第27卷第1期 上海师范大学学报(自然科学版) Vol. 27, No. 1
1998年3月 J. of Shanghai Teachers Univ. (Natural Sciences) Mar, 1998

单片机模糊控制器的软件设计

王修才 冷 京 张 勇

提 要 介绍了用单片机语言进行模糊控制器软件设计的数据结构及其算法,并给出了程序流程.

关键词 模糊控制;软件设计;单片机;数据结构
中图法分类号 TP368.1

如图1所示,单片机模糊控制器系统和一般微机数字控制器就其系统结构而言没什么两样,所不同的是这里的控制器是用模糊控制算法代替了传统的PID算法.

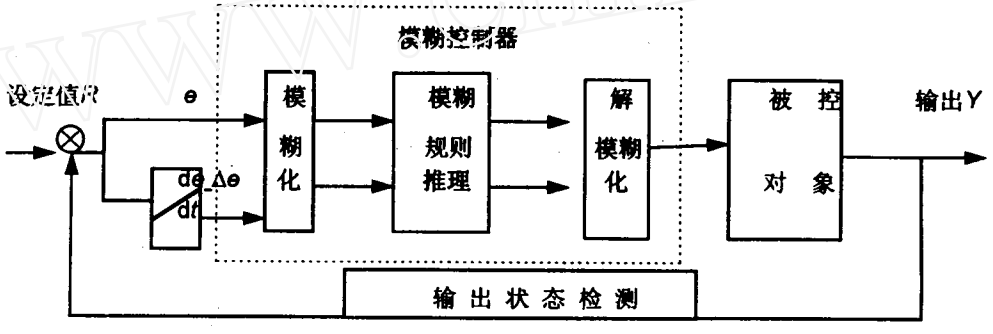

图1 单片机模糊控制系统

作者以最为流行的模糊控制算法max-min法[1]为例来说明上述3个模块的处理过程. 用单片机编程语言设计模糊控制软件的技术关键在于:(1)如何存放各个论域的隶属函数和如何求取特定输入量的隶属函数. (2)如何存放模糊推理规则和模糊输出量的隶属度.

1 模糊化

所谓模糊化,即把输入的精确量转化为模糊子集的过程. 可分两步进行:(1)取实时输入

收稿日期:1997-09-17
第一作者王修才,男,教授,上海师范大学理工信息学院,上海,200234

值,(2)把输入值与已定的隶属函数进行比较组合,产生相应的模糊输入量.

实践表明,控制系统的优良度只与所定义的隶属函数个数及其所覆盖的论域范围大小有关,而对隶属函数的形状并不敏感. 因此,为了简化计算、节约内存,常采样如图 2 所示的三角形隶属函数[2].

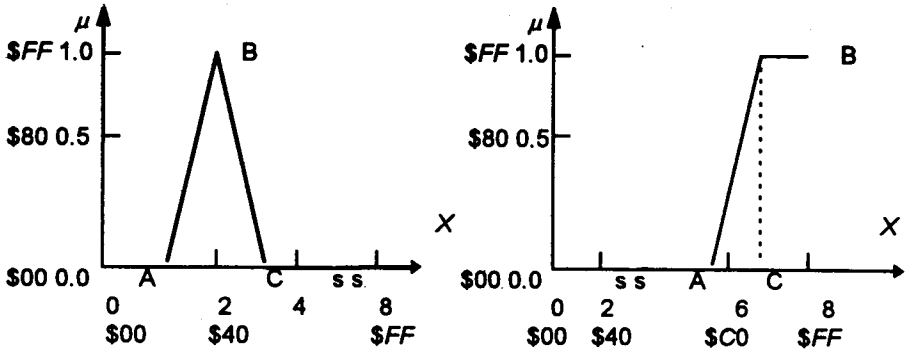

图 2(a) 三角形隶属函数算法示图　　图 2(b) 半三角形隶属函数算法示图

三角形隶属函数算法描述:对于图 2(a)所示三角形有左右两个底点,一个顶点. 因此存储一个三角形隶属函数需占用 3 个字节;图中水平坐标表示论域,垂直坐标表示隶属度. 隶属度取值范围本应 0~1,但对于 8 位单片机,为了编程方便,一般都是把论域和隶属度坐标最大用一个字节表示,即 $00～$ FF.

用 P_1,P_2 和 S 分别代三角形隶属函数的 3 个字节,即用 P_1 字节表示三角形左底点 A 的 x 坐标,$P_1 = xa$;用 P_2 字节表示顶点 B 的 x 坐标,即 $P_2 = xb$;用 S 字节表示右斜边的斜率,即 S = $ $FF(xc - xb)$;我们称其为点斜法. 确定了 P_1,P_2,S 后,就可以用来求取任一瞬时输入量 x 的隶属度 $\mu(x)$,在使用中应考虑下述 3 种情况:

① 当 $x < P_1$ 时,$\mu(x) = 0$;

② 当 $P_1 < x < P_2$ 时,$\mu(x) = (x - P_1) * \$FF(P_2 - P_1)$;

③ 当 $x >= P_2$ 时,$\mu(x) = \$FF - (x - P_2) * S$.

对于如图 2(b)中所示的半三角形(B,C 两点 x 值相同)隶属函数,只需要两个字节存储即可:1 个字节存放带斜边的底点 A,即 $P = xa$;1 个字节存放半三角形的斜边,即 S = $ $FF(xb - xa)$. 在使用中应注意 ① 当 $x < P$ 时,$\mu(x) = 0$;② 当 $x > P$ 时,$\mu(x) = (x - P) * S$;求得 $\mu >$ $ FF 时,取 $\mu =$ $ FF.

图 3 给出了对于任一瞬时输入值用点斜法求取其隶属函数的程序流程.

对于输出语言变量的定义,为了简化计算,通常把输出论域的隶属函数定义为如图 4(a)所示的单值点函数,或者,也可用三角形定义输出论域的隶属函数,如图 4(b);但是我们只取各隶属函数的中点值为其有效值,而其余部分均视为 0.

对于大多数单片机模糊控制系统而言,通常是把系统偏差和偏差变化率的实际范围作为输入论域,而把输出变量的允许变化范围作为输出论域;在每个论域上可取若干个语言变量. 例如图 5 是某温控系统输入语言变量 e 和 $\triangle e$ 的论域及其隶属函数.

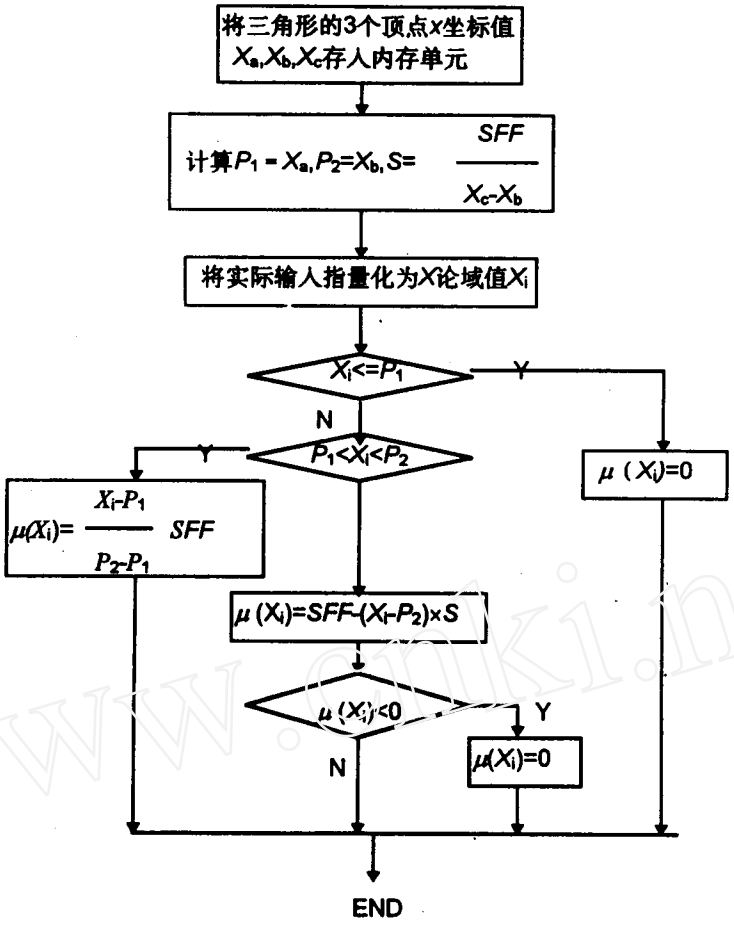

图 . 点斜法计算三角形隶属度

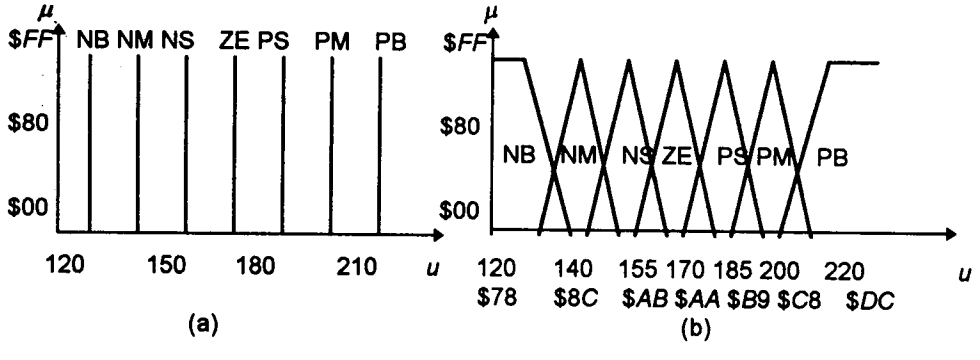

图 . 输出模糊量的两种定义

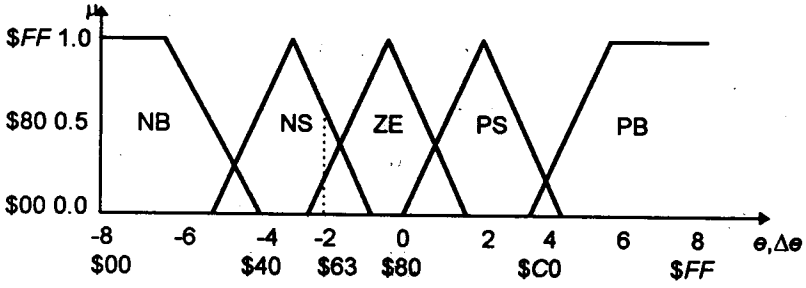

图 5 某温控系统输入模糊量的描述

以上我们对输入输出量的模糊化的算法作了描述,并给出了实施程序流程;但是,还有以下几个具体问题需要解决:

(1) 开辟适量的存储空间

例如对图 5 所示的隶属函数,如果用点斜法表示,那么存储偏差 e 的 5 个隶属函数就需要 15 个字节(3×5),同样,存储偏差变化率 $\triangle e$ 也需要 15 个字节;而对于图 5 所示的输出语言变量 u 的 7 个隶属函数,由于只需存入各隶属函数的中心元素值 Y_i,故开辟 7 个字节单元就够了,这样,一共需要 37 个字节的 ROM 空间就可存放该系统的输入输出语言变量所定义的隶属函数,并且按照顺序存放格式存储.

(2) 确定实时输入变量

从图 1 可以看出,该系统是一个典型的两输入单输出结构;输入变量是系统偏差和偏差变化率,而输出是模糊控制量. 于是,实时输入量为:

$$e^* = R - Y, \quad \triangle e^* = e_1 - e_2$$

其中,R 是设定值,Y 是被控对象当前输出;e_2 为系统上一时刻的偏差,e_1 是系统当前时刻偏差.

注意:①R 与 Y 或 e_1 与 e_2 应具相同的数制和量纲;

②如果输入量为模拟量,需经 A/D 转换成相应的数字量.

(3) 确定模糊输入量

把输入变量的实时值和已经定义的隶属函数进行比较组合求出相应的模糊输入量. 从图 5 可见,在对输入量进行模糊化时,两个输入量 e 和 $\triangle e$ 中的任何都会和该输入量所对应的 5 个隶属函数进行模糊化处理,会产生 10 个(2×5)模糊输入量,然而,由于我们所取隶属函数为具有一定的重叠率和重叠强度[3]的三角形,故对某个瞬时输入量只会有两个隶属函数与其交迭,即只会有两个模糊输入量不等于 0,而其余全为 0. 例如,某时刻偏差 e = \$63,它落在模糊变量 e 所定义的隶属函数"NS"和"ZE"之间;在图 5 上,如果在 x = \$63 处向上作垂线,它与隶属函"NS"和"ZE"产生交迭,故该偏差只可模糊化为"NS"和"ZE"模糊输入值. 如用点斜法还可计算出模糊输入量的隶属度:

$$\mu_{ns}(\$63) = (\$70 - \$63) * \$FF(\$70 - \$50) = \$68 \text{ 相当于十进制 } 0.4$$

$$\mu_{ze}(\$63) = (\$63 - \$60) * \$FF(\$80 - \$60) = \$18 \text{ 相当于十进制 } 0.1$$

即 e = \$63 属于"NS"的程度为 0.4,属于"ZE"程度为 0.1,而属于"NB","PS","PB"的程度均为 0. 对于偏差变化率 $\triangle e$ 的模糊量及其隶属度的求法与上述方法类似. 不再赘述. 最后,将 e 和 $\triangle e$ 经模糊化后得到的对应于各模糊量的类似度存储于内存中. 至此,我们完成了输入量

的模糊化过程.

2　模糊推理

模糊推理也称规则评价,通过评价和推理产生模糊输出. 系统的控制规则是根据操作者的经验总结归纳出来的;推理算法的效率取决于语言变量和控制规则在内存里的数据格式.

对于一共具有两输入单输出的模糊控制系统来说,推理规则的典型格式是:

if　$e = A_i$　and　$\triangle e = B_i$　then　$u = C_i$.

这种推理语句根据前件推断出后件.

由于控制规则少则几条,多则几十条,甚至上百条,考虑到节约内存和查找方便,要合理安排存放格式,必要时还应对规则数据进行压缩存储. 例如,某温控系统有如下几条规则:

R_1:　if　e=ZE　and　$\triangle e$=PS　then　u=NS;

R_2:　if　e=ZE　and　$\triangle e$=ZE　then　u=ZE;

R_3:　if　e=NS　and　$\triangle e$=NS　then　u=PS;

R_4:　if　e=NS　and　$\triangle e$=ZE　then　u=PS.

顺便指出,在规则 R_2 中,控制量 u=ZE=0,应理解为模糊量为 0,不是真正取值为 0,其真正值可以是某个电压值,比如是 170V. 见图 4(b).

现将 e、$\triangle e$ 和 u 的取值分别用数字代码表示,即定义

e、$\triangle e$:　NB=0,NS=1,ZE=2,PS=3,PB=4;

u:　NB=0,NM=1,NS=2,ZE=3, PS=4,PM=5,PB=6.

这样,我们就可以用 4 位二进制数表示输入、输出语言变量的各个取值;把每条的两个前件用一个字节表示,这条规则的第一前件放入高 4 位,第二前件放入低 4 位. 其后件也用一个字节表示,但是最高位恒为 1,以资区别. 于是,上述 4 条规则的数据结构为:

地址	数据	注释
$0100	$23	R_1 前件
$0101	$82	R_1 后件
$0102	$22	R_2 前件
$0103	$83	R_2 后件
$0104	$11	R_3 前件
$0105	$84	R_3 后件
$0106	$12	R_4 前件
$0107	$84	R_4 后件

4 条规则占用 8 个字节单元,n 条规则就需要 $2n$ 个字节单元. 采用这种顺序存储格式不仅节约内存,而且便于查找,从而提高了推理速度. 以下以规则 R1 为例说明推理查找过程. 见图 6. ① 取第一规则 R_1 的前件字节的高 4 位作为地址偏移量,加上模糊化后的偏差模糊输入量在内存 RAM 中存放的首地址(即表 2 所示存储区的首地址 $50),则可从 RAM 内偏差模糊量 e 的存放区域中查找出对应于当前特定的偏差输入量 e^* 隶属于该规则第一前件的语言变量"ZE"的隶属度 $\mu_{ze}(e^*)$;如果 $\mu_{ze}(e^*)$ 不等于 0(注),则再取该规则前件字节的低 4 位作为地

址偏移量,加上偏差变化率的模糊输入量在内存 RAM 中存放的首地址(即表 2 中地址 $55),则可从 RAM 内偏差变化率△$e$ 的区域中查找出对应于当前特定的偏差变化率△e^* 隶属于该规则第二前件所取的语言变量"PS"的隶属度 $\mu_{ps}(\triangle e^*)$.

② 根据 max-min 推理法,取两个前件隶属函数中的最小者作为该规则后件所取语言变量值"NS"的隶属度,即 $\mu_{ns}(u^*) = \min((\mu_{ze}(e^*), \mu_{ps}(\triangle e^*))$.

③ 取 R_1 后件字节的低 4 位作为地址偏移量,加上模糊输出量的 RAM 存放区首地址,便可求得对应与后件取值为"NS"的隶属度存放地址,而后将 $\mu_{ns}(u^*)$存入该地址单元中. 至此,我们就完成了规则 R_1 的查找与推理.

对 R_2,R_3 和 R_4 作类似处理,而且会有:

$\mu_{ze}(u^*) = \min(\mu_{ze}(e^*), \mu_{ze}(\triangle e^*))$;

$\mu_{ps}(u^*) = \min(\mu_{ns}(e^*), \mu_{ns}(\triangle e^*))$;

$\mu_{pb}(u^*) = \min(\mu_{ns}(e^*), \mu_{ze}(\triangle e^*))$.

对每条规则都作如此处理,便可求出当前所有模糊输出量的隶属度,并存放RAM单元中.

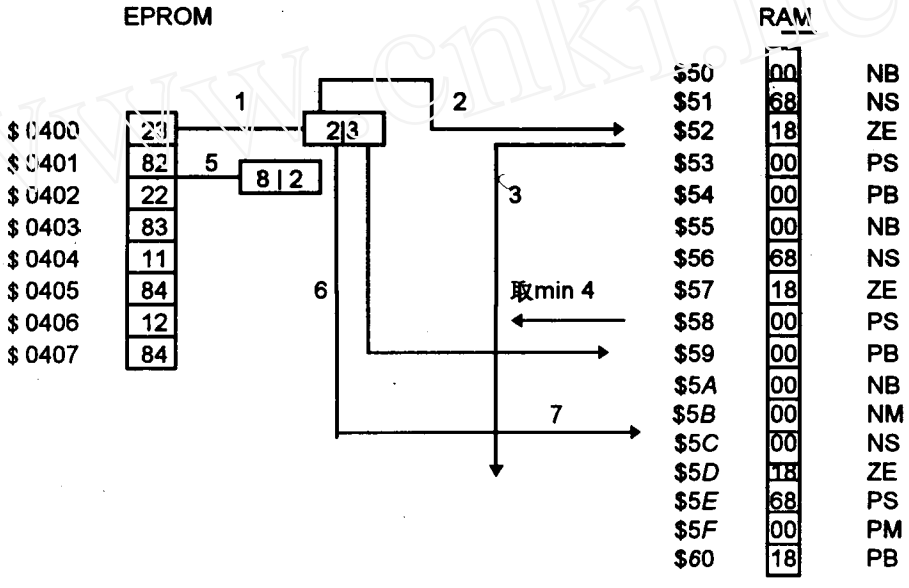

图 6 模糊推理过程示意图

$50～$54 为存储区,$55～$59 为△e 存储区,$5A～$60 为 μ 存储区.

3 解模糊化

解模糊化的目的就是求出所有隶属度不为 0 的模糊输出量对最后输出控制量的贡献. 根据 max-min 法,对所有推理后件取最大运算. 即求取所有模糊输出量的重心,从而得到精确量

u^*，即

$$u^* = \sum_{i=1}^{n} \mu_i Y_i / \sum_{i=1}^{n} \mu_i$$

式中 μ_i 为第 i 个模糊输出量的隶属度;Y_i 为该模糊输出量的单点位值或中心元素的位置;n 为系统输出模糊量的个数.

图 7 是模糊推理程序流程框图.

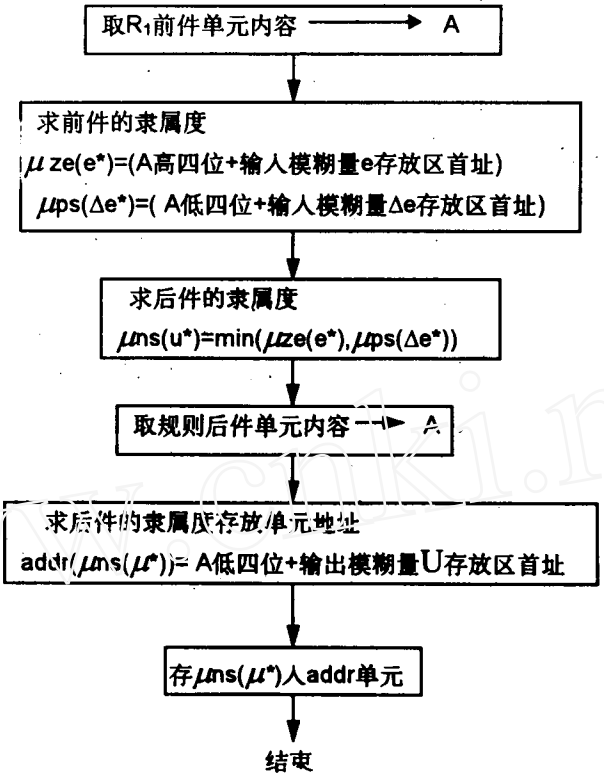

图 7　模糊推理程序框图

把上述 3 个模块连接起来,就可以实现基本模糊逻辑控制,因此主程序的编制应该包括如下几步:

STEP1:　定义各输入.输出量的模糊子集及其相应的模糊函数,按推荐的数据格式存放数据.

STEP2:　定义模糊控制规则,按推荐的数据结构存放.

STEP3:　采集输入数据,按 STEP1 中隶属函数进行模糊化.

STEP4:　根据模糊输入量从规则集中找出所有被激活的规则,并计算出模糊输出集.

STEP5:　用重心法求出实际控制量输出.

STEP6:　若时间未到,转 STEP3.

STEP7:　运行结束处理.

这种方法灵活性很大,具有很强的通用性,适用于绝大多数硬件运行环境.

有了数据结构,又有了程序流程图,编制程序就比较容易了;限于篇幅,MCS-51 单片机模糊控制程序从略.

参 考 文 献

1 戊月莉. 计算机模糊控制原理及应用. 北京:航空航天大学出版社,1995
2 余永权. 单片机模糊逻辑控制. 北京:航空航天大学出版社,1995
3 王修才等. 通用模糊控制器鉴定会资料. 1996
4 Wang Xiucai. Design of an Adaptive Fuzzy Control System. Proceeding of IEEE Inter-national Conference on Intelligent Processing System, 1997

The Software Design of Fuzzy Logic Cotroller with Single-chip Microcomputer

Wang XiuCai Leng Jing Zhang Yong

Abstract This paper not only introduces that how to realize the algorithm routine, but how to arrange the data structure of fuzzy logic controller. They both are key problems in realization of fuzzy logic controller.

Key words Fuzzy Logic Control; Software Design; Single-chip Microcomputer; Data Structure