定系统具有不同的响应特性,当系数 k_i 取隶属度 $\mu_v(v_i)$ 时,就转化为重心法。

反模糊化方法的选择与隶属度函数形状的选择、推理方法的选择相关。

MATLAB 提供了 5 种解模糊化方法,分别为 centroid(面积重心法)、bisector(面积等分法)、mom(最大隶属度平均法)、som(最大隶属度取小法)和 lom(最大隶属度取大法)。

在 MATLAB 中,可通过 setfis() 设置解模糊化方法,通过 defuzz() 执行反模糊化运算。例如,重心法通过下列程序来实现:

```
x = -10:1:10;
mf = trapmf(x, [-10, -8, -4, 7]);
xx = defuzz(x, mf, 'centroid');
```

在模糊控制中,重心法可通过下列语句来设定:

```
a1 = setfis(a, 'DefuzzMethod', 'centroid')
```

其中, a 为模糊规则库。

3.3.2 模糊控制器的 MATLAB 仿真

根据上述步骤,建立二输入单输出模糊控制系统,该系统包括两个部分,即模糊控制器的设计和位置跟踪。

1. 模糊控制器的设计

针对某线性系统的正弦跟踪,设计模糊规则表如表 3.6 所示,控制规则为 9 条。通过运行 showrule(a),可得到用于描述模糊系统的 9 条模糊规则。

表 3.6 模糊规则表

u		e		
		N	ZO	P
ec	N	N	ZO	ZO
	ZO	N	ZO	P
	P	ZO	ZO	P

模糊控制器的设计仿真程序见 chap3_2fuzz.m。在仿真时,根据模糊推理系统 a2,模糊规则可由命令 showrule(a2)得到

- Rule1: If (e is N) and (ec is N) then (u is N) (1)
- Rule2: If (e is N) and (ec is Z) then (u is N) (1)
- Rule3: If (e is N) and (ec is P) then (u is Z) (1)
- Rule4: If (e is Z) and (ec is N) then (u is Z) (1)
- Rule5: If (e is Z) and (ec is Z) then (u is Z) (1)
- Rule6: If (e is Z) and (ec is P) then (u is Z) (1)
- Rule7: If (e is P) and (ec is N) then (u is Z) (1)
- Rule8: If (e is P) and (ec is Z) then (u is P) (1)
- Rule9: If (e is P) and (ec is P) then (u is P) (1)

取误差 e 、误差变化 ec 的范围均为 $[-0.03, 0.03]$,控制输入 u 的范围为 $[-300, 300]$ 。模糊推理系统输入输出隶属函数可由命令 plotfis(a2)得到,如图 3.7~图 3.9 所示。

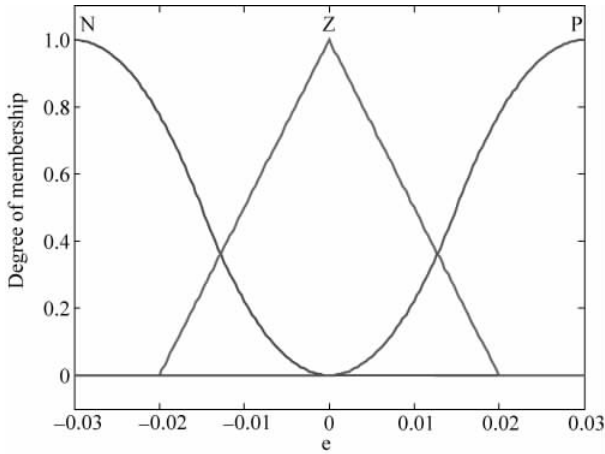


图 3.7 偏差隶属度函数

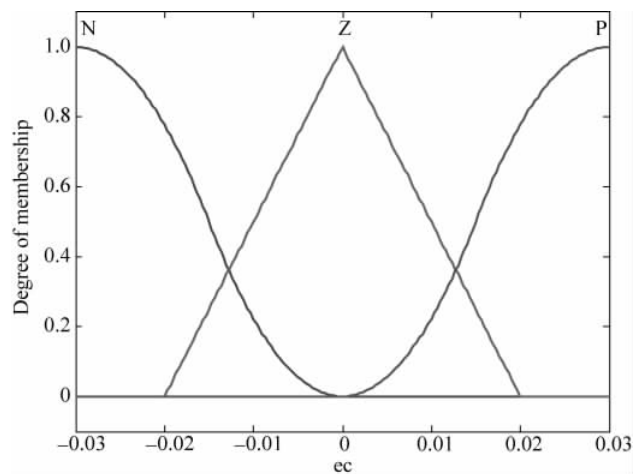


图 3.8 偏差变化率隶属度函数

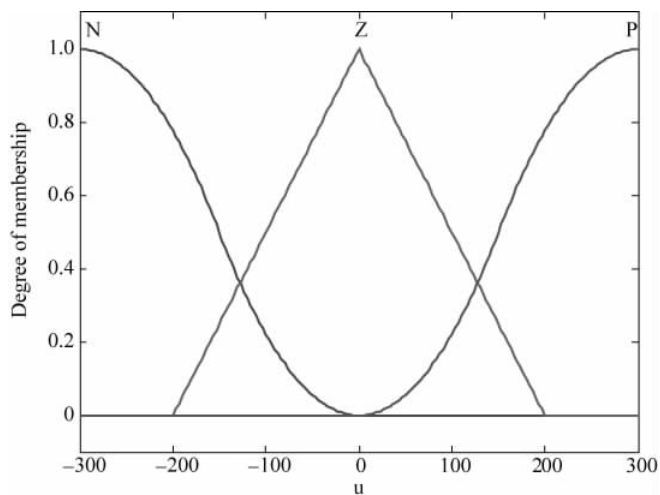


图 3.9 控制器输出隶属度函数

模糊系统设计仿真程序：chap3_2fuzz.m

```
% Fuzzy Controller Design
clear all;
close all;

a = newfis('fuzzf');

f1 = 0.010;
a = addvar(a,'input','e',[-f1*3,f1*3]);      % Parameter e
a = addmf(a,'input',1,'N','zmf',[-f1*3,0]);
a = addmf(a,'input',1,'Z','trimf',[-f1*2,0,f1*2]);
a = addmf(a,'input',1,'P','smf',[0,f1*3]);

f2 = 0.010;
```

```

a = addvar(a, 'input', 'ec', [- f2 * 3, f2 * 3]);      % Parameter ec
a = addmf(a, 'input', 2, 'N', 'zmf', [- f2 * 3, 0]);
a = addmf(a, 'input', 2, 'Z', 'trimf', [- f2 * 2, 0, f2 * 2]);
a = addmf(a, 'input', 2, 'P', 'smf', [0, f2 * 3]);

f3 = 100;
a = addvar(a, 'output', 'u', [- f3 * 3, f3 * 3]);      % Parameter u
a = addmf(a, 'output', 1, 'N', 'zmf', [- f3 * 3, 0]);
a = addmf(a, 'output', 1, 'Z', 'trimf', [- f3 * 2, 0, f3 * 2]);
a = addmf(a, 'output', 1, 'P', 'smf', [0, f3 * 3]);

rulelist = [1 1 1 1 1;                                % Edit rule base
            1 2 1 1 1;
            1 3 2 1 1;

            2 1 2 1 1;
            2 2 2 1 1;
            2 3 2 1 1;

            3 1 2 1 1;
            3 2 3 1 1;
            3 3 3 1 1];

a = addrule(a, rulelist);
showrule(a)                                             % Show fuzzy rule base

a1 = setfis(a, 'DefuzzMethod', 'mom');                % Defuzzy
writefis(a1, 'fuzzf');                                 % save to fuzzy file "fuzzf.fis" which can be
                                                        % simulated with fuzzy tool

a2 = readfis('fuzzf');

figure(1);
plotfis(a2);
figure(2);
plotmf(a, 'input', 1);
figure(3);
plotmf(a, 'input', 2);
figure(4);
plotmf(a, 'output', 1);

```

2. 模糊控制位置跟踪

被控对象为

$$G(s) = \frac{133}{s^2 + 25s}$$

首先运行模糊控制器程序 chap3_2fuzz.m, 并将模糊控制系统保存在 a2 之中, 然后运行模糊控制的 Simulink 仿真程序, 位置指令取 $\sin(t)$, 仿真结果如图 3.10 所示。

模糊控制位置跟踪的 Simulink 仿真程序见 chap3_2sim.mdl, 作图程序见 chap3_2plot.m。

仿真程序

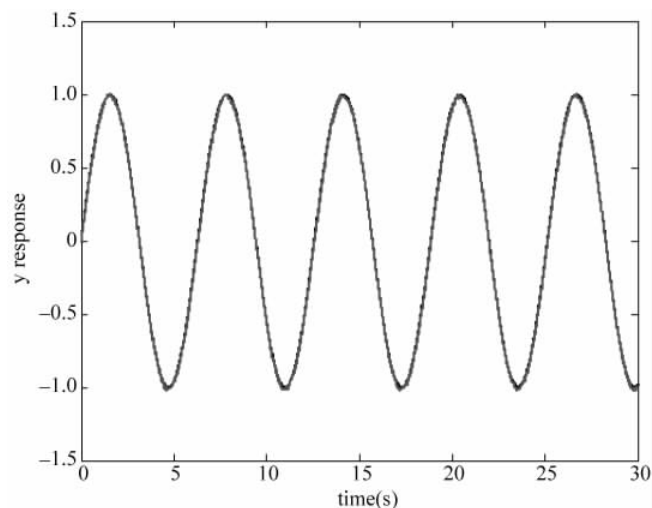
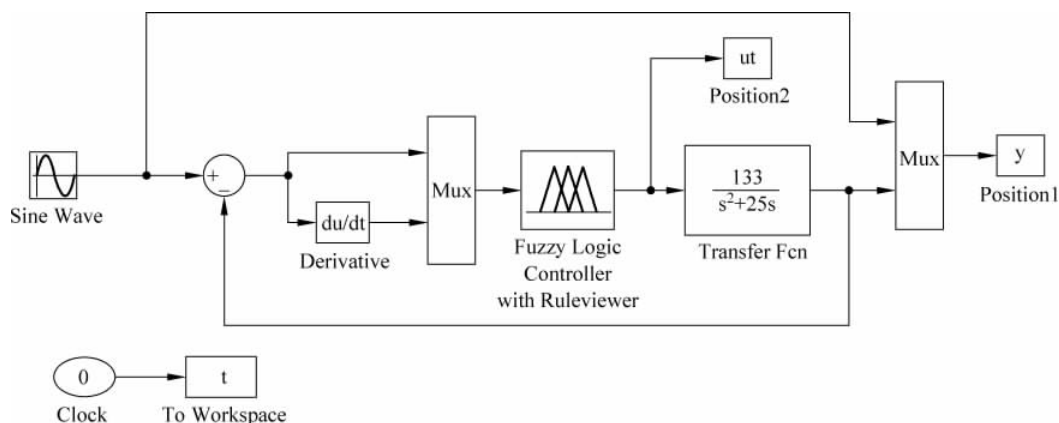


图 3.10 位置跟踪

Simulink 主程序: chap3_2sim.mdl



作图程序: chap3_2plot.m

```
close all;

figure(1);
plot(t,y(:,1),'k',t,y(:,2),'r','linewidth',2);
xlabel('time(s)');ylabel('y response');

figure(2);
plot(ut(:,1),'k','linewidth',2);
xlabel('x1');ylabel('ut');
```

3.4 模糊控制应用实例——洗衣机的模糊控制

下面以模糊洗衣机洗涤时间的模糊控制系统设计为例进行介绍,其控制是一个开环的决策过程,模糊控制按以下步骤进行。

1. 确定模糊控制器的结构

选用单变量二维模糊控制器,控制器的输入为衣物的污泥和油脂,输出为洗涤时间。

2. 定义输入输出模糊集

将污泥分为 3 个模糊集,即 SD(污泥少)、MD(污泥中)和 LD(污泥多),取值范围为[0, 100]。

3. 定义隶属函数

选用如下隶属函数

$$\mu_{\text{污泥}} = \begin{cases} \mu_{\text{SD}}(x) = (50 - x)/50 & 0 \leq x \leq 50 \\ \mu_{\text{MD}}(x) = \begin{cases} x/50 & 0 \leq x \leq 50 \\ (100 - x)/50 & 50 < x \leq 100 \end{cases} \\ \mu_{\text{LD}}(x) = (x - 50)/50 & 50 < x \leq 100 \end{cases}$$

采用三角形隶属函数可实现污泥的模糊化。采用 MATLAB 进行仿真,仿真程序见 chap3_3.m,仿真结果如图 3.11 所示。

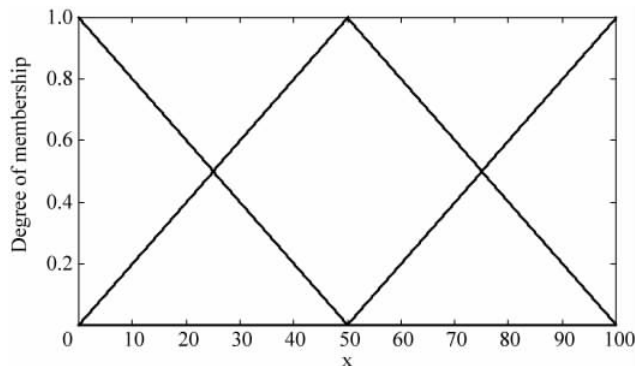


图 3.11 污泥隶属函数

污泥隶属函数设计仿真程序: chap3_3.m

```
% Define N + 1 triangle membership function
clear all;
close all;
N = 2;

x = 0:0.1:100;
for i = 1:N + 1
    f(i) = 100/N * (i - 1);
end

u = trimf(x, [f(1), f(1), f(2)]);
figure(1);
plot(x, u);

for j = 2:N
    u = trimf(x, [f(j - 1), f(j), f(j + 1)]);
```

```

hold on;
plot(x,u);
end
u = trimf(x,[f(N),f(N+1),f(N+1)]);
hold on;
plot(x,u);
xlabel('x');
ylabel('Degree of membership');

```

将油脂分为 3 个模糊集,分别为 NG(无油脂)、MG(油脂中)和 LG(油脂多),取值范围为 $[0,100]$ 。选用如下隶属函数

$$\mu_{\text{油脂}} = \begin{cases} \mu_{\text{NG}}(y) = (50 - y)/50 & 0 \leq y \leq 50 \\ \mu_{\text{MG}}(y) = \begin{cases} y/50 & 0 \leq y \leq 50 \\ (100 - y)/50 & 50 < y \leq 100 \end{cases} \\ \mu_{\text{LG}}(y) = (y - 50)/50 & 50 \leq y \leq 100 \end{cases}$$

采用三角形隶属函数实现油脂的模糊化,如图 3.12 所示,仿真程序同 chap3_3.m。

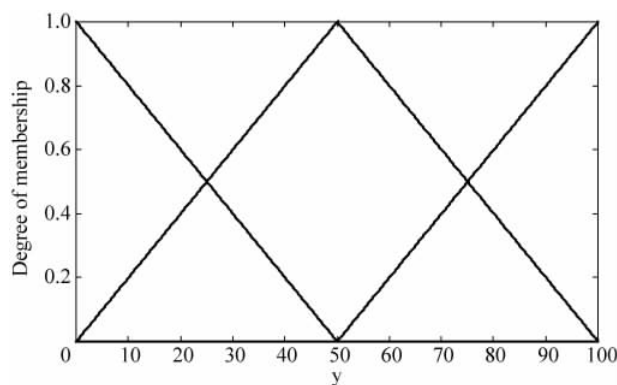


图 3.12 油脂隶属函数

将洗涤时间分为 5 个模糊集即 VS(很短)、S(短)、M(中等)、L(长)和 VL(很长),取值范围为 $[0,60]$,选用如下隶属函数

$$\mu_{\text{洗涤时间}} = \begin{cases} \mu_{\text{VS}}(z) = (10 - z)/10 & 0 \leq z \leq 10 \\ \mu_{\text{S}}(z) = \begin{cases} z/10 & 0 \leq z \leq 10 \\ (25 - z)/15 & 10 < z \leq 25 \end{cases} \\ \mu_{\text{M}}(z) = \begin{cases} (z - 10)/15 & 10 \leq z \leq 25 \\ (40 - z)/15 & 25 < z \leq 40 \end{cases} \\ \mu_{\text{L}}(z) = \begin{cases} (z - 25)/15 & 25 \leq z \leq 40 \\ (60 - z)/20 & 40 < z \leq 60 \end{cases} \\ \mu_{\text{VL}}(z) = (z - 40)/20 & 40 \leq z \leq 60 \end{cases}$$

采用三角形隶属函数实现洗涤时间的模糊化,如图 3.13 所示,仿真程序见 chap3_4.m。

采用 MATLAB 仿真,可实现洗涤时间隶属函数的设计。仿真程序见 chap3_4.m。

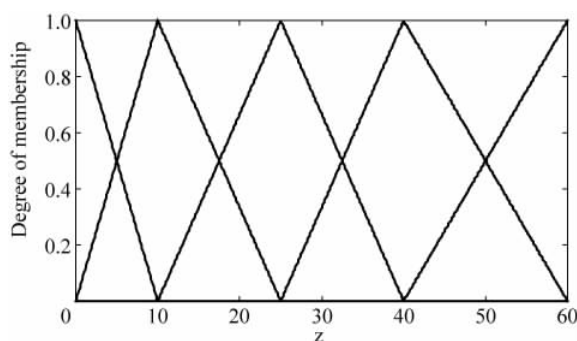


图 3.13 洗涤时间隶属函数

洗涤时间隶属函数设计仿真程序: chap3_4.m

```
% Define N+1 triangle membership function
```

```
clear all;
```

```
close all;
```

```
z = 0:0.1:60;
```

```
u = trimf(z,[0,0,10]);
```

```
figure(1);
```

```
plot(z,u);
```

```
u = trimf(z,[0,10,25]);
```

```
hold on;
```

```
plot(z,u);
```

```
u = trimf(z,[10,25,40]);
```

```
hold on;
```

```
plot(z,u);
```

```
u = trimf(z,[25,40,60]);
```

```
hold on;
```

```
plot(z,u);
```

```
u = trimf(z,[40,60,60]);
```

```
hold on;
```

```
plot(z,u);
```

```
xlabel('z');
```

```
ylabel('Degree of membership');
```

4. 建立模糊控制规则

根据人的操作经验设计模糊规则,模糊规则设计的标准为“污泥越多,油脂越多,洗涤时间越长”、“污泥适中,油脂适中,洗涤时间适中”以及“污泥越少,油脂越少,洗涤时间越短”。

5. 建立模糊控制表

根据模糊规则的设计标准,建立模糊规则表,如表 3.7 所示。

表 3.7 模糊洗衣机的洗涤规则

洗涤时间 z		污泥 x		
		NG	MG	LG
油脂 y	SD	VS*	M	L
	MD	S	M	L
	LD	M	L	VL

第 * 条规则为“IF 衣物污泥少且没有油脂 THEN 洗涤时间很短”。

6. 模糊推理

模糊推理分以下几步进行。

(1) 规则匹配。

假定当前传感器测得的信息为： x_0 (污泥)=60, y_0 (油脂)=70,分别代入所属的隶属函数中求隶属度

$$\mu_{MD}(60) = \frac{4}{5}, \quad \mu_{LD}(60) = \frac{1}{5}$$
$$\mu_{MG}(70) = \frac{3}{5}, \quad \mu_{LG}(70) = \frac{2}{5}$$

通过上述 4 种隶属度,可得到 4 条相匹配的模糊规则,如表 3.8 所示。

表 3.8 模糊推理结果

洗涤时间 z		污泥 x		
		NG	MG(3/5)	LG(2/5)
油脂 y	SD	0	0	0
	MD(4/5)	0	$\mu_M(z)$	$\mu_L(z)$
	LD(1/5)	0	$\mu_L(z)$	$\mu_{VL}(z)$

(2) 规则触发。

由表 3.8 可知,被触发的规则有如下 4 条

Rule1: IF y is MD and x is MG then z is M

Rule2: IF y is MD and x is LG then z is L

Rule3: IF y is LD and x is MG then z is L

Rule4: IF y is LD and x is LG then z is VL

(3) 规则前提推理。

在同一条规则内,前提之间通过“与”的关系得到规则结论,前提之间通过取小运算,得到每一条规则总前提的可信度:

Rule1 前提的可信度为 $\min(4/5, 3/5) = 3/5$

Rule2 前提的可信度为 $\min(4/5, 2/5) = 2/5$

Rule3 前提的可信度为 $\min(1/5, 3/5) = 1/5$

Rule4 前提的可信度为 $\min(1/5, 2/5) = 1/5$

由此得到洗衣机规则前提可信度表,即规则强度表,如表 3.9 所示。

表 3.9 规则前提可信度

洗涤时间 z		污泥 x		
		NG	MG(3/5)	LG(2/5)
油脂 y	SD	0	0	0
	MD(4/5)	0	3/5	2/5
	LD(1/5)	0	1/5	1/5

(4) 将上述两个表进行“与”运算。

得到每条规则总的输出,如表 3.10 所示。

表 3.10 规则总的可信度

洗涤时间 z		污泥 x		
		NG	MG(3/5)	LG(2/5)
油脂 y	SD	0	0	0
	MD(4/5)	0	$\min\left(\frac{3}{5}, \mu_M(z)\right)$	$\min\left(\frac{2}{5}, \mu_L(z)\right)$
	LD(1/5)	0	$\min\left(\frac{1}{5}, \mu_L(z)\right)$	$\min\left(\frac{1}{5}, \mu_{VL}(z)\right)$

(5) 模糊系统总的输出。

模糊系统总的输出为各条规则推理结果的并,即

$$\begin{aligned}\mu_{\text{agg}}(z) &= \max\left\{\min\left(\frac{3}{5}, \mu_M(z)\right), \min\left(\frac{2}{5}, \mu_L(z)\right), \min\left(\frac{1}{5}, \mu_L(z)\right), \min\left(\frac{1}{5}, \mu_{VL}(z)\right)\right\} \\ &= \max\left\{\min\left(\frac{3}{5}, \mu_M(z)\right), \min\left(\frac{2}{5}, \mu_L(z)\right), \min\left(\frac{1}{5}, \mu_{VL}(z)\right)\right\}\end{aligned}$$

(6) 反模糊化。

模糊系统总的输出 $\mu_{\text{agg}}(z)$ 实际上是 3 个规则推理结果的并集,需要进行反模糊化,才能得到精确的推理结果。下面以最大平均法为例,进行反模糊化。

将 $\mu = \frac{3}{5}$ 带入洗涤时间隶属函数中的 $\mu_M(z)$, 得到规则前提隶属度 $\mu = \frac{3}{5}$ 与规则结论隶属度 $\mu_M(z)$ 的交点

$$\mu_M(z) = \frac{z-10}{15} = \frac{3}{5}, \quad \mu_M(z) = \frac{40-z}{15} = \frac{3}{5}$$

得: $z_1 = 19, z_2 = 31$ 。

采用最大平均法,可得精确输出

$$z^* = \frac{z_1 + z_2}{2} = \frac{19 + 31}{2} = 25$$

采用 MATLAB 中模糊控制工具箱可设计洗衣机模糊控制系统。洗衣机模糊控制系统仿真程序见 chap3_5.m。

取 $x=60, y=70$, 反模糊化采用重心法, 模糊推理结果为 33.6853。利用命令 showrule 可观察规则库, 利用命令 ruleview 可实现模糊控制的动态仿真, 如图 3.14 所示。

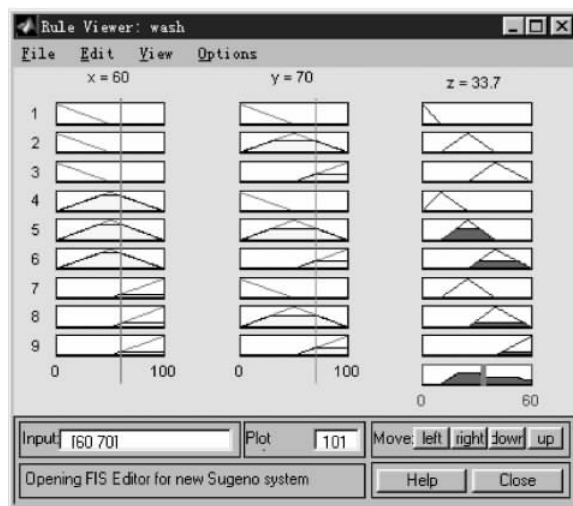


图 3.14 动态仿真模糊系统

洗衣机模糊控制系统仿真程序：chap3_5.m

```
% Fuzzy Control for washer
clear all;
close all;

a = newfis('fuzz_wash');

a = addvar(a, 'input', 'x', [0,100]);          % Fuzzy Stain
a = addmf(a, 'input', 1, 'SD', 'trimf', [0,0,50]);
a = addmf(a, 'input', 1, 'MD', 'trimf', [0,50,100]);
a = addmf(a, 'input', 1, 'LD', 'trimf', [50,100,100]);

a = addvar(a, 'input', 'y', [0,100]);          % Fuzzy Axunge
a = addmf(a, 'input', 2, 'NG', 'trimf', [0,0,50]);
a = addmf(a, 'input', 2, 'MG', 'trimf', [0,50,100]);
a = addmf(a, 'input', 2, 'LG', 'trimf', [50,100,100]);

a = addvar(a, 'output', 'z', [0,60]);          % Fuzzy Time
a = addmf(a, 'output', 1, 'VS', 'trimf', [0,0,10]);
a = addmf(a, 'output', 1, 'S', 'trimf', [0,10,25]);
a = addmf(a, 'output', 1, 'M', 'trimf', [10,25,40]);
a = addmf(a, 'output', 1, 'L', 'trimf', [25,40,60]);
a = addmf(a, 'output', 1, 'VL', 'trimf', [40,60,60]);

rulelist = [1 1 1 1 1;                      % Edit rule base
            1 2 3 1 1;
            1 3 4 1 1;

            2 1 2 1 1;
            2 2 3 1 1;
            2 3 4 1 1;
```

```

3 1 3 1 1;
3 2 4 1 1;
3 3 5 1 1];

a = addrule(a, rulelist);
showrule(a) % Show fuzzy rule base

a1 = setfis(a, 'DefuzzMethod', 'mom'); % Defuzzy
writefis(a1, 'wash'); % Save to fuzzy file "wash.fis"
a2 = readfis('wash');

figure(1);
plotfis(a2);
figure(2);
plotmf(a, 'input', 1);
figure(3);
plotmf(a, 'input', 2);
figure(4);
plotmf(a, 'output', 1);

ruleview('wash'); % Dynamic Simulation

x = 60;
y = 70;
z = evalfis([x, y], a2) % Using fuzzy inference

```

3.5 模糊自适应整定 PID 控制

3.5.1 模糊自适应整定 PID 控制原理

在工业生产过程中,许多被控对象随着负荷变化或干扰因素影响,其对象特性参数或结构发生改变。自适应控制运用现代控制理论在线辨识对象特征参数,实时改变其控制策略,使控制系统品质指标保持在最佳范围内,但其控制效果的好坏取决于辨识模型的精确度,这对于复杂系统是非常困难的。因此,在工业生产过程中,大量采用的仍然是 PID 算法, PID 参数的整定方法很多,但大多数都以对象特性为基础。

随着计算机技术的发展,人们利用人工智能的方法将操作人员的调整经验作为知识存入计算机中,根据现场实际情况,计算机能自动调整 PID 参数,这样就出现了专家 PID 控制器。这种控制器把古典的 PID 控制与先进的专家系统相结合,实现系统的最佳控制。这种控制必须精确地确定对象模型,首先将操作人员(专家)长期实践积累的经验知识用控制规则模型化,然后运用推理便可对 PID 参数实现最佳调整。

由于操作者经验不易精确描述,控制过程中各种信号量以及评价指标不易定量表示,专家 PID 方法受到局限。模糊理论为解决这一问题的有效途径,所以人们运用模糊数学的基本理论和方法,把规则的条件和操作用模糊集表示,并把这些模糊控制规则以及有关信息(如评价指标、初始 PID 参数等)作为知识存入计算机知识库中,然后计算机根据控制系统

的实际响应情况(即专家系统的输入条件),运用模糊推理,即可自动实现对 PID 参数的最佳调整,这就是模糊自适应 PID 控制。模糊自适应 PID 控制器目前有多种结构形式,但其工作原理基本一致。

自适应模糊 PID 控制器以误差 e 和误差变化 ec 作为输入(利用模糊控制规则在线对 PID 参数进行修改),以满足不同时刻的 e 和 ec 对 PID 参数自整定的要求。自适应模糊 PID 控制器结构如图 3.15 所示。

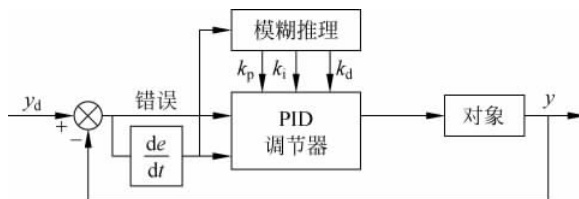


图 3.15 自适应模糊控制器结构

离散 PID 控制算法为

$$u(k) = k_p e(k) + k_i T \sum_{j=0}^k e(j) + k_d \frac{e(k) - e(k-1)}{T} \quad (3.9)$$

式中, k 为采样序号, T 为采样时间。

PID 参数模糊自整定是找出 PID 3 个参数与 e 和 ec 之间的模糊关系,在运行中通过不断检测 e 和 ec ,根据模糊控制原理对 3 个参数进行在线修改,以满足不同 e 和 ec 时对控制参数的不同要求,而使被控对象有良好的动态性能和静态性能。

从系统的稳定性、响应速度、超调量和稳态精度等各方面来考虑, k_p, k_i, k_d 的作用如下:

(1) 比例系数 k_p 的作用是加快系统的响应速度,提高系统的调节精度。 k_p 越大,系统的响应速度越快,系统的调节精度越高,但易产生超调,甚至会导致系统不稳定。 k_p 取值过小,则会降低调节精度,使响应速度缓慢,从而延长调节时间,使系统静态性能和动态性能变坏。

(2) 积分作用系数 k_i 的作用是消除系统的稳态误差。 k_i 越大,系统的静态误差消除越快。但 k_i 过大,在响应过程的初期会产生积分饱和现象,从而引起响应过程的较大超调。若 k_i 过小,将使系统静态误差难以消除,影响系统的调节精度。

(3) 微分作用系数 k_d 的作用是改善系统的动态特性,其作用主要是在响应过程中抑制偏差向任何方向的变化,对偏差变化进行提前预报。但 k_d 过大,会使响应过程提前制动,从而延长调节时间,而且会降低系统的抗干扰性能。

以 PI 参数整定为例,必须考虑到在不同时刻两个参数的作用以及相互之间的互联关系。模糊自整定 PI 是在 PI 算法的基础上,通过计算当前系统误差 e 和误差变化率 ec ,利用模糊规则进行模糊推理,查询模糊矩阵表进行参数调整。针对 k_p 和 k_i 两个参数分别整定的模糊控制表如下。

(1) k_p 整定原则。

当响应在上升过程时(e 为 P), Δk_p 取正,即增大 k_p ; 当超调时(e 为 N), Δk_p 取负,即降低 k_p 。当误差在 0 附近时(e 为 Z),分 3 种情况: ec 为 N 时,超调越来越大,此时 Δk_p 取负; ec 为 Z 时,为了降低误差, Δk_p 取正; ec 为 P 时,正向误差越来越大, Δk_p 取正。 k_p 整定的模糊规则表如表 3.11 所示。

表 3.11 k_p 的模糊规则表

Δk_p		ec		
e		N	Z	P
N		N	N	N
Z		N	P	P
P		P	P	P

(2) k_i 整定原则。

采用积分分离策略,即误差在 0 附近时, Δk_i 取正,否则 Δk_i 取 0。 k_i 整定的模糊规则表如表 3.12 所示。

表 3.12 k_i 的模糊规则表

Δk_i		ec		
e		N	Z	P
N		Z	Z	Z
Z		P	P	P
P		Z	Z	Z

将系统误差 e 和误差变化率 ec 变化范围定义为模糊集上的论域

$$e, ec = \{-1, 0, 1\} \quad (3.10)$$

其模糊子集为 $e, ec = \{N, O, P\}$,子集中元素分别代表负,零,正。设 e, ec 和 k_p, k_i 均服从正态分布,因此可得出各模糊子集的隶属度,根据各模糊子集的隶属度赋值表和各参数模糊控制模型,应用模糊合成推理设计 PI 参数的模糊矩阵表,查出修正参数代入式(3.11)计算

$$k_p = k_{p0} + \Delta k_p, \quad k_i = k_{i0} + \Delta k_i \quad (3.11)$$

在线运行过程中,控制系统通过对模糊逻辑规则的结果处理、查表和运算,完成对 PID 参数的在线自校正,其工作流程图如图 3.16 所示。

3.5.2 仿真实例

被控对象为

$$G_p(s) = \frac{133}{s^2 + 25s}$$

采样时间为 1ms,采用 Z 变换进行离散化,离散化后的被控对象为

$$y(k) = -\text{den}(2)y(k-1) - \text{den}(3)y(k-2) + \text{num}(2)u(k-1) + \text{num}(3)u(k-2)$$

位置指令为幅值为 1.0 的阶跃信号, $y_d(k) = 1.0$ 。仿真

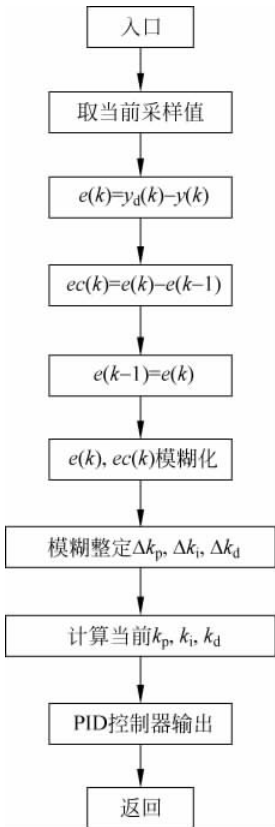


图 3.16 模糊 PID 工作流程图

时,先运行模糊推理系统设计程序 chap3_6.m,实现模糊推理系统 fuzzpid.fis,并将此模糊推理系统调入内存中,然后运行模糊控制程序 chap3_7.m。在程序 chap3_6.m 中,根据模糊规则如表 3.11~表 3.12 所示,分别对 e 、 ec 、 k_p 、 k_i 进行隶属函数的设计。根据位置指令、初始误差和经验设计 e 、 ec 、 k_p 、 k_i 的范围。

在 MATLAB 环境下,对模糊系统 a,运行 plotmf 命令,可得到模糊系统 e 、 ec 、 k_p 、 k_i 的隶属函数,如图 3.17~图 3.20 所示,运行命令 showrule 可显示模糊规则,可显示 9 条模糊规则,描述如下:

- Rule1: If (e is N) and (ec is N) then (k_p is N)(k_i is Z) (1)
 Rule2: If (e is N) and (ec is Z) then (k_p is N)(k_i is Z) (1)
 Rule3: If (e is N) and (ec is P) then (k_p is N)(k_i is Z) (1)
 Rule4: If (e is Z) and (ec is N) then (k_p is N)(k_i is P) (1)
 Rule5: If (e is Z) and (ec is Z) then (k_p is P)(k_i is P) (1)
 Rule6: If (e is Z) and (ec is P) then (k_p is P)(k_i is P) (1)
 Rule7: If (e is P) and (ec is N) then (k_p is P)(k_i is Z) (1)
 Rule8: If (e is P) and (ec is Z) then (k_p is P)(k_i is Z) (1)
 Rule9: If (e is P) and (ec is P) then (k_p is P)(k_i is Z) (1)

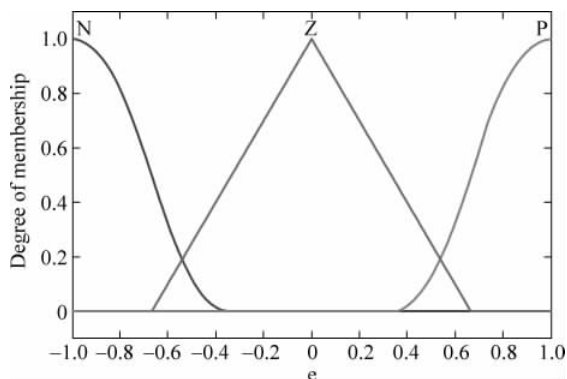


图 3.17 误差的隶属函数

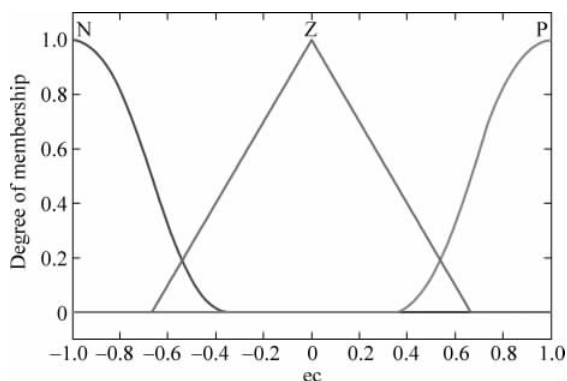
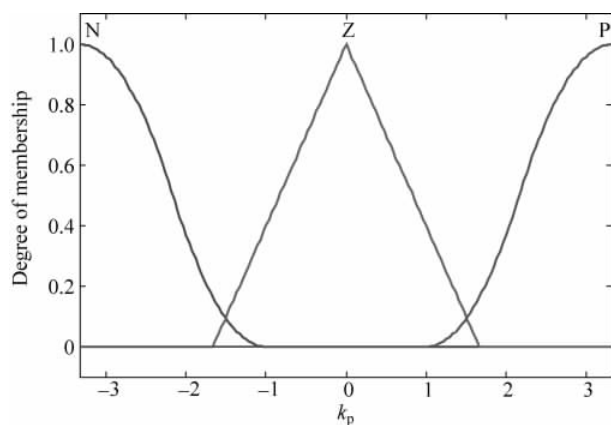
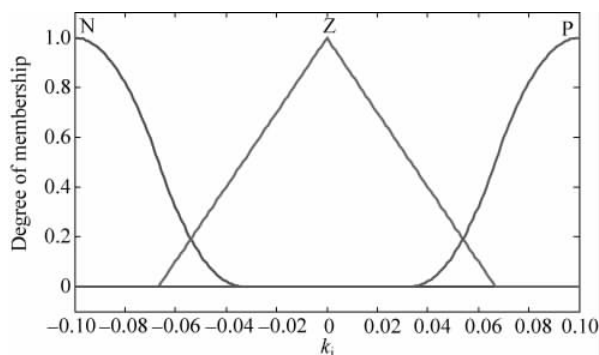


图 3.18 误差变化率的隶属函数

图 3.19 k_p 的隶属函数图 3.20 k_i 的隶属函数

另外,针对模糊推理系统 fuzzpid. fis,运行命令 fuzzy 可进行规则库和隶属函数的编辑,如图 3.21 所示,运行命令 ruleview 可实现模糊系统的动态仿真,如图 3.22 所示。

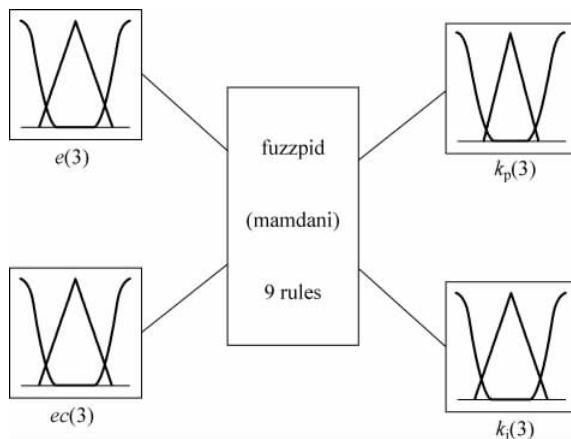


图 3.21 模糊系统 fuzzpid. fis 的结构

在程序 chap3_7.m 中,利用所设计的模糊系统 fuzzpid. fis 进行 PI 控制参数的整定。为了显示模糊规则调整效果,取 k_p 、 k_i 的初始值为 0,响应结果及 PI 控制参数的自适应变化如图 3.23~图 3.24 所示。

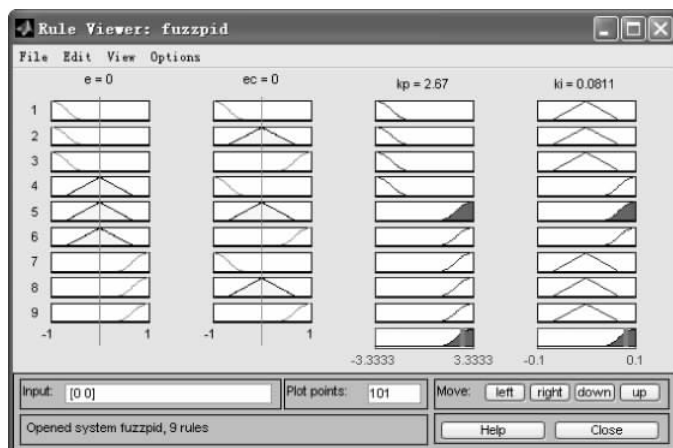


图 3.22 模糊推理系统的动态仿真环境

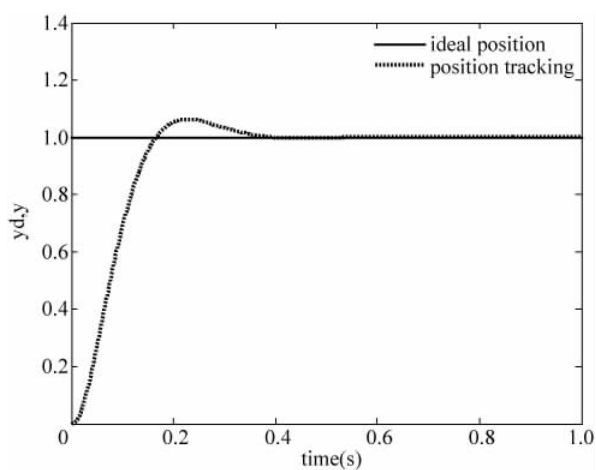
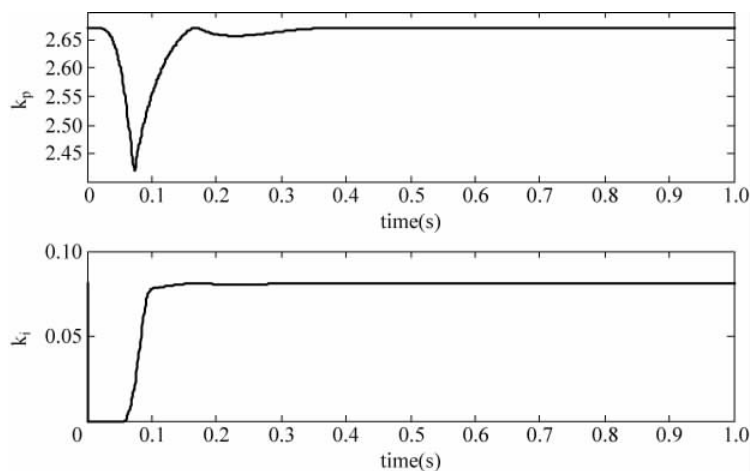


图 3.23 模糊 PI 控制阶跃响应

图 3.24 k_p 和 k_d 的模糊自适应调整

仿真程序:

(1) 模糊系统设计程序: chap3_6.m

```
% Fuzzy Tunning PI Control
clear all;
close all;
a = newfis('fuzzpid');

a = addvar(a, 'input', 'e', [-1, 1]); % Parameter e
a = addmf(a, 'input', 1, 'N', 'zmf', [-1, -1/3]);
a = addmf(a, 'input', 1, 'Z', 'trimf', [-2/3, 0, 2/3]);
a = addmf(a, 'input', 1, 'P', 'smf', [1/3, 1]);

a = addvar(a, 'input', 'ec', [-1, 1]); % Parameter ec
a = addmf(a, 'input', 2, 'N', 'zmf', [-1, -1/3]);
a = addmf(a, 'input', 2, 'Z', 'trimf', [-2/3, 0, 2/3]);
a = addmf(a, 'input', 2, 'P', 'smf', [1/3, 1]);

a = addvar(a, 'output', 'kp', 1/3 * [-10, 10]); % Parameter kp
a = addmf(a, 'output', 1, 'N', 'zmf', 1/3 * [-10, -3]);
a = addmf(a, 'output', 1, 'Z', 'trimf', 1/3 * [-5, 0, 5]);
a = addmf(a, 'output', 1, 'P', 'smf', 1/3 * [3, 10]);

a = addvar(a, 'output', 'ki', 1/30 * [-3, 3]); % Parameter ki
a = addmf(a, 'output', 2, 'N', 'zmf', 1/30 * [-3, -1]);
a = addmf(a, 'output', 2, 'Z', 'trimf', 1/30 * [-2, 0, 2]);
a = addmf(a, 'output', 2, 'P', 'smf', 1/30 * [1, 3]);

rulelist = [1 1 1 2 1 1;
            1 2 1 2 1 1;
            1 3 1 2 1 1;

            2 1 1 3 1 1;
            2 2 3 3 1 1;
            2 3 3 3 1 1;

            3 1 3 2 1 1;
            3 2 3 2 1 1;
            3 3 3 2 1 1];
a = addrule(a, rulelist);
a = setfis(a, 'DefuzzMethod', 'centroid');
writefis(a, 'fuzzpid');

a = readfis('fuzzpid');
figure(1);
plotmf(a, 'input', 1);
figure(2);
plotmf(a, 'input', 2);
figure(3);
plotmf(a, 'output', 1);
```

```

figure(4);
plotmf(a, 'output', 2);
figure(5);
plotfis(a);

fuzzy fuzzpid;
showrule(a)
ruleview fuzzpid;

(2) 模糊控制程序: chap3_7.m

% Fuzzy PI Control
close all;
clear all;

warning off;
a = readfis('fuzzpid'); % Load fuzzpid.fis

ts = 0.001;
sys = tf(133, [1, 25, 0]);
dsys = c2d(sys, ts, 'z');
[num, den] = tfdata(dsys, 'v');

u_1 = 0; u_2 = 0;
y_1 = 0; y_2 = 0;
e_1 = 0; ec_1 = 0; ei = 0;

kp0 = 0; ki0 = 0;
for k = 1:1:1000
    time(k) = k * ts;

    yd(k) = 1;
    % Using fuzzy inference to tuning PI
    k_pid = evalfis([e_1, ec_1], a);
    kp(k) = kp0 + k_pid(1);
    ki(k) = ki0 + k_pid(2);
    u(k) = kp(k) * e_1 + ki(k) * ei;

    y(k) = -den(2) * y_1 - den(3) * y_2 + num(2) * u_1 + num(3) * u_2;
    e(k) = yd(k) - y(k);
    %%%%%%%%% Return of parameters %%%%%%%%%
    u_2 = u_1; u_1 = u(k);
    y_2 = y_1; y_1 = y(k);

    ei = ei + e(k) * ts; % Calculating I

    ec(k) = e(k) - e_1;
    e_1 = e(k);
    ec_1 = ec(k);
end
figure(1);

```

```

plot(time,yd,'r',time,y,'k:', 'linewidth',2);
xlabel('time(s)');ylabel('yd,y');
legend('ideal position','position tracking');
figure(2);
subplot(211);
plot(time,kp,'r','linewidth',2);
xlabel('time(s)');ylabel('kp');
subplot(212);
plot(time,ki,'r','linewidth',2);
xlabel('time(s)');ylabel('ki');
figure(3);
plot(time,u,'r','linewidth',2);
xlabel('time(s)');ylabel('Control input');

```

3.6 大时变扰动下切换增益模糊调节的滑模控制

在滑模控制中,针对较大的扰动,为了保证闭环系统稳定,需要较大的切换增益,这就造成抖振,抖振是滑模控制中难以避免的问题。

针对跟踪问题,设计滑模函数为 $s(t) = ce(t) + \dot{e}(t)$, 其中, $e(t)$ 和 $\dot{e}(t)$ 分别为跟踪误差及其变化率, $c > 0$ 。可见,当 $s(t) = 0$ 时, $ce(t) + \dot{e}(t) = 0$, $\dot{e}(t) = -ce(t)$, 即 $\frac{1}{e(t)} \dot{e}(t) = -c$, 积分得 $\int_0^t \frac{1}{e(t)} \dot{e}(t) dt = \int_0^t -c dt$, 则 $\ln e(t) \big|_0^t = -ct$, 从而得到

$$\ln \frac{e(t)}{e(0)} = -ct$$

收敛结果为

$$e(t) = e(0) \exp(-ct)$$

即当 $t \rightarrow \infty$ 时, 误差指数收敛于 0, 收敛速度取决于 c 值。如果通过控制律的设计, 保证 $s(t)$ 也是指数收敛于 0, 则当 $t \rightarrow \infty$ 时, 误差变化率也是指数收敛于 0。

模糊逻辑的设计不依靠被控对象的模型, 其突出优点是能够将人的控制经验通过模糊规则融入到控制器中, 通过设计模糊规则, 实现高水平的控制器。采用模糊规则, 可根据滑模到达条件对切换增益进行有效地估计, 并利用切换增益消除干扰项, 从而消除抖振。

3.6.1 系统描述

考虑如下模型

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u(t) + d(t) \end{aligned} \quad (3.12)$$

其中, $d(t)$ 为时变干扰。

3.6.2 滑模控制器设计

取滑模函数为

$$s = \dot{e} + ce, \quad c > 0 \quad (3.13)$$

其中, e 为跟踪误差, $e = x_{1d} - x_1$, x_1 为位置信号, x_{1d} 为指令角度。

取 Lyapunov 函数为

$$V = \frac{1}{2}s^2$$

设计滑模控制器为

$$u = \ddot{x}_{1d} + c\dot{e} + K(t)\operatorname{sgn}s \quad (3.14)$$

取

$$K(t) = \max |d(t)| + \eta \quad (3.15)$$

其中, $\eta > 0$ 。则

$$\dot{V} = s\dot{s} = s(\ddot{e} + c\dot{e}) = s(\ddot{x}_{1d} - \ddot{x}_1 + c\dot{e}) = s(\ddot{x}_{1d} - u - d(t) + c\dot{e})$$

将控制律代入, 得

$$s\dot{s} = s(-K(t)\operatorname{sgn}(s) - d(t)) = -K(t)|s| - d(t)s \leqslant -\eta|s| \quad (3.16)$$

在滑模控制律式(3.14)中, 切换增益 $K(t)$ 值是造成抖阵的原因。 $K(t)$ 用于补偿不确定项 $d(t)$, 以保证滑模存在性条件得到满足。如果 $d(t)$ 时变, 则为了降低抖阵, $K(t)$ 也应该时变。可采用模糊规则, 根据经验实现 $K(t)$ 的变化。

3.6.3 模糊规则设计

滑模存在条件为 $s\dot{s} < 0$, 当系统到达滑模面后, 将会保持在滑模面上。由式(3.16)可见, $K(t)$ 为保证系统运动得以到达滑模面的增益, 其值必须足以消除不确定项的影响, 才能保证滑模存在条件 $s\dot{s} < 0$ 成立。

模糊规则设计的原则为: 在满足 $s\dot{s} < 0$ 条件下, 尽量采用较小的增益 $K(t)$, 以降低抖阵。设计模糊规则如下

如果 $s\dot{s} > 0$, 则 $K(t)$ 应增大

如果 $s\dot{s} < 0$, 则 $K(t)$ 应减小

由式(3.16)可设计关于 $s\dot{s}$ 和 $\Delta K(t)$ 之间关系的模糊系统, 在该系统中, $s\dot{s}$ 为输入, $\Delta K(t)$ 为输出。系统输入输出的模糊集分别定义如下

$$s\dot{s} = \{\text{NB} \quad \text{NM} \quad \text{ZO} \quad \text{PM} \quad \text{PB}\}$$

$$\Delta K = \{\text{NB} \quad \text{NM} \quad \text{ZO} \quad \text{PM} \quad \text{PB}\}$$

其中, NB 为负大, NM 为负中, ZO 为零, PM 为正中, PB 为正大。

模糊系统的输入输出隶属函数如图 3.25 和图 3.26 所示。

模糊规则设计如下

Rule1: If $s\dot{s}$ is PB then ΔK is PB

Rule2: If $s\dot{s}$ is PM then ΔK is PM

Rule3: If $s\dot{s}$ is ZO then ΔK is ZO

Rule4: If $s\dot{s}$ is NM then ΔK is NM

Rule5: If $s\dot{s}$ is NB then ΔK is NB

采用积分的方法对 $\hat{K}(t)$ 的上界进行调节

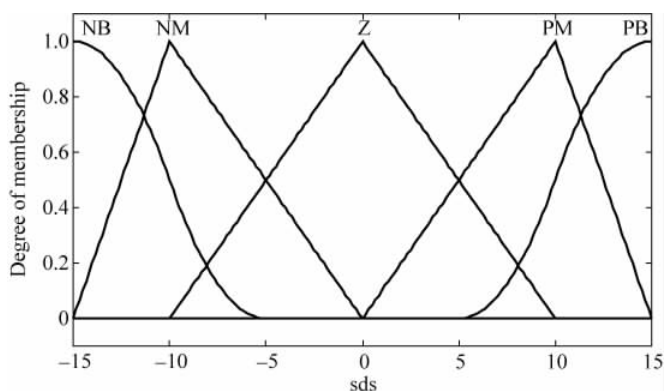


图 3.25 模糊输入的隶属函数

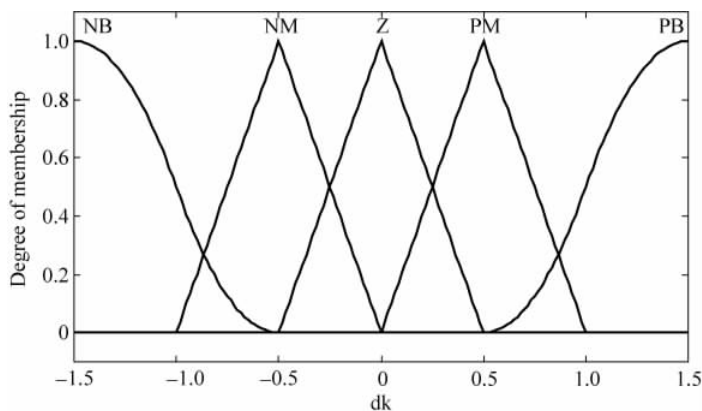


图 3.26 模糊输出的隶属函数

$$\hat{K}(t) = G \int_0^t \Delta K dt \quad (3.17)$$

其中, $G > 0$ 为比例系数, 根据经验确定。

用 $\hat{K}(t)$ 代替式(3.14)的 $K(t)$, 则控制律变为

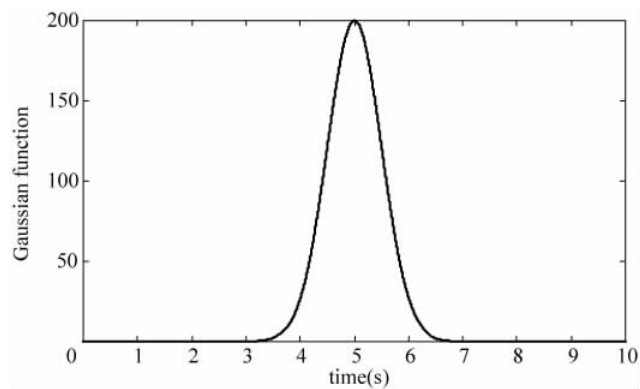
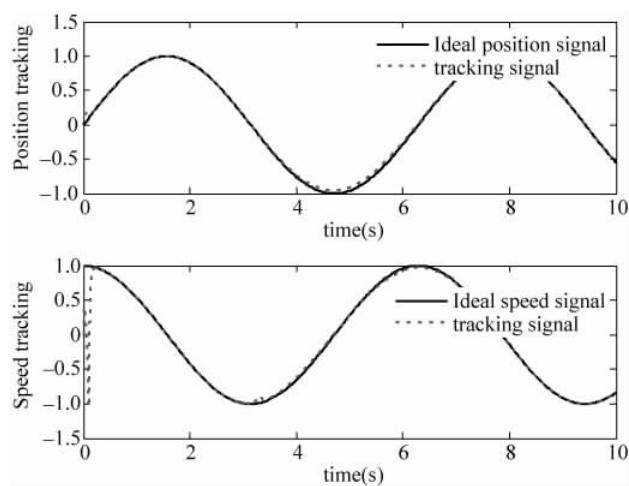
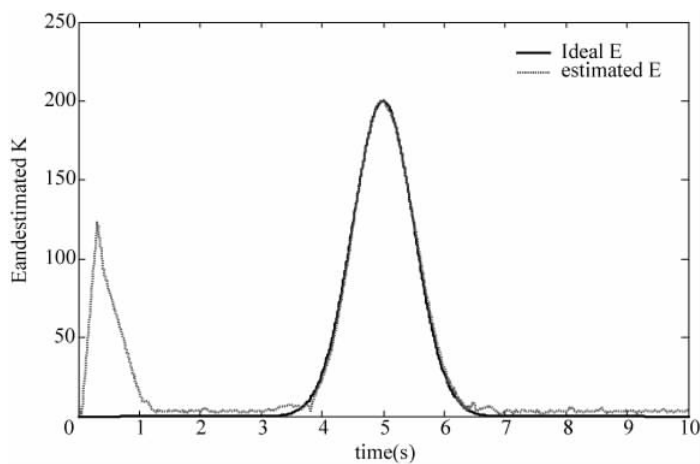
$$u = \ddot{x}_{1d} + c\dot{e} + \hat{K}(t)\text{sgns} \quad (3.18)$$

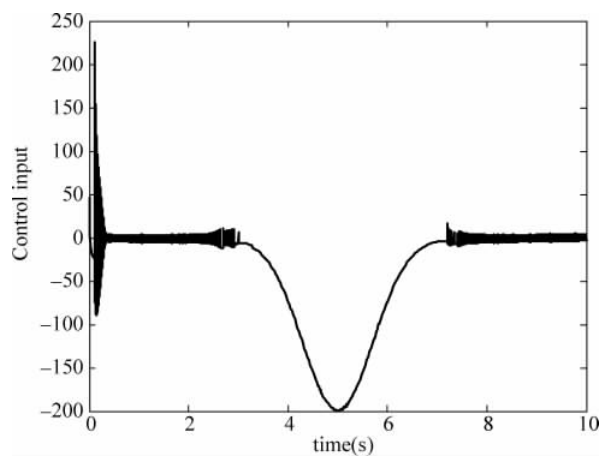
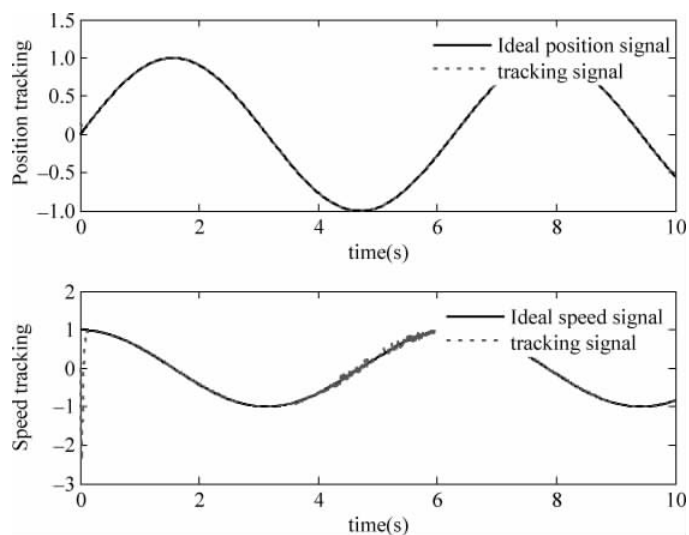
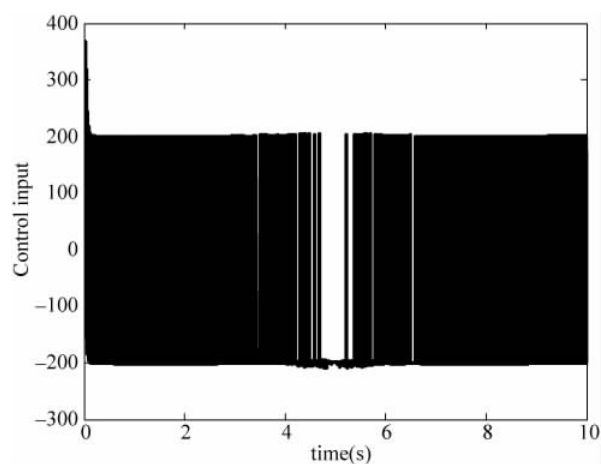
3.6.4 仿真实例

被控对象为式(3.12), 采用高斯函数的形式表示 $d(t)$, $d(t) = 200\exp(-t^2)$, 如图 3.27 所示。

位置指令信号为 $x_{1d} = \sin t$ 。首先运行程序 chap3_8fuzz.m 建立模糊系统, 模糊规则库保存在 chap3_8fis.fis 中, 并得到模糊系统输入输出的隶属函数图, 如图 3.25 和图 3.26 所示。

在控制律中, 取 $D=200$, $c=50$, $\eta=3.0$ 。取 $M=2$, 采用模糊调节增益的控制律式(3.18), $G=400$, 仿真结果如图 3.28~图 3.30 所示。取 $M=1$, 采用传统的控制律式(3.14), 仿真结果如图 3.31 和图 3.32 所示。可见, 采用基于模糊规则的模糊滑模控制方法, 可有效地通过切换增益消除干扰项, 从而消除抖振。

图 3.27 高斯函数形式的干扰 $d(t)$ 图 3.28 模糊调节增益时的位置和速度跟踪 ($M=2$)图 3.29 扰动 $d(t)$ 和增益 $\hat{K}(t)$ 的调节 ($M=2$)

图 3.30 模糊调节增益时的控制输入($M=2$)图 3.31 采用传统控制器时的位置和速度跟踪($M=1$)图 3.32 传统控制时的控制输入($M=1$)

仿真程序：

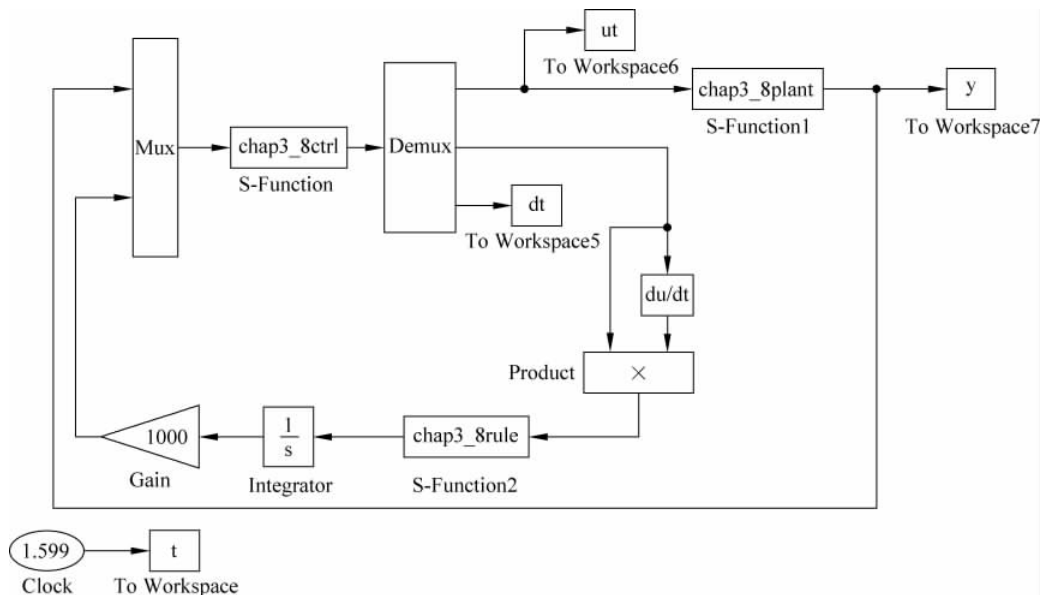
(1) 高斯函数：chap3_8func.m

```
clear all;
close all;

ci = 5;
ts = 0.001;
for k = 1:1:10000
    t(k) = k * ts;
    d(k) = 200 * exp( - (t(k) - ci)^2 );
end
figure(1);
plot(t,d,'k','linewidth',2);
```

(2) 模糊系统设计程序：chap3_8fuzz.m

(3) Simulink 主程序：chap3_8sim.mdl



(4) 控制器 S 函数：chap3_8ctrl.m

(5) 被控对象 S 函数：chap3_8plant.m

(6) 模糊系统 S 函数：chap3_8rule.m

(7) 作图程序：chap3_8plot.m

思考题

1. 简述模糊控制器由哪几部分组成？各完成什么功能？
2. 简述模糊控制器设计的步骤。
3. 已知某一炉温控制系统，要求温度保持在 600°C 恒定。针对该控制系统有以下控制经验：

- (1) 若炉温低于 600°C , 则升压, 低得越多升压越高。
- (2) 若炉温高于 600°C , 则降压, 高得越多降压越低。
- (3) 若炉温等于 600°C , 则保持电压不变。

设模糊控制器为一维控制器, 输入语言变量为误差, 输出为控制电压。输入输出变量的量化等级为七级, 取 5 个语言值, 隶属度函数根据确定的原则任意确定。试设计误差变化划分表、控制电压变化划分表和模糊控制规则表。

4. 已知被控对象为 $G(s) = \frac{1}{10s+1}e^{-0.5s}$ 。假设系统给定为阶跃值 $r=30$, 采样时间为 $0.5s$, 系统的初始值 $r(0)=0$ 。试分别设计常规的 PID 控制器、常规的模糊控制器和模糊 PID 控制器, 分别对 3 种控制器进行 MATLAB 仿真, 并比较控制效果。

5. 在 3.6 节中, 控制输入抖振仍然存在, 如图 3.30 所示, 如何通过设计模糊规则, 进一步降低控制输入抖振。