

3.2 模糊控制器的结构和设计

3.2.7 模糊控制器的软件实现（Matlab）

1 模糊控制查询表的实现

- 初始化
- 总结模糊关系
- 总结模糊查询表

3.2 模糊控制器的结构和设计

1) 初始化

假设**E**、**EC**和**U**的论域: $\{-6,-5,\dots,-1,0,1,\dots,5,6\}$;

E、**EC**和**U**定义了7个语言值{**NB**, **NM**, **NS**, **Z**, **PS**, **PM**, **PB**};

则我们在**Matlab**中通过定义三个向量来表示这些语言值:

```
Input1_Terms=[1,2,3,4,5,6,7];  
Input2_Terms=[1,2,3,4,5,6,7];  
output_Terms=[1,2,3,4,5,6,7];
```

3.2 模糊控制器的结构和设计

各语言值的隶属函数采用三角函数，其分布可用下表表示：

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
NB	1	0.5	0	0	0	0	0	0	0	0	0	0	0
NM	0	0.5	1	0.5	0	0	0	0	0	0	0	0	0
NS	0	0	0	0.5	1	0.5	0	0	0	0	0	0	0
Z	0	0	0	0	0	0.5	1	0.5	0	0	0	0	0
PS	0	0	0	0	0	0	0	0.5	1	0.5	0	0	0
PM	0	0	0	0	0	0	0	0	0	0.5	1	0.5	0
PB	0	0	0	0	0	0	0	0	0	0	0	0.5	1

3.2 模糊控制器的结构和设计

在matlab中，可以用一个矩阵来表示该表。

```
Input1_Terms_Membership= [1,0.5,0,0,0,0,0,0,0,0,0,0,0;  
                           0,0.5,1,0.5,0,0,0,0,0,0,0,0,0;  
                           0,0,0,0.5,1,0.5,0,0,0,0,0,0,0;  
                           0,0,0,0,0,0.5,1,0.5,0,0,0,0,0;  
                           0,0,0,0,0,0,0,0.5,1,0.5,0,0,0;  
                           0,0,0,0,0,0,0,0,0,0.5,1,0.5,0;  
                           0,0,0,0,0,0,0,0,0,0,0,0.5,1];
```

```
Input2_Terms_Membership=Input1_Terms_Membership;  
Output_Terms_Membership=Input1_Terms_Membership;
```

3.2 模糊控制器的结构和设计

假设控制规则表总结如下

U		EC						
		NB	NM	NS	Z	PS	PM	PB
E	NB	NB	NB	NB	NB	NM	Z	Z
	NM	NB	NB	NB	NB	NM	Z	Z
	NS	NM	NM	NM	NM	Z	PS	PS
	Z	NM	NM	NS	Z	PS	PM	PM
	PS	NS	NS	Z	PM	PM	PM	PM
	PM	Z	Z	PM	PB	PB	PB	PB
	PB	Z	Z	PM	PB	PB	PB	PB

3.2 模糊控制器的结构和设计

将语言值按顺序编号，NB、NM、NS、Z、PS、PM、PB分别对应1、2、3、4、5、6、7号。

则上表可用一个矩阵表示为：

```
Rule= [1,1,1,1,2,4,4;  
       1,1,1,1,2,4,4;  
       2,2,2,2,4,5,5;  
       2,2,3,4,5,6,6;  
       3,3,4,6,6,6,6;  
       4,4,6,7,7,7,7;  
       4,4,6,7,7,7,7];
```

3.2 模糊控制器的结构和设计

2)总结模糊关系

某条规则蕴涵的模糊关系

```
Output_Terms_Index=Rule(Input1_Terms_Index,Input2_Terms_Index );
```

注: *Input1_Terms_Index* 代表输入E的语言值的序号, *Input2_Terms_Index* 代表输入EC语言值的序号, *Output_Terms_Index* 代表输出U语言值的序号

```
A=Input1_Terms_Membership(Input1_Terms_Index,:);  
B=Input2_Terms_Membership(Input2_Terms_Index,:);  
C=Output_Terms_Membership(Output_Terms_Index,:);
```

注: *A*代表输入E的模糊值, *B*代表输入EC的模糊值, *C*代表输出U的模糊值

程序
段
1

3.2 模糊控制器的结构和设计

程序段
2

```
for i=1:13
    for j=1:13
        R1(i,j)=min(A(i),B(j));
    end
end
```

注: $R1 = A \times B$

```
R2=[ ];
for k=1:13
    R2=[R2;R1(k,:)]';
end
```

注: $R2 = R1^T$

```
for i=1:169
    for j=1:13
        R3(i,j)=min(R2(i),C(j));
    end
end
```

注: $R3 = R2 \times C, 169 \times 13$

3.2 模糊控制器的结构和设计

所有规则蕴涵的模糊关系:

```
for i=1:169
    for j=1:13
        R(i,j)=0;
    end
end

for Input1_Terms_Index=1:7
    for Input2_Terms_Index=1:7
        程序段1;
        程序段2;
        R=max(R,R3);
    end
end
```

注: 初始化R

注: R为所有规则模糊关系的并集,
共49条

$$\underset{\sim}{R} = (\underset{\sim}{A_1} \times \underset{\sim}{B_1}) \cup (\underset{\sim}{A_2} \times \underset{\sim}{B_2}) \cup \dots \cup (\underset{\sim}{A_n} \times \underset{\sim}{B_n}) = \bigcup_{i=1}^n (\underset{\sim}{A_i} \times \underset{\sim}{B_i})$$

3.2 模糊控制器的结构和设计

3)总结模糊查询表

a)模糊化

```
Input1_value_membership=Input1_Terms_Membership(:,Input1_value_index);
```

注: *Input1_value_index*表示输入*E*的精确值的序号,
*Input1_value_membership*为*E*的精确值属于其各个模糊值的隶属度组成的向量

```
[Max_Input1_value,Max_Input1_index]=max(Input1_value_membership);
```

注: *Max_Input1_index*表示隶属度最大的模糊值的序号, *Max_Input1_value*为与之对应的隶属度

```
Ad=Input1_Terms_Membership(Max_Input1_index,:);
```

注: *Ad*为与输入*E*的精确值相对应的隶属度最大的模糊值, 也就是模糊化后的*E*的模糊值。

3.2 模糊控制器的结构和设计

同理可以得到输入EC模糊化后的模糊值Bd

```
Input2_value_membership=Input2_Terms_Membership(:,Input2_value_index);  
[Max_Input2_value,Max_Input2_index]=max(Input2_value_membership);  
Bd=Input2_Terms_Membership(Max_Input2_index,:);
```

b) 推理

```
for i=1:13  
    for j=1:13  
        Rd1(i,j)=min(Ad(i),Bd(j));  
    end  
end
```

$$Rd1 = Ad \times Bd$$

3.2 模糊控制器的结构和设计

```
Rd2=[ ];  
for k=1:13  
    Rd2=[Rd2,Rd1(k,:)];  
end
```

$$Rd2 = Rd1^T$$

```
for j=1:13  
    Cd(j)=max(min(Rd2',R(:,j)));  
end
```

$Cd = Rd2 \circ R$, Cd 为推理后得到的模糊输出

C) 去模糊化

```
sum1=0;  
sum2=0;  
for i=1:13  
    sum1=sum1+Cd(i);  
    sum2=sum2+Cd(i)*Output(i);  
end  
OUT=round(sum2/sum1);
```

加权平均法

3.2 模糊控制器的结构和设计

对于每种可能的E、EC的精确取值进行a)b)c)的运算可以得到模糊查询表:

```
for Input1_value_index=1:13
    for Input2_value_index=1:13
        模糊化;
        推理;
        去模糊化;
        Fuzzy_Table(Input1_value_index,Input2_value_index)=OUT;
    end
end
```

整个程序清单

3.2 模糊控制器的结构和设计

运行结果:

Fuzzy_Table =

-6	-6	-6	-6	-6	-6	-5	-5	-3	-3	-2	-2	0
-6	-6	-6	-6	-6	-6	-5	-5	-3	-3	-2	-2	0
-5	-5	-5	-5	-5	-5	-3	-3	-2	-2	-1	-1	1
-5	-5	-5	-5	-5	-5	-3	-3	-2	-2	-1	-1	1
-4	-4	-4	-4	-3	-3	-2	-2	0	0	0	0	2
-4	-4	-4	-4	-3	-3	-2	-2	0	0	0	0	2
-3	-3	-2	-2	0	0	0	0	0	0	2	2	3
-3	-3	-2	-2	0	0	0	0	0	0	2	2	3
-2	-2	0	0	0	0	2	2	3	3	4	4	4
-2	-2	0	0	0	0	2	2	3	3	4	4	4
-1	-1	1	1	2	2	3	3	5	5	5	5	5
-1	-1	1	1	2	2	3	3	5	5	5	5	5
0	0	2	2	3	3	5	5	6	6	6	6	6

3.2 模糊控制器的结构和设计

2 模糊控制在线运行代码

```
ek=refk-yk;           %计算第k个采样周期的误差和误差变化率
eck=(ek-ek_1)/t;
E=round(ke*(ek-(eh+el)/2)); %将E的论域转换到模糊控制器的论域
if E>6
    E=6;
elseif E<-6
    E=-6;
end
EC=round(kec*(eck-(ech+ecl)/2)); %将EC的论域转换到模糊控制器的论域
if EC>6
    EC=6;
elseif EC<-6
    EC=-6;
end
U=Fuzzy_Table (E+7,EC+7); %查模糊控制查询表得到输出值U
u=Ku*U+(uh+ul)/2; %将输出转换到实际论域
```