

文章编号: 1002-0411(2000)06-0559-05

一种新型自学习模糊控制器

雷德明

(武汉交通科技大学自控系 武汉 430063)

摘要: 本文设计了一种新的自学习模糊控制器, 其模糊规则与隶属函数分别由整数编码遗传算法与混沌优化算法在线学习, 仿真结果验证了设计的合理性与有效性。

关键词: 学习, 遗传算法, 混沌, 模糊控制器

中图分类号: TP13

文献标识码: B

1 引言

自从马丹尼(Mamdani)首次在蒸汽发动机的压力与速度控制中尝试模糊控制以来, 模糊控制器在许多领域, 特别是家电、过程控制等领域应用广泛, 由于其具有万能的逼近能力, 能有效处理非精确信息, 使其在多输入、无数学模型的非线性复杂系统的控制方面具有其它控制方法无法比拟的优势。

应用模糊控制器的关键: 合理地确定相应论域的模糊子集的划分与建立完善有效的规则库。通常人们根据对受控对象的观测和自身的知识、经验, 运用模糊逻辑的方法确定一个模糊规则表, 然后根据此表计算控制量控制受控对象, 这种方法主观成分多, 不易获取有效的规则与隶属函数, 从而难以保证系统具有良好的动、静态特性。而且规则表一旦确定很难在控制过程中作适当调整, 缺乏适应性, 另外, 按上述方法所建立的模糊控制器精度低, 限制了它的应用。

解决上述问题的办法: 在模糊控制器中引入自学习算法, 使控制器具备自学习与自改善功能。通过不断学习系统性能的相关信息, 并以学习结果为基础修正(或生成)原有(或新的)模糊规则与隶属函数, 改善系统的动、静态特性。神经网络模糊控制器就是一类能学习的控制器, 它利用神经网络的学习能力, 改善控制器性能。但由于人工神经网络易陷入局部解, 收敛速度也慢, 使这种控制器难以应用于实时控制的场合。如何合理地设计高效的学习算法满足自学习模糊控制器的实时要求, 是建立自学习模糊控制器的关键?

遗传算法作为一种智能计算方法, 近年来, 一直是人们研究的焦点, 有关遗传算法在模糊控制器中的应用的论文也不断被报道^[1-3]。在模糊控制器中加入遗传算法, 利用该算法使控制器具有学习能力, 满足实时控制的需要, 是人们期待已久的事, 也是本文所要解决的问题。

2 关于模糊控制器学习问题的分析

模糊控制器的核心是模糊规则库, 它决定如何从输入得到输出。模糊规则的一般形式为:

$$\text{if } e \text{ is } A_i, \Delta e \text{ is } B_i \text{ then } u \text{ is } C_i$$

其中 e 、 Δe 分别为误差、误差变化率, 它们是模糊控制器的输入, u 为控制器输出, A_i 、 B_i 、 C_i 分别是上述三个语言变量论域上的模糊子集.

要建立完善的模糊规则库, 首先必须确定输入输出论域的模糊子集的划分, 因为这种划分会直接决定规则的语义及最大规则数.

2.1 隶属函数的在线学习

相应论域模糊子集的划分包括: 确定隶属函数的类型、个数、形状与中心点位置. 前二者由设计者确定, 而隶属函数的形状与位置需在控制过程中在线调整, 以不断改善控制品质. 如果同时优化形状与位置, 会导致优化参数过多, 搜索空间过大, 难以取得良好的学习效果, 往往也很难保证规则的完整性.

实际上, 并非每个隶属函数的形状与位置都需在线优化, 并且, 一般情况下, 根据如下原则: 误差与误差变化率大的区域, 模糊子集的分辨率低, 反之, 则高, 设计者能为每个隶属函数设置较合理的形状, 选择较适当的位置. 因此在设计者初步设计基础上, 仅在线优化部分隶属函数的位置, 然后根据计算结果, 在保证相邻隶属函数曲线重叠合理条件下确定其形状. 这样就能大大减少优化参数的个数, 以三角形隶属函数为例, 参数至少减少了 $2/3$, 又能保证规则的完整性, 改善学习效果. 另外, 每个被优化的位置都应设置相应的调整范围, 不能交叉, 以避免出现违背常识的划分. 如模糊集“正大”的中心位于“正中”的中心的左边.

本文采用下面的混沌优化算法在线优化隶属函数. 令

$$t_{k+1} = ut_k(1 - t_k) \quad (1)$$

$k = 0, 1, 2, \dots, 0 \leq t_0 < 1$, u 为控制参量, $u = 4$ 时, 系统(1)处于完全混沌状态.

优化问题一般形式: $\min f(x)$

$$a_i \leq x_i \leq b_i \quad i = 1, 2, \dots, n; \quad x = (x_1, x_2, \dots, x_n)$$

混沌优化算法的基本步骤如下:

STEP 1: 初始化: 对(1)中的 t_k 分别 n 个具有微小差异的初值, 可得到 n 个轨迹不同的混沌变量 $t_i(k)$, $k = 1$ $f^* = f(x^*)$, x^* 为当前最优解.

STEP 2: 用混沌变量搜索:

$$x_i(k) = x_i^* + C_i t_i(k) - d_i \quad (2)$$

计算性能指标: $f(k) = f(x(k))$, $x(k) = (x_1(k), x_2(k), \dots, x_n(k))$

STEP 3: 若 $f(k) < f^*$, 则 $x^* = x(k)$, $f^* = f(k)$, 如给定的迭代次数已达到, 则结束, 否则转(2).

2.2 模糊规则的在线学习

若 e 、 Δe 、 u 的语言取值均为 NB (负大)、 NM (负中)、 NS (负小)、 ZE (零)、 PS (正小)、 PM (正中)、 PB (正大), 则模糊规则表为 7×7 矩阵, 由于利用遗传算法学习模糊规则时, 规则表即学习对象, 因此首先应对规则表编码.

2.2.1 编码

整数编码: 若控制器输出 u 的语言取值为 NB 、 NM 、 NS 、 ZE 、 PS 、 PM 、 PB , 则编码后个体形如 - 3 - 2 - 10123 - 44... .., 其中 $[-3, 3]$ 内的整数对应有用规则, 其所在位置决定了 e 与 Δe 的语言取值, 其值则表示 u 的语言值大小, 如 - 3 表示 NB , - 2 表示 NM 等, 而 - 4, 4 则与无用规则相对应, 模糊推理时不使用这些无用规则.

尽管整数编码简单、直观, 但因整数编码遗传算法常收敛于未成熟解, 限制了它的应用, 为

克服其缺陷, 本文作如下处理:

(1) 取消交叉

整数编码遗传算法缺陷产生的主要原因在于: 交叉产生新个体的能力有限, 基因互换无法产生完全相异于互换基因的新基因.

(2) 所有个体机会均等地成长为最优个体

由于每个个体都含有一些最优或次优规则, 只是个数不同而已, 因而都可以通过适当增加进化次数等手段成为最优个体. 这样做既能有效避免陷入局部解, 同时也加快了种群进化.

2.2.2 如何提高学习效率?

在用遗传算法离线优化模糊规则时, 每个个体都对应一个模糊规则集, 要得到个体适应值, 就必须评价模糊控制器, 且每次必须评价多个, 这导致计算量显著增加; 同时, 由于适应值是针对整个模糊规则表的, 很难反映出单条规则的性能, 从而难以提高单条规则的性能. 以上两个方面的原因, 导致遗传算法的学习效率低. 为了满足在线控制对学习效率的要求, 采用如策略.

(1) 利用小群体遗传算法优化规则.

(2) 多点变异

由于仅改变一、二条规则, 很难引起适应值明显的改变, 只有多点变异, 才可能明显地改善适应值.

多点变异: 随机地选择多个位置, 然后从每个选中的基因码的一些可能取值中随机地选出一个, 替代原基因码. 若新个体的适应值大于原个体的适应值, 则取代原个体, 否则原个体不变.

变异时, 适应值低的个体, 变异的位置多, 次数也相应增加, 而适应值高的个体, 其变异次数与位置数则少一些. 另外, 为了加快进化, 所有个体被分成长度相差较大的两段. 然后对长度大的那段进行多点变异, 每隔一定代数后, 集中对最优个体中较短的那段进行多次变异.

(3) 简化适应值函数

不再为得到个体的适应值而评价模糊控制器, 这样可大大降低每次进化的计算量. 如: $F_i = a_1 |e| + a_2 |\Delta e|$, a_1, a_2 为加权系数.

(4) 种群置换

由于遗传算法在搜索接近全局最优解时, 收敛速度明显减慢, 为了加快搜索, 往往与局部方法混合. 因规则表优化比较特殊, 很难找到合适的局部方法与遗传算法混合. 为了提高遗传算法的学习效率, 采用种群置换策略, 即当误差接近 0 时, 随机产生新群体, 新群体通过几代进化后, 被更新的种群替换, 这样不断更换群体, 直到学习过程结束. 因为新种群的进化能力往往强于那些经过多代进化的种群, 种群不断更迭, 就能加快搜索.

2.3 学习算法

本文设计的学习算法由混沌优化算法与遗传算法组成, 它们分别用于隶属函数与模糊规则的在线学习, 其主要步骤如下:

(1) 对规则表编码, 生成初始群体, 针对每个个体, 根据知识、经验, 选择隶属函数的类型、个数、形状与位置. 计算各个体的适应值.

(2) 根据复制概率 $P_i = F_i / \sum_i F_i$, 按给定的种群规模选择个体进入下一代.

(3) 从种群中选择一定数量适应值各异的个体进行多点变异.

(4) 利用混沌优化算法对部分个体对应的部分隶属函数的位置作优化计算, 并确定相应的形状, 若个体的适应值提高了, 则用新的位置、形状替代旧的, 否则不变.

(5) 计算各个体的适应值并转(2).

关于上述学习算法作两点说明

(1) 无论是混沌优化算法, 还是遗传算法, 都不是用来搜索最优的隶属函数分布及最优的规则表, 主要是希望通过这两种算法的学习与调整, 使控制器性能不断改善, 同时满足实时控制的要求.

(2) 混沌优化时目标函数即遗传优化时的适应值函数. 遗传优化时, 有用规则的优化与无用规则的删除同时进行, 以减少那些冗余的或缺乏良好定义, 引起冲突的规则对控制性能的影响.

3 仿真

利用上述学习算法, 设计一种自学习模糊控制器如图1所示.

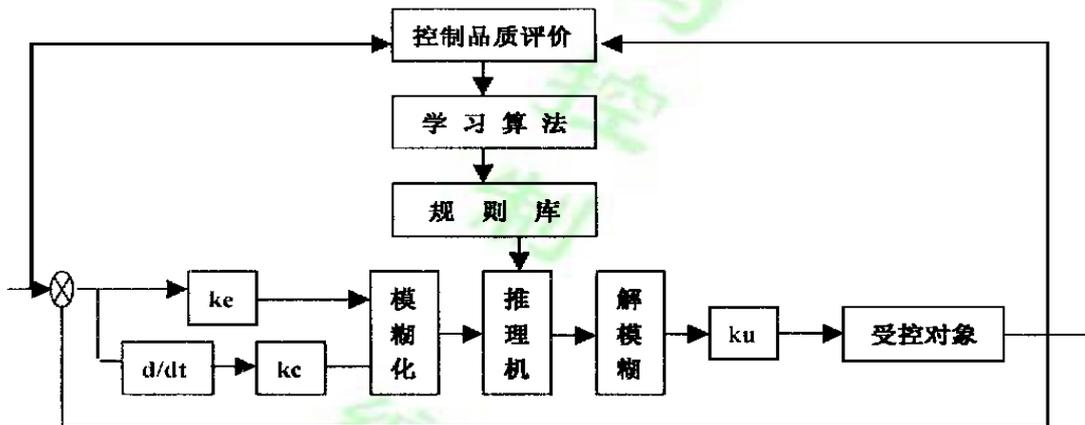


图1 自学习模糊控制器的结构

仿真时, 采用单值模糊化方法, 高斯型隶属函数即 $u_{Gi}(x) = \exp[-\frac{x_i - x_i^*}{\delta_i}]^2$, 以及中心平均反模糊化方法, 受控对象为非线性的倒立摆, 其状态方程如下:

$$\begin{cases} \dot{\theta} = \frac{d\theta}{dt} \\ \ddot{\theta} = \frac{g \sin(\theta) - (\cos(\theta)/(m_p + m_c))(m_p l \dot{\theta}^2 \sin(\theta) + F(t))}{4l/3 - m_p l \cos^2(\theta)/(m_p + m_c)} \end{cases}$$

式中 $g = 9.8 \text{ m/s}^2$, m_p 为杆的质量, m_c 为小车的质量, l 为 $1/2$ 杆长, $F(t)$ 为外作用力, 控制目的就是通过调整施加给小车的水平力使杆处于垂直位置. 仿真时 $m_p = 0.1 \text{ kg}$, $m_c = 0.9 \text{ kg}$, $l = 0.5 \text{ m}$, 用四阶龙格-库塔法对倒立摆进行仿真. 仿真结果如图 2、3 所示.

4 结论

为了满足自学习模糊控制的实时要求, 学习算法应具有高计算效率, 并且不易陷入局部解, 但现有的算法往往很难具备这两个性质, 本文提供一种有效的途径使模糊控制器具有自主

学习能力.

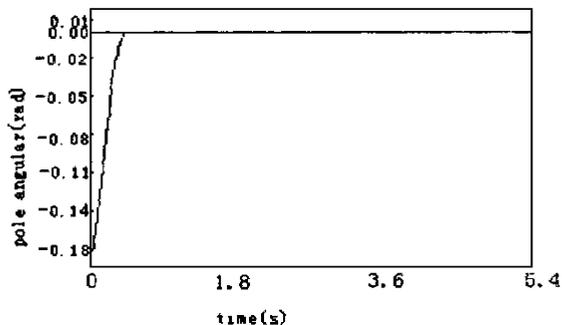


图 2 仿真结果

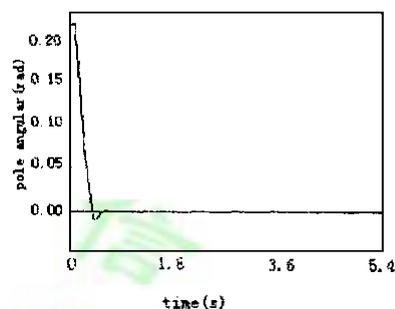


图 3 仿真结果

参 考 文 献

- 1 Chin T C, Qi X M. Genetic Algorithm for Learning the Rule Base of Fuzzy Logic Controller. *Fuzzy Sets and Systems*. 97: 1- 7, 1998
- 2 张晓绩, 戴冠中, 徐乃平. 遗传算法在抽取与过滤模糊控制规则中的应用. *控制理论与应用*, 1998, 15(3): 379~ 384
- 3 胡 炜, 沈 理. 遗传优化模糊逻辑控制器. *计算机科学*, 1997, 21(6): 10~ 16
- 4 雷德明. 利用混沌搜索全局最优解的一种混合遗传算法. *系统工程与电子技术* (待发表)

A NEW TYPE OF SELF-LEARNING FUZZY CONTROLLER

LEI De-m ing

(Wuhan transportation University, Wuhan 430063)

Abstract: A new self-learning fuzzy controller is designed in this paper. Its fuzzy rules and membership functions are on-line learned separately by the integer coded genetic algorithm and the chaotic optimization algorithm. The simulation results demonstrate the validity of the design.

Keywords: Learning, genetic algorithm, chaos, fuzzy controller

作者简介

雷德明(1968-), 硕士, 讲师. 研究领域为进化算法及其在控制中的应用.