洗衣机模糊推理系统的实现

1. **研究背景和目的**

为了更好的模仿人工洗涤过程，提高洗衣机的洗涤质量，针对以单片机为核心的洗衣机控制系统，引入了模糊推理技术。

随着计算机及其相关技术的发展，模糊控制由最初的经典模糊控制发展到自适应模糊控制，专家模糊控制和基于神经网络的自学习模糊控制。模糊控制作为智能领域中最具有世纪意义的一种控制方法，已经在家用电器和自动化领域等其他区很多的行业中解决了传统控制方法无法或者难以解决的问题。

早期人们洗衣时使用洗衣板，进而使用定时式洗衣机，是机械半电子式，后来是全自动洗衣机，现在，人们希望是智能洗衣机。即把衣物放置洗衣机中，启动电源，洗衣就能自动判别衣质、衣量、污浊度并自动确定洗涤时间，最后能把衣物洗干净，

模糊控制洗衣机是一种智能型的、真正全自动的洗衣机。它能自动识别衣质、衣量及肮脏程度，自动决定水量，自动投入适量的洗涤剂，从而全部自动地完成整个洗涤过程。洗涤程序是通过模糊推理来决定的，因此有极高的洗涤效能，不仅提高了洗衣机的全自动化程度，而且大大提高了洗衣质量。

1. **系统总体设计**

**2.1洗衣机控制器软件总体设计思想**

在完善的硬件装置的基础上，洗衣机控制器系统的自己要公平能均通过软件完成。洗衣机控制器系统的软件设计是整个洗衣机控制器的核心，它关系到洗衣机控制器系统的性能及其功能的实现。在洗衣机控制器的软件设计中，主要的设计思想有三个方面：一是满足控制的实时性要求；二是软件要有很强的灵活性；

首先阐述了模糊洗衣机的基本原理，完成了系统硬件设计；在软件方面，对检测到的水温及肮脏度、衣质、衣量等物理量进行模糊化处理，建立了洗涤规则库，通过模糊推理，实现对洗涤时间、水流强弱、洗涤剂用量以及脱水时间的自动控制，提高了洗衣机的自动控制和智能化程度。

控制系统的基本原理

模糊洗衣机通过对衣质、衣量、衣物肮脏度及初始水位高低情况等输入量，来决定洗涤时间、水流强弱、洗涤剂用量以及脱水时间等。因此，系统的被控对象主要是洗涤时间、水流强弱、洗涤剂用量及脱水时间等，即输出量。洗衣机中被控对象的运动规律极其复杂，很难用数学语言的形式来表达。而模糊控制无需建立被控对象的数学模型，因此适用于洗衣机控制。模糊型洗衣机无需人为按键设置水位、水流方式、洗涤时间、清洗时间和脱水时间，用户只需放进待洗衣物，按下启动键，洗衣机便能自动完成洗涤过程，实现了真正的“全自动”。  
    模糊洗衣机要求控制系统能根据测得的衣质、衣量情况，确定适当水位、水流强度、洗涤时间、洗涤剂用量，并根据肮脏度的变化合理地修正洗涤时间，以达到节水和节能的目的。

1. 模糊洗衣机的模糊推理

在模糊洗衣机中，浑浊度，布质，布量等常见的量都是通过对现行状态的检测再通过模糊推理得出的。在模糊推理中，需要推理的前件和后件，即输入条件和输出结果，在模糊洗衣机中，主要是考虑布质、布量、水温和肮脏程度这几项条件，而从这几项条件求取水位、洗涤时间和水流、漂洗和脱水时间等。模糊洗衣机的推理如图：

水位

模糊推理

洗涤时间

布质

水流

布量

漂洗方式

水温

肮脏度

脱水时间

在模糊洗衣机中，水质和布量是无法通过物理传感器测出的，所以，它们的求取都是采用间接的方法。布质、布量和洗涤的过程有很大的关系。从人们一般的经验来说，棉质的布料，洗涤会困难一点，如果是化纤材质，则困难会小一点，布量多一点，洗涤时间长一点，反之短一点，除了肮脏度之外，模糊推理还考虑布质和布量。

1. **详细设计**

洗农机模糊控制系统的主要工作就是依据语言规则进行模糊推理决策。因此，在进行模糊规则推理之前，先要确定好语言控制规则(即知识库)。语言控制规则是根据经验知识来确定的，并在实验过程中不断加以修正和完善。它相当于计算机程序设计语言的条件语句：“IF……THEN……”。

模糊推理系统是一个多输入多输出的控制系统。实际中，模糊推理的前件和后件之间的关系对于不同的因素也有所不同，。例如，肮脏程度和水温可以确定洗涤投放的衣量和洗涤时间，而布量、布质等可以确定水位和水流、脱水时间等。因此，在推理中八有关前件和后件进行处理，这种处理分为主要因素推理和顺序因素推理两种。考虑到洗衣机过程中的两种情况，一种为静态的，即洗涤浓度；另一种为动态的即洗衣水流及时间，因此推理分为洗涤剂浓度推理和洗衣推理两大部分。

洗涤剂浓度推理，规则如下 ：

如果浑浊度高，则洗涤剂投入量大；

如果浑浊度偏高，则洗涤剂投入量偏大；

如果浑浊度低，则洗涤剂投入量小；

洗衣推理，规则如下：  
    规则一：如果布量少，布质以化纤偏多，而且水温高，则水流为特弱，洗涤时间特短

    规则二： 如果布量多，布质以棉布偏多，而且水温低，则水流为特弱，洗涤时间特长

下图为洗衣机模糊控制表：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 布质  水温  布量 | | 棉布偏多 | | | 棉布与化纤各半 | | | 化纤偏多 | | |
| 低 | 中 | 高 | 低 | 中 | 高 | 低 | 中 | 高 |
|  | 水流 | 特强 | 强 | 强 | 强 | 强 | 中 | 中 | 中 | 中 |
| 时间 | 特长 | 长 | 中 | 长 | 长 | 中 | 长 | 中 | 中 |
|  | 水流 | 强 | 中 | 中 | 中 | 中 | 中 | 中 | 弱 | 弱 |
| 时间 | 长 | 中 | 短 | 长 | 中 | 中 | 中 | 中 | 短 |
|  | 水流 | 弱 | 弱 | 弱 | 弱 | 弱 | 弱 | 弱 | 弱 | 特弱 |
| 时间 | 中 | 中 | 短 | 中 | 短 | 短 | 中 | 短 | 特短 |

上表给出了洗衣推理的所有规则，这些规则前件有三个因素，后件有两个因素。因此它们是一种多输入多输出推理。对于输入量，即前件各个因素的模糊量定义不同。布量的模糊量为：“多”、“中”、“少”；水温的模糊量为：“高”、“中”、“低”；布质的模糊量为：“棉布偏多”、“棉布化纤各半”、“化纤偏多”。而输出量，即后件中，水流的模糊量为：“特强”、“强”、“中”、“弱”、“特弱”，时间的模糊量为：“特长”、“长”、“中”、“短”、“特短”。

上述模糊量中，各自的隶属函数都不同，水温、布量和时间的模糊量如图：

U

1

低 中 高

水温T

U

1

少 中 多

布量W

U

1

特短 短 中 长 特长

时间t

图三 水温、布量和时间的模糊量

图三中，水温的模糊量是采用梯形隶属函数的；同样，布量的模糊量也是用梯形隶属函数；；对于时间而言，其模糊量则是采用三角形隶属函数。在模糊推理中，模糊量的范围的选取有很大的作用，其形状所起的作用和范围相比就小的多。

对于主要因素推理和顺序因素推理这两种推理，它们之间是有一层隐含的推理关系的。主要因素推理是以采用人的思维中的主要因素起到决定作用原理执行的。顺序因素推理则是把前一种推理的结果作为本次推理的前件，从而推理出新的结果。在洗衣机中，如果考虑浑浊度，洗涤剂投放量、水流、洗涤时间等因素的推理。作为主要因素推理显然有：

如果浑浊度高，洗涤剂投入量大，如果布量多，布质以棉布偏多且水温高，则水流为强，洗涤时间为重；但实际上，洗涤剂投入量大时，要求洗涤时间较长才能洗干净。故还需要考虑顺序因素推理，如果洗涤剂投入量大，则洗涤时间长；如果洗涤剂投入量中，则洗涤时间中；

当顺序推理和主要因素退出的某一个后件因素的隶属度不同时，则采用大者优先的原则处理。

1. **编码实现过程**

1、设计推理系统的步骤：

确定输入/输出的模糊子集以及论域

该系统中，设计了一个衣服上的油污、污渍的参数，

污泥{SD(污泥少), MD(中等污泥), LD(污泥多)}

油脂{NG(无油脂), MG(中等油脂), LG(油脂多)}

控制对象是洗衣机的洗涤时间，论域：[0, 60] 。

输入是被洗衣物的污泥和油脂，论域：[0, 100]

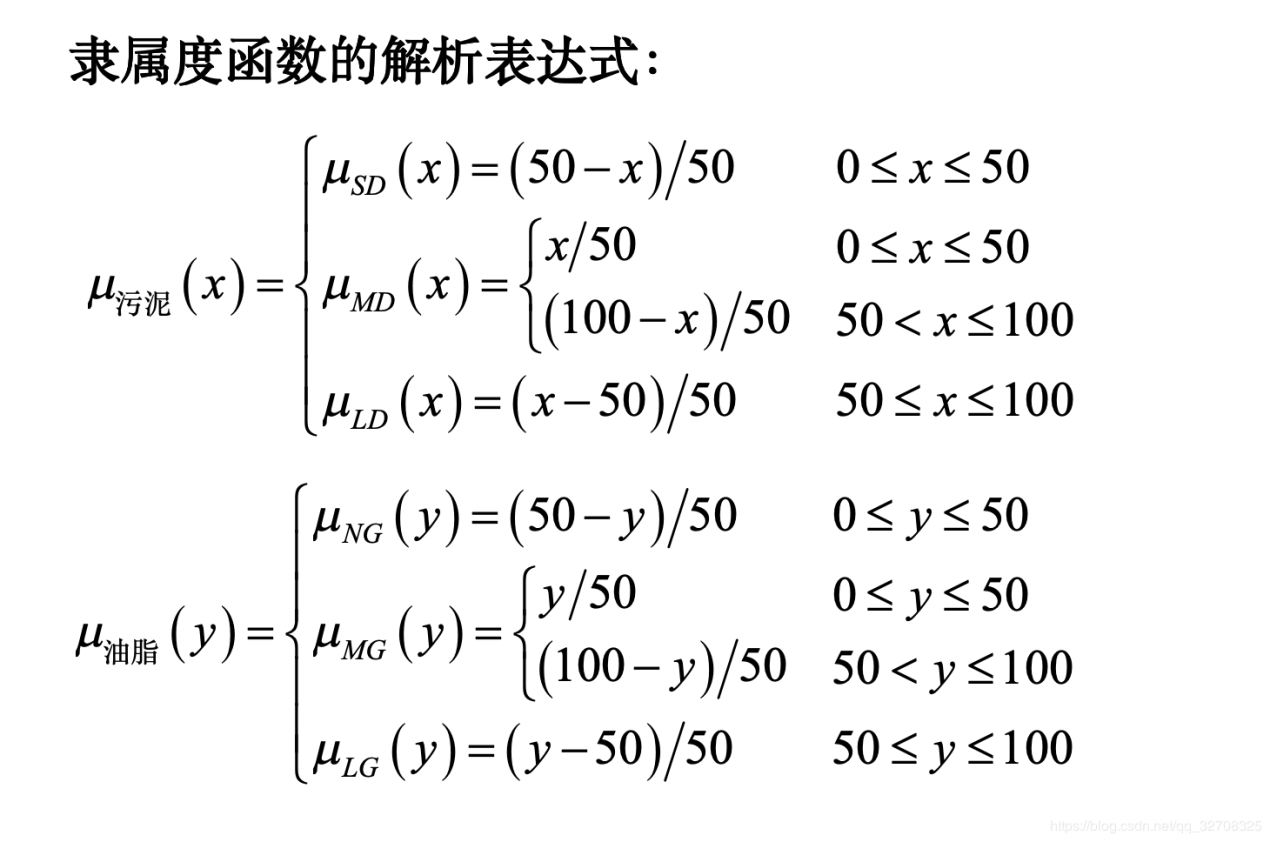
输出的是洗衣机的洗涤时间：

洗涤时间{VS(很短), S(短), M(中等), L(长), VL(很长)}

选择控制规则

|  |  |  |  |
| --- | --- | --- | --- |
| 油脂  污泥 | NG（无油脂） | MG（中等油脂） | LG（油脂多） |
| SD（污泥少） | VS | M | L |
| MD（中等污泥） | S | M | L |
| LD（污泥多） | M | L | VL |

规则的运算关系

污泥、污渍以及洗涤时间的隶属度函数的解析表达式如下：  


3、代码块实现：

#include "stdio.h"

#include "stdlib.h"

#include "math.h"

/\*\*

设计一个洗衣机的模糊推理系统

人工智能ppt2中的洗衣机控制系统

洗涤时间的论域 0-60

油污以及灰尘的论域 0-100

输入格式：油污 Oil（S：少 M：中等 L:大量） 污渍 Stain（S：少 M：中等 L:大量）

输出格式： 洗涤时间 time（VS，S，M，L，VL）

\*\*/

// 定义油污以及污渍的全局变量

#define OIL 100.0

#define Stain 100.0

struct elem

{

double oil;

double stain;

int result;

};

// 三个隶属度函数，用来计算污渍的隶属度.1代表SD，2代表MD，3代表LD

double ruleMD(double stain){

if(stain<0||stain>100) return 0.0; // 当返回的参数不在论域中时，返回错误

else {

// 当传入的参数在0-100之间时，该处有两种情况

double result;

if(stain>=0&&stain<=50)

// 计算MD的结果，并且和同参数下的SD结果相比较，得出一个结果

return stain/50.0;

else {

// 同上的操作，得出结果和同参数下的LD相比较

return (100-stain)/50;

}

}

}

double ruleSD(double stain){

// SD部分的rule

// 当输入的参数0<=x<=50,执行该方法

double result = (50-stain)/50.0;

double returnMDresult = ruleMD(stain);

// 传参数到MD中，计算，并比较

// 1、相同，则返回结果为SD，2、SD的结果大，则返回SD，3、MD的结果大，则返回MD的返回值

if(result<returnMDresult)

return 2.0;

else return 1.0;

}

double ruleLD(double stain ){

// LD部分的rula

// 当输入的参数在50-100之间时，执行

// 同时将参数传入给MD，同时比较MD方法传回来的参数和该方法求出的值相比较，求出最后的最适合的预测值

double returnMDresult = ruleMD(stain);

double result = (stain - 50) / 50.0;

// 比较后，得到预测值

if(result<returnMDresult)

return 2.0;

else return 3.0;

}

double ruleMG(double oil){

// 当传入的参数在0-100之间时，该处有两种情况

if(oil<0||oil>100) return 0; // 当在论域之外时，直接返回无结果

else {

double result;

if(oil>=0&&oil<=50)

// 计算MD的结果，并且和同参数下的SD结果相比较，得出一个结果

return oil/50.0;

else {

// 同上的操作，得出结果和同参数下的LD相比较

return (100-oil)/50;

}

}

}

// 三个隶属度函数，用来计算油污的隶属度，1代表SG，2代表MG，3代表LG

double ruleSG(double oil){

if(oil<0||oil>50)return 0.0;

else {

// SG部分的rule

// 当输入的参数0<=x<=50,执行该方法

double result = (50-oil)/50.0;

double returnMGresult = ruleMG(oil);

// 传参数到MD中，计算，并比较

// 1、相同，则返回结果为SD，2、SD的结果大，则返回SD，3、MD的结果大，则返回MD的返回值

if(result<returnMGresult)

return 2.0;

else return 1.0;

}

}

double ruleLG(double oil){

// LD部分的rula

// 当输入的参数在50-100之间时，执行

// 同时将参数传入给MG，同时比较MG方法传回来的参数和该方法求出的值相比较，求出最后的最适合的预测值

double returnMGresult = ruleMG(oil);

double result = (oil - 50) / 50.0;

// 比较后，得到预测值

if(result<returnMGresult)

return 2.0;

else return 3.0;

}

// F函数，总的函数，从该函数中分流到rule的三个函数中

int Function(double oil,double stain){

/\*\* VS: SD,SG

S: MD,SG

M: SD,MG MD,MG LD,SG

L: SD,LG MD，LG LD，MG

XL: LD,LG

\*\*/

// 根据规则输出最后的洗涤时间

double result\_D,result\_G;

// 需要客户的正确输入

if(oil<0||oil>OIL||stain<0||stain>Stain) return 0.0;

else {

// 根据参数的大小，分别传送给各个rula

if(oil>=0&&oil<=50) result\_G = ruleSG(oil);

else result\_G = ruleLG(oil);

if(stain>=0&&stain<=50) result\_D = ruleSD(stain);

else result\_D = ruleLD(stain);

// 比较最后的结果

if(result\_D==1.0&&result\_G==1.0) return 1; // return VS

else if(result\_G==1.0&&result\_D==2.0) return 2; // return S

else if((result\_D==1.0&&result\_G==2.0)||(result\_G==2.0&&result\_D==2.0)||(result\_G==1.0&&result\_D==3.0)) return 3; // return M

else if((result\_D==1.0&&result\_G == 3.0)||(result\_D == 2.0&&result\_G==3.0)||(result\_D==3.0&&result\_G==2.0)) return 4;

else if(result\_G==3.0&&result\_D==3.0) return 5;

}

}

int main(int argc, char const \*argv[])

{

/\* code \*/

elem element[10]={{20.0,20.0,0},{20.0,30.0,0},{25.0,25.0,0},{30.0,30.0,0},{30.0,20.0,0},{65.0,77.0,0},{20.0,80.0,0},{80.0,80.0,0},{49.0,30.0,0},{50.0,50.0,0}};

// 输入参数oil，stain

for(int i=0;i<10;i++){

element[i].result = Function(element[i].oil,element[i].stain);

printf("油污：%.1lf 污渍：%.1lf 要进行的洗涤时间为：", element[i].oil,element[i].stain);

switch(element[i].result){

case 1:printf("VS\n");break;

case 2:printf("S\n");break;

case 3:printf("M\n");break;

case 4:printf("L\n");break;

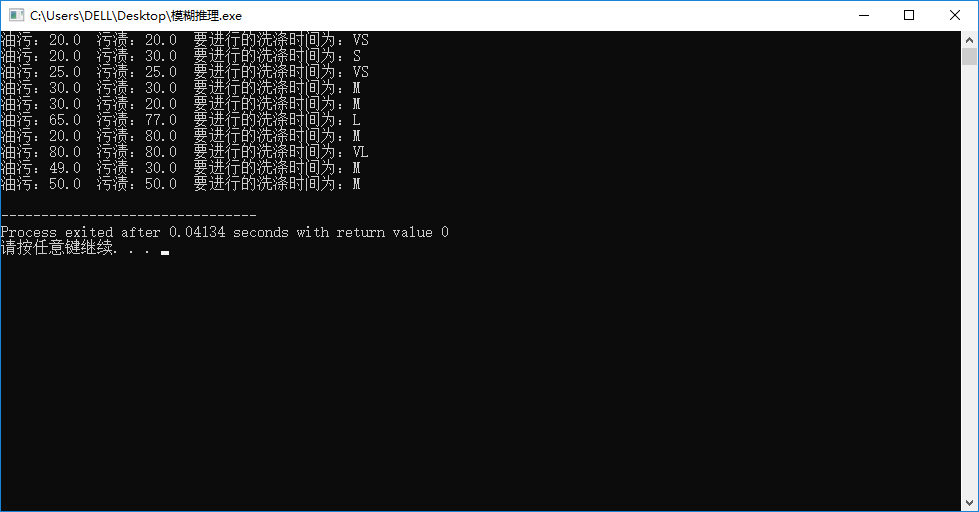
case 5:printf("VL\n");break;

}

}

return 0;

1. **结果展示**



1. **总结**