

最优模糊控制的两种设计方法

金耀初 蒋静坪

(浙江大学电机系 杭州 310027)

提 要 本文基于二次型性能指标,给出了采用梯度法和遗传算法的模糊控制规则优化方法,并比较了它们的优缺点。仿真结果表明,这两种方法对未知系统的模糊规则优化具有明显的应用价值。

关键词: 模糊控制 优化 梯度法 遗传算法

1 引 言

自六十年代创立模糊集合理论以来,它在控制系统中的应用所取得的成功最为引人注目^[1]。近年来,随着人工智能理论、电力电子技术和计算机技术的发展,模糊控制不但在理论上取得重大突破,在工程应用中也日益显示出其特殊的优点,成为一种最具实际应用价值的智能控制器。

传统模糊控制器存在的主要不足是过于依赖控制专家的经验,学习能力不强。因此,控制规则的自学习越来越引起人们的重视。目前,神经网络与模糊控制的成功结合给模糊控制注入了新的活力^[2],模糊控制规则的提取也从简单的经验方法走向系统化的模糊辨识,模糊规则所蕴含的知识由浅层转向深层。

本文考虑模糊控制规则的优化问题,分别采用梯度法和遗传算法实现,并对它们各自的特点进行了比较。结果发现,这两种方法都对系统知识要求甚少,且具有良好的控制品质,适用于具有非线性、时变性和不确定性的系统控制。由于模糊控制本质上是一种模糊建模过程,因此它还适合于复杂系统的建模和预报。

2 模糊规则的优化

考虑如下非线性系统:

$$X(K+1) = f(X(K), \dots, X(K-P), u(K), \dots, u(K-Q)) \quad (1)$$

式中, $f(\cdot)$ 为某未知的非线性函数, $X(K), \dots, X(K-P), u$

$(K), \dots, u(K-Q)$ 分别为系统状态和控制量。假设控制系统采用如下模糊规则:

R^1 : 若 $e(K)$ 为 \tilde{A}^1 , $\Delta e(K)$ 为 \tilde{B}^1 , 则 $u(K)$ 为 u_1 ;

R^2 : 若 $e(K)$ 为 \tilde{A}^2 , $\Delta e(K)$ 为 \tilde{B}^2 , 则 $u(K)$ 为 u_2 ;

⋮

R^N : 若 $e(K)$ 为 \tilde{A}^N , $\Delta e(K)$ 为 \tilde{B}^N , 则 $u(K)$ 为 u_N 。

式中, $e(K) = X^d(K) - X(K)$, $X^d(K)$ 为 K 时刻系统的期望状态, $\Delta e(K)$ 为误差变化, $u(K)$ 为控制器输出, \tilde{A}^i, \tilde{B}^i ($i=1, 2, \dots, N$) 分别为相应论域上的模糊子集, 常采用高斯型或三角形隶属函数, u_i ($i=1, 2, \dots, N$) 则为每条模糊规则的清晰输出值。于是上述模糊规则系统可得到如下控制输出:

$$\mu_i(K) = \tilde{A}^i(e(K)) \wedge \tilde{B}^i(\Delta e(K)) \quad (2)$$

$$u(K) = \sum_{i=1}^N \mu_i(K) \cdot u_i / \sum_{i=1}^N \mu_i(K) \quad (3)$$

式中, ' \wedge ' 为极小运算, $\mu_i(K)$ 为第 i 条规则的真值。

模糊规则的优化目的, 就是根据系统的具体情况, 不断调整每条规则的前提参数(隶属函数参数)和结论参数(控制量), 使系统获得最佳的静态性能。为此, 根据最优控制的一般方法, 采用如下二次型性能指标:

$$J = \frac{1}{2} \{ [X^d(K) - X(K)]^T Q [X^d(K) - X(K)] + u^T(K) R u(K) \} \quad (4)$$

式中, Q 和 R 为对称的正定权矩阵, 可用于调整系统的性能。

2.1 梯度法

为了使性能指标(J)极小, 最直接的方法是采用梯度法优化方法。为推导方便, 假设模糊子集 \tilde{A}^i 和 \tilde{B}^i 均采用如下高斯型隶属函数(对非高斯型隶属函数, 作适当变换后即可):

$$\mu(z) = \exp(-b \cdot (z - a)^2) \quad (5)$$

于是, 对控制量 u_i , 根据梯度法有:

$$\begin{aligned} \frac{\partial J}{\partial u_i} &= \frac{1}{2} \cdot \frac{\partial [X^d(K) - X(K)]^T Q [X^d(K) - X(K)]}{\partial u_i} \\ &\quad + \frac{1}{2} \frac{\partial [u^T(K) R u(K)]}{\partial u_i} \\ &= -Q[X^d(K) - X(K)] \frac{\partial X(K)}{\partial u_i} + R u(K) \\ &\quad \cdot \frac{\partial u(K)}{\partial u_i} \end{aligned}$$

本文于 1995 年 6 月 12 日收到, 于 1995 年 8 月 7 日改回。系国家自然科学基金资助项目。

金耀初 1966 年生, 浙江大学电机系讲师, 在职博士生。曾独立或参与完成专著及译著 3 部, 在 IEEE 会刊等国内外主要刊物发表论文近 20 篇。目前主攻复杂系统的智能建模与控制。

蒋静坪 1935 年生, 浙江大学电机系教授, 博士生导师。主要从事自动控制理论及应用, 计算机控制与智能控制, 机器人控制等方面的教学、科研工作。

$$\begin{aligned}
&= \left[-Q \cdot e(K) \cdot \frac{\partial X(K)}{\partial u(K)} + Ru(K) \right] \\
&\quad \cdot \frac{\partial u(K)}{\partial u_i} \\
&= \left[-Q \cdot e(K) \cdot \frac{\partial X(K)}{\partial u(K)} + Ru(K) \right] \\
&\quad \cdot \mu_i(K) / \sum_{i=1}^N \mu_i(K) \quad (6)
\end{aligned}$$

上式中, $\partial X(K)/\partial u(K)$ 为系统状态与系统控制量的偏导比, 它反映了系统状态随系统控制量的变化趋势。当系统(1)部分已知时, 偏导很容易求取。当系统(1)未知或不确定时, 式(6)可用下式来代替:

$$\begin{aligned}
\frac{\partial J}{\partial u_i} &= \{ -Qe(K) \text{sign}[(X(K) - X(K-1)) / \\
&\quad (u(K) - u(K-1))] + Ru(K) \} \\
&\quad \cdot \mu_i(K) / \sum_{i=1}^N \mu_i(K) \quad (7)
\end{aligned}$$

式中, $\text{sign}(\cdot)$ 为符号函数。容易证明, 用式(7)代替式(6)后, 梯度算法依然收敛^[3]。在实际应用中, 为避免 $u(K)$ 与 $u(K-1)$ 相等时出现的计算困难, 本文用 $[X(K) - X(K-1)] \cdot [u(K) - u(K-1)]$ 代替 $[X(K) - X(K-1)] / [u(K) - u(K-1)]$ 。很显然, 在符号函数中, 两者是等价的。因此, u_i 的修正公式如下:

$$\begin{aligned}
u'_i &= u_i - \eta \cdot \frac{\partial J}{\partial u_i} \\
&= u_i - \eta \cdot \{ -Qe(K) \cdot \text{sign}[(X(K) - X(K-1)) \cdot (u(K) - u(K-1))] \\
&\quad + Ru(K) \} \cdot \frac{\mu_i(K)}{\sum_{i=1}^N \mu_i(K)} \quad (8)
\end{aligned}$$

其中 η 为大于零的学习速率。同理可分别得到隶属函数参数 a_i 和 b_i 的学习公式:

$$\begin{aligned}
a'_i &= a_i - 2\xi \cdot \{ -Q \cdot e(K) \cdot \text{sign}[(X(K) - X(K-1)) \cdot (u(K) - u(K-1))] \\
&\quad + Ru(K) \} \cdot \mu_i(K) \cdot (u_i - u(K)) \\
&\quad \cdot b_i(z - a_i) / \sum_{i=1}^N \mu_i(K) \quad (9)
\end{aligned}$$

$$\begin{aligned}
b'_i &= b_i + \xi \cdot \{ -Q \cdot e(K) \cdot \text{sign}[(X(K) - X(K-1)) \cdot (u(K) - u(K-1))] \\
&\quad + Ru(K) \} \cdot \mu_i(K) (u_i - u(K)) \\
&\quad \cdot (z - a_i)^2 / \sum_{i=1}^N \mu_i(K) \quad (10)
\end{aligned}$$

式(9)、(10)中, z 表示 $e(K)$ 或 $\Delta e(K)$, 至于究竟更新 $e(K)$ 的隶属函数参数还是 $\Delta e(K)$ 的隶属函数参数, 一般方法是选取产生极小输出的那个变量。但在初始隶属函数分布不佳的情况下, 这种方法不一定合理, 此时可采用文[4]建议的一步搜索法, 可取得较好的结果。

从上述过程可看到, 梯度法具有以下优点:

1) 模糊规则的前提参数和结论参数均能在某种性能指标的意义下进行优化, 且这种优化是实时进行的。

2) 优化过程对系统并无苛刻的先验知识要求。

同时, 梯度法有以下不足:

1) 容易陷入局部最小。

2) 用式(7)代替式(6)在多输入多输出情况下使学习速度减慢^[5]。

2.2 遗传算法

遗传算法是六十年代后发展起来的一种优化算法^[6]。它通过模拟生物的进化机制, 实现了搜索空间的全局优化。遗传算法引入了三个主要算子, 分别称为基因复制, 基因重组和基因突变。遗传算法的一般步骤见图1, 本文为了克服遗传算法存在的早熟性收敛并防止搜索时间过长, 作以下改进:

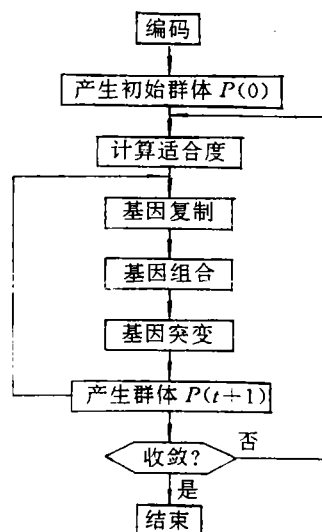


图1 遗传算法的实现过程

Fig. 1 The implementation diagram of GA

1) 引入“杂交”算子。即按概率随机产生一定的新基因作为父代的一方, 与从原群体中选取的基因作为父代的另一方进行基因重组, 产生具有新特征的子代, 以避免“近亲繁殖”。

2) 引入“优生”措施。即选取一定比例的性能最佳的个体, 直接作为子代的个体。

容易看出, “杂交”算子的引入主要用于克服早熟性收敛, 使寻优算法得到全局解。同时, “优生”措施保证了群体中的最佳个体不被丢失, 从而避免了搜索不易收敛的缺陷。一般说来, 在群体的总适合度较小时, “杂交”概率取稍大些, “优生”比例取稍小些; 当群体总适合度增大到一定程度时, “杂交”概率减小, “优生”比例增大。通过这种调整, 能较好地解决早熟收敛和搜索时间之间的矛盾。

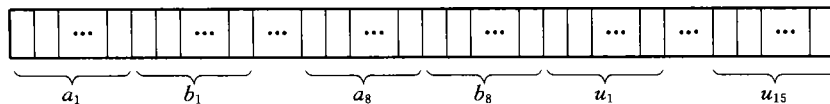
由于遗传算法的寻优结果是使适合度最大, 故每个个体的适合度取为:

$$F_i = \frac{1}{1 + J'_i} \quad (11)$$

式中,

$$J'_i = \sum_{K=0}^M \left\{ \frac{1}{2} [X^d(K) - X(K)]^T Q [X^d(K) - X(K)] + \frac{1}{2} u^T(K) R u(K) \right\} \quad (12)$$

其中 M 为考察每个个体的时间长度。从理论上来说, M 一般取 5 步左右即可, 但考虑到实际应用的情况, 一般 M 为系统



即基因总长度为 260 位。仿真中, 群体总数取为 50。

从遗传算法的优化设计过程看出, 它具有以下特征:

1) 遗传算法的搜索在所有参数空间上同时进行, 并同时考虑多个点的搜索, 有很强的并行运算能力。

2) 遗传算法对系统本身并无任何先验要求, 只需给定恰当的评价指标即可, 因而具有更广的应用领域。

但目前遗传算法的在线优化能力不强。

3 仿真研究

考虑如下非线性系统:

$$X(K+1) = \frac{X(K) \cdot X(K-1)}{(1 + X^2(K) + X^2(K-1))} + u(K-1) + \sin(X(K)) \cdot u(K) \quad (13)$$

设定的期望响应为方波曲线。图 2 为采用梯度法时系统的响应结果, 其中, Q 取 0.6, r 取为 0.01。可以看出, 系统的动、静态特性都是令人满意的。图 3(a)、(b) 分别为优化后的误差和误差变化的隶属函数分布。

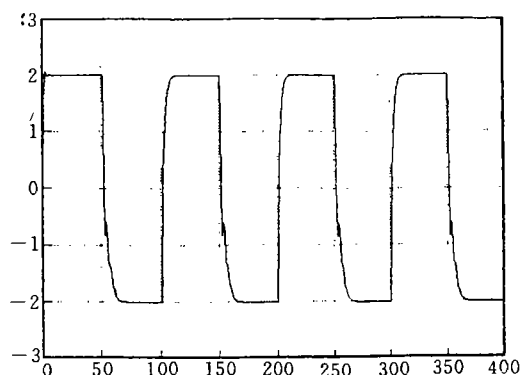


图 2 梯度法寻优下的系统响应

Fig. 2 System response under gradient optimization

图 4 为遗传算法优化后的系统响应结果。与梯度法相比, 上升时间减少, 但也出现了微弱的振荡。图 5(a)、(b) 则为遗传算法优化后模糊规则前提参数的分布。我们发现, 它并未与梯度法完全一致。这种不一致性可以从以下两个方面来说明:

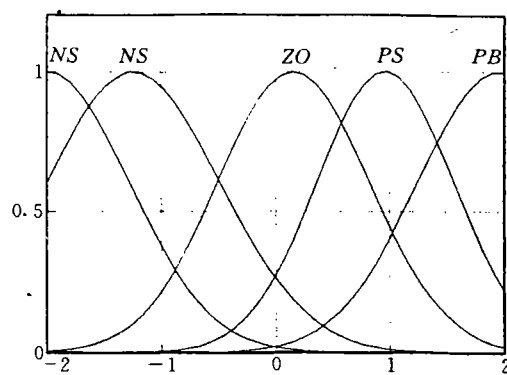
1) 某变量优化后的隶属函数形状应充分反映该变量在样本空间的分布。由于两种优化方法在具体调节过程中误差和误差变化的样本分布不一致, 由此得到的隶属函数也就不

仿真的总长度。

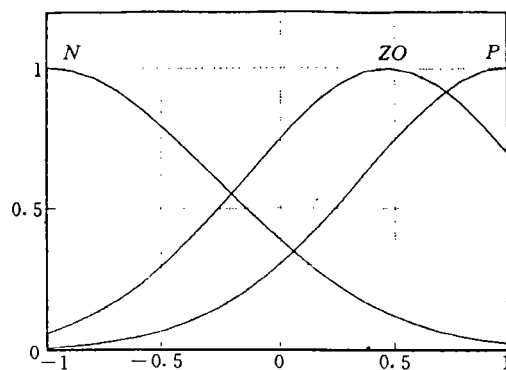
在编码时, 为了限制个体的基因长度, 以缩小搜索空间, 本文把误差、误差变化和控制量分别规范在 $[-2, 2]$, $[-1, 1]$ 和 $[-4, 4]$ 内。其中, 误差和误差变化分别划分成 5 个和 3 个模糊子空间, 因而共有 15 条模糊规则。若隶属函数也取为高斯型, 则隶属函数共有 16 个参数, 加上 15 个结论参数, 需搜索的参数共有 31 个。编码方式如下:

一致。

2) 由于模糊系统的复杂性, 其规则的优化结果并不是唯一的, 可能存在若干性能指标相近的局部最优解。这些解尽管不是严格的全局最优解, 但也是充分最优解, 且能很好地满足实际应用的需要。



(a)



(b)

图 3 梯度法寻优下的隶属函数分布 (a) 误差 (b) 误差变化

Fig. 3 Membership function distribution under gradient optimization a) error b) error change

总结两种不同方法得到的仿真结果, 我们可以得到以下结论:

1) 对由误差和误差变化构成的二维模糊控制器来说, 控制专家的经验判断基本符合客观情况, 但应根据不同的对象作一定的调整和再学习。而且, 模糊规则的优化结果并不一定

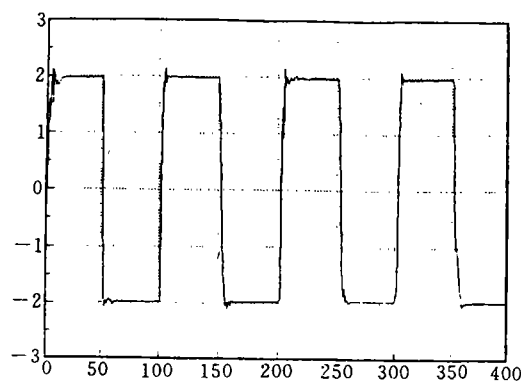
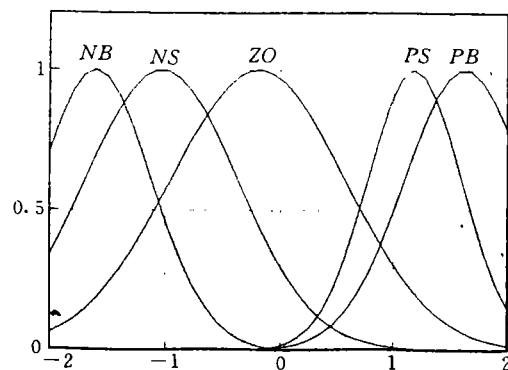
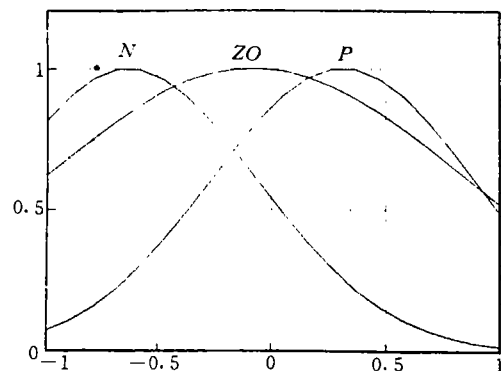


图4 遗传算法寻优下的系统响应

Fig. 4 System response under genetic optimization



(a)



(b)

图5 遗传算法寻优下的隶属函数分布 (a)误差 (b)误差变化

Fig. 5 Membership function distribution under genetic optimization a) error b) error change

是唯一的。

2) 采用优化的模糊规则可大量减少模糊规则数。在本文

中,模糊规则仅有15条。若采用未经优化的经验规则,模糊规则一般在49条以上。

3) 梯度法和遗传算法均能在复杂系统的优化控制中取得良好的结果。相比之下,梯度法适合于在线优化,而遗传算法则适合于复杂未知系统的优化。不言而喻,若把遗传算法用于系统的离线学习,而梯度法用于在线微调,则将具有更好的应用效果。目前,本文作者正着手这方面的工作。

4) 本文提出的模糊规则优化方法同样适用于复杂系统的建模和预测。

4 结束语

本文讨论了模糊控制规则的优化问题,给出了基于梯度法和遗传算法的两种设计方法。仿真结果表明,模糊控制规则的优化能使模糊控制器在规则减少的情况下依然获得良好的效果。本文的研究还表明,梯度法和遗传算法具有各自的特点,若两者能取长补短则更为有效。

5 参考文献

- 1 Benrenji H R. The Unique Strength of Fuzzy Logic Control. IEEE Expert, 1994; 9(2): 9-13
- 2 Kosko B. Neural Networks and Fuzzy Systems: A Dynamic System Approach to Machine Intelligence. N J: Prentice-Hall, 1992
- 3 金耀初. 智能控制理论及应用. 浙江大学研究生讲义. 1993
- 4 Jin Yaochu, Jiang Jingping, Zhu Jing. Adaptive Fuzzy Identification and Its Application. Int J of Systems Sci, 1995; 26(2): 197-212
- 5 金耀初, 蒋静坪, 诸静. 结合模糊推理的多变量神经自适应控制, 信息与控制, 1994; 23(4): 223-228
- 6 Goldberg D E. Genetic Algorithms in Search, Optimization and Machine Learning. M A: Addison-Wesley, 1989

Two Approaches to Optimal Fuzzy Control

Jin Yaochu Jiang Jinping

(Zhejiang University Hangzhou 310027 China)

Abstract This paper compares two approaches to fuzzy control rule optimization, namely, gradient method and genetic algorithm, on the basis of a quadratic performance index. The advantages and disadvantages of each method are discussed carefully. The simulation results show that they are both quite valuable in fuzzy rule optimization for unknown systems.

Key Words: fuzzy control optimization gradient method genetic algorithm