

文章编号:1671-6833(2003)04-0006-05

基于神经网络的自学习模糊控制

冯冬青, 张希平

(郑州大学电气工程学院, 河南 郑州 450002)

摘要: 将神经网络与模糊控制相结合, 提出了一种基于神经网络实现自学习模糊控制的方法, 并给出了神经网络训练、控制器离线自学习、控制器在线自学习的相应算法. 利用该方法, 可以实现控制器的离线自学习和在线自学习, 从而在控制对象发生变化时, 通过控制器自学习改善系统的控制性能. 仿真结果表明了该方法的有效性.

关键词: 模糊控制; 神经网络; 自学习; 仿真

中图分类号: TP 273.5 **文献标识码:** A

0 引言

模糊控制技术和神经网络控制技术是智能控制技术的两个主要组成部分, 两者有着强烈的互补性, 将两者结合是智能控制技术研究与应用的一个发展方向. 神经网络和模糊控制的融合方式有三种: 基于神经网络的模糊控制、模糊神经网络和用模糊逻辑增强的神经网络^[1].

基于神经网络的模糊控制有多种实现方法. 其中一种是利用多层神经网络逐层实现模糊控制算法, 如首先用一个三层网络实现模糊化, 再用多个两层网络实现隶属度函数求取, 然后用一个四层网络存储控制规则, 最后用一个三层网络实现反模糊化, 整个网络有十层. 对这种方法适当简化后用的网络层数虽然较少, 但也常常达到五、六层^[1,2]. 网络层数太多势必造成学习算法复杂, 对学习参数的选择要求较高. 另一种方法是利用神经网络记忆模糊逻辑推理系统输入输出论域中对应数据的影射关系, 以实现基于神经网络的模糊控制, 所用网络层数较少^[3,4]. 本文则利用神经网络较强的非线性映射能力学习模糊控制系统对典型输入信号响应过程中模糊逻辑系统输入输出数据的对应关系, 用三层神经网络实现模糊控制, 并用反向传播算法实现控制对象参数变化时控制器的离线和在线自学习.

1 用神经网络实现的模糊控制

工程中许多过程可以近似为一阶或二阶对象. 对这类对象, 可以用模糊控制实现较好的控制性能. 虽然模糊控制对过程参数变化有较强的适应能力, 但是如果控制对象参数变化较大, 模糊控制的效果仍会变差. 提高模糊控制对过程参数变化适应能力的方法多种多样, 用神经网络实现模糊控制及控制器的自学习是一种有效的方法. 这里我们首先用神经网络实现模糊控制.

1.1 模糊控制

以二阶对象为例, 设对象模型为

$$G(s) = \frac{12}{10s^2 + 6s + 1} \quad (1)$$

控制系统结构如图 1 所示, 采用增量式模糊控制. 隶属度函数选择图 2 所示对称等腰三角形, 控制规则如表 1 所示, 其中 E_1, dE_1, U_1 分别表示 e_1, de_1, du_1 的模糊量; NB, NM, NS, ZO, PS, PM, PB 分别为模糊集负大、负中、负小、零、正小、正中、正大, 模糊推理及非模糊化采用 MIN-MAX-重心法. 量化因子取 $k_e = 5, k_c = 20$; 比例因子取 $k_u = 0.001$; 采样时间取 $t_s = 0.02s$. 则控制系统的阶跃响应曲线如图 3 所示. 由阶跃响应曲线可以看到模糊控制的效果比较好.

1.2 用神经网络实现的模糊控制

下面我们神经网络代替图 1 中的模糊控制

收稿日期: 2002-09-05; 修订日期: 2002-10-11

基金项目: 河南省自然科学基金资助项目(0311011300)

作者简介: 冯冬青(1958-), 男, 广东省佛山市人, 郑州大学教授, 上海大学博士研究生, 主要从事智能控制及计算机控制方面的研究.

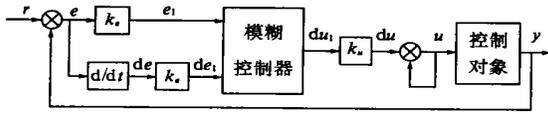


图 1 模糊控制系统

Fig. 1 Fuzzy control system

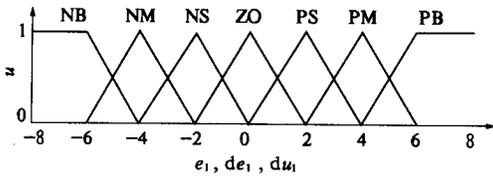


图 2 e_1, de_1 和 du_1 的隶属度函数曲线

Fig. 2 Membership functions of variables e_1, de_1 and du_1

表 1 模糊控制规则

Tab. 1 Fuzzy control rules

E_1	dE_1						
	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NB	NM	
NM	NM	NM	NM	NM	NS	NS	NS
NS	NM	NS	NS	NS	ZO	ZO	ZO
ZO	NM	NS	NS	ZO	PS	PS	PM
PS	ZO	ZO	ZO	PS	PS	PS	PM
PM	PS	PS	PS	PM	PM	PM	PM
PB	PM	PM	PB	PB	PB	PB	PB

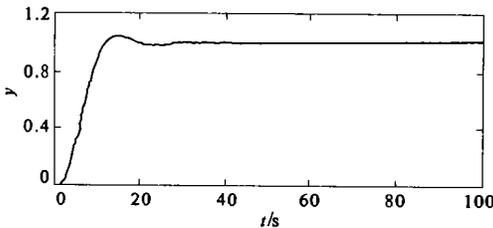


图 3 模糊控制系统的阶跃响应曲线

Fig. 3 Step responses for fuzzy control system

器. 神经网络的结构如图 4, 为 2-8-1 结构, 输入为 e_1, de_1 , 输出为 du_1 . 第一层神经元的作用仅是接受输入信号并分别传送给第二层各神经元. 各层神经元的序号分别为 i, j, k , 状态分别为 x_i, x_j, x_k , 传递函数分别为 $f_1(x) = x; f_2(x) = \frac{1-e^{-2x}}{1+e^{-2x}}; f_3(x) = x$. 神经元的权值、阈值、加权输入分别用 w, b, net 表示. 则

$$net_{i=1} = e_1 \quad (2)$$

$$net_{i=2} = de_1 \quad (3)$$

$$net_j = \sum_{i=1}^2 (w_{ji}x_i + b_j) \quad (4)$$

$$net_k = \sum_{j=1}^8 (w_{kj}x_j + b_k) \quad (5)$$

$$x_{i=1} = f_1(net_{i=1}) = e \quad (6)$$

$$x_{i=2} = f_2(net_{i=2}) = de \quad (7)$$

$$x_j = f_2(net_j) \quad (8)$$

$$x_k = f_3(net_k) = du_1 \quad (9)$$

$$f_1'(net_i) = 1 \quad (10)$$

$$f_2'(net_j) = 4e^{-2net_j} / (1 + e^{-2net_j})^2 \quad (11)$$

$$f_3'(net_k) = 1 \quad (12)$$

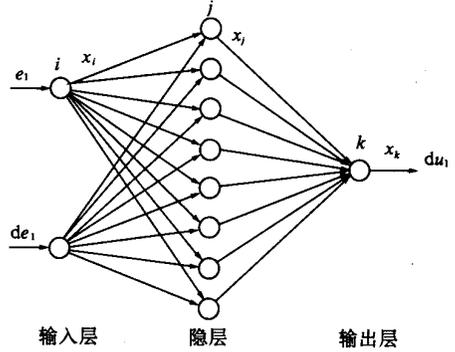


图 4 神经网络结构

Fig. 4 Structure of neural network

网络训练指标为误差平方平均值, 目标函数为

$$E(n) = \frac{1}{2} \cdot \frac{1}{M} \sum_{m=1}^M [e(m)]^2 = \frac{1}{2} \frac{1}{M} \sum_{m=1}^M [du_k(m) - du_p(m)]^2 \quad (13)$$

式中: n 为训练次数; M 为样本总数; e 为误差; du_k 为网络输出目标值 (模糊控制器输出值), du_p 为网络输出实际值. 训练方法为加入动量项的反向传播算法, 调整权值和阈值的表达式为

$$w_{ij}(n+1) = w_{ij}(n) - \eta_j(n) \frac{\partial E(n)}{\partial w_{ij}(n)} + \alpha \Delta w_{ij}(n) \quad (14)$$

$$b_k(n+1) = b_k(n) - \eta_k(n) \frac{\partial E(n)}{\partial b_k(n)} + \alpha \Delta b_k(n) \quad (15)$$

$$w_{ji}(n+1) = w_{ji}(n) - \eta_i(n) \frac{\partial E(n)}{\partial w_{ji}(n)} + \alpha \Delta w_{ji}(n) \quad (16)$$

$$b_j(n+1) = b_j(n) - \eta_j(n) \frac{\partial E(n)}{\partial b_j(n)} + \alpha \Delta b_j(n) \quad (17)$$

式中:

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = -\frac{1}{M} \sum_{m=1}^M \hat{q}_k(m) x_j(m) =$$

$$-\frac{1}{M} \{f_3[net_k(m)] e(m) x_j(m)\};$$

$$\frac{\partial E(n)}{\partial b_k(n)} = -\frac{1}{M} \sum_{m=1}^M \hat{q}_k(m) =$$

$$-\frac{1}{M} \{f_3[net_k(m)] \cdot e(m)\};$$

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -\frac{1}{M} \sum_{m=1}^M [\hat{y}(m) x_j(m)] =$$

$$-\frac{1}{M} \sum_{m=1}^M \{ f_2[\text{net}_j(m)] \hat{y}_k(m) w_{ij}(n) \cdot x_j(m) \};$$

$$\frac{\partial E(n)}{\partial b_j(n)} = -\frac{1}{M} \sum_{m=1}^M \hat{y}(m) =$$

$$-\frac{1}{M} \sum_{m=1}^M \{ f_2[\text{net}_j(m)] \hat{y}_k(m) w_{ij}(n) \}.$$

为加速训练过程,利用delta-bar-delta规则对学习率 η 进行自适应调整^[3]:

$$\Delta \eta(n) = \begin{cases} K, & S(n-1)D(n) > 0 \\ -\beta \eta(n), & S(n-1)D(n) < 0 \\ 0, & \text{其它} \end{cases} \quad (18)$$

$$\eta(n+1) = \eta(n) + \Delta \eta(n) \quad (19)$$

其中 S 与 D 定义为

$$D(n) = \frac{\partial E(n)}{\partial w(n)} \text{ 或 } D(n) = \frac{\partial E(n)}{\partial b(n)};$$

$$S(n) = (1 - \zeta) D(n) + \zeta S(n-1),$$

式中: K, β, ζ 为正的常数.

用模糊控制系统正、负阶跃响应时前 150 个采样点模糊控制器的输入输出数据对神经网络进行训练.训练前对输入数据作归一化处理,以避免隐层神经元工作在传递函数的平坦区域.相关参数为: $\alpha=0.75, K=0.005, \beta=0.2, \zeta=0.3, \eta_k(0)=0.9, \eta_b(0)=0.75$, 权值和阈值的初值取绝对值小于 1 的随机数.经过 6 491 次训练, $E < 10^{-3}$. 用训练好的神经网络代替原模糊推理系统,并对量化因子 k_e, k_c 进行相应调整,可得到控制系统的阶跃响应曲线如图 5 所示.阶跃响应曲线表明,神经网络很好地实现了对控制对象的控制,控制效果与原模糊控制系统几乎一样.

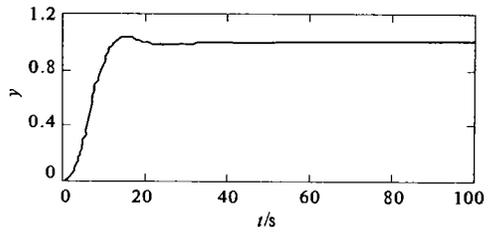


图 5 神经网络控制的阶跃响应曲线

Fig. 5 Step responses for neural network control

2 控制器的离线自学习

当对象模型变为

$$G(s) = \frac{16}{5s^2 + 18s + 1} e^{-0.02s} \quad (20)$$

时,模糊控制系统对长度为 200s 的方波信号的响应曲线如图 6 曲线①所示.响应速度减小,超调量增大,调节时间延长.

现在利用反向传播算法使神经网络控制器进行离线自学习.离线学习时,对量化因子 k_e, k_c 和比例因子 k_u 也采用反传算法进行调整,系统结构图如图 7 所示.

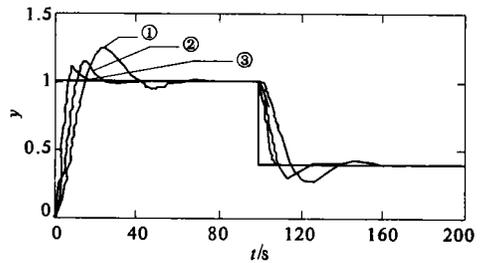


图 6 离线自学习响应曲线

Fig. 6 Responses for off-line self-learning

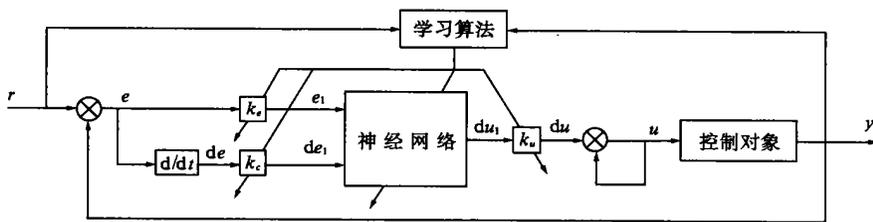


图 7 基于神经网络的自学习模糊控制系统

Fig. 7 Self-learning fuzzy control system based on neural network

目标函数为

$$E(n) = \frac{1}{2} \frac{1}{N} \sum_N \chi(t) [e(t)^2 + \lambda u(t-1)^2] =$$

$$\frac{1}{2} \frac{1}{N} \sum_N \chi(t) \{ [r(t) - y(t)]^2 + \lambda u(t-1)^2 \} \quad (21)$$

式中: n 为学习次数; N 为每次学习的采样点总数; e 为控制系统输入输出间的误差; du 为控制

量的增量, $t-1, t$ 为相邻两次采样时刻.

du 的平方项是对控制增量进行约束,防止响应曲线的振荡, λ 为该项的加权系数.

$\chi(t)$ 为可变加权系数. e 较小时, $\chi(t)$ 较大; e 较大时, $\chi(t)$ 较小,以防止在采样点总数较少情况下,小误差时的数据被大误差时的数据淹没而造成系统响应虽快,但稳态误差大或稳态时响应

曲线振荡.

学习算法为

$$ku(n+1) = ku(n) - \eta_{ku} \frac{\partial E(n)}{\partial ku(n)} + \alpha \Delta ku(n) \quad (22)$$

$$wj(n+1) = w_{ij}(n) - \eta_{ij}(n) \frac{\partial E(n)}{\partial w_{ij}(n)} + \alpha \Delta w_{ij}(n) \quad (23)$$

$$b_k(n+1) = b_k(n) - \eta_k(n) \frac{\partial E(n)}{\partial b_k(n)} + \alpha \Delta b_k(n) \quad (24)$$

$$w_{ji}(n+1) = w_{ji}(n) - \eta_{ji}(n) \frac{\partial E(n)}{\partial w_{ji}(n)} + \alpha \Delta w_{ji}(n) \quad (25)$$

$$b_j(n+1) = b_j(n) - \eta_j(n) \frac{\partial E(n)}{\partial b_j(n)} + \alpha \Delta b_j(n) \quad (26)$$

$$k_e(n+1) = k_e(n) - \eta_{ke} \frac{\partial E(n)}{\partial k_e(n)} + \alpha \Delta k_e(n) \quad (27)$$

$$k_c(n+1) = k_c(n) - \eta_{kc} \frac{\partial E(n)}{\partial k_c(n)} + \alpha \Delta k_c(n) \quad (28)$$

由于

$$\frac{\partial \{ \chi(t) [e(t)^2 + \lambda u(t-1)^2] \}}{\partial ku(n)} = \chi(t) \frac{\partial [r(t) - y(t)]^2}{\partial y(t)} \cdot \frac{\partial y(t)}{\partial u(t-1)} \cdot \frac{\partial u(t-1)}{\partial ku(n)} \quad (29)$$

$$\chi(t) \gamma \frac{\partial [du(t-1)]^2}{\partial u(t-1)} \frac{\partial u(t-1)}{\partial ku(n)} = -2\chi(t) [r(t) - y(t)] \frac{y(t) - y(t-1)}{du(t-1)} du(t-1) + 2\chi(t) \lambda u(t-1) du(t-1) = -2\tilde{u}(t-1) du(t-1),$$

其中:

$$\tilde{u}(t-1) = \chi(t) [r(t) - y(t)] \frac{y(t) - y(t-1)}{du(t-1)} - \chi(t) \lambda u(t-1),$$

所以

$$\frac{\partial E(n)}{\partial k_u(n)} = \frac{1}{2N} \cdot$$

$$\sum_N \frac{\partial \{ \chi(t) e(t)^2 + \chi(t) \lambda u(t-1)^2 \}}{\partial k_u(n)} = -\frac{1}{N} \sum_N \tilde{u}(t-1) du(t-1) \quad (30)$$

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = -\frac{1}{N} \sum_N \tilde{q}_k(t-1) x_j(t-1) \quad (31)$$

$$\frac{\partial E(n)}{\partial b_k(n)} = -\frac{1}{N} = -\frac{1}{N} \sum_N \tilde{q}_k(t-1) \quad (32)$$

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -\frac{1}{N} \cdot \sum_N \tilde{q}(t-1) x_i(t-1) \quad (33)$$

$$\frac{\partial E(n)}{\partial b_j(n)} = -\frac{1}{N} \cdot \sum_N \tilde{q}(t-1) \quad (34)$$

$$\frac{\partial E(n)}{\partial k_e(n)} = -\frac{1}{N} \sum_N \tilde{q}(t-1)]_{i=1e}(t-1) \quad (35)$$

$$\frac{\partial E(n)}{\partial k_c(n)} = -\frac{1}{N} \sum_N \tilde{q}(t-1)]_{i=1e}(t-1) \quad (36)$$

其中:

$$\tilde{q}_k(t-1) = f_3[\text{net}_k(t-1)] \tilde{q}(t-1) k_u(n);$$

$$\tilde{q}(t-1) = f_2[\text{net}_j(t-1)] \tilde{q}(t-1) w_{ij}(n);$$

$$[\tilde{q}_k(t-1)]_{i=1} = \{ f_1[\text{net}_i(t-1)] \sum_{j=1}^8 \tilde{q}(t-1) w_{ji}(n) \}_{i=1};$$

$$[\tilde{q}_k(t-1)]_{i=2} = \{ f_1[\text{net}_i(t-1)] \sum_{j=1}^8 \tilde{q}(t-1) w_{ji}(n) \}_{i=2}$$

离线学习时仍利用 $\bar{\text{delta}}-\text{delta}$ 规则对学习率进行自适应调整.

相关参数为: $\gamma=5, \lambda(t) = |de(t)| / |r(t) - y(t)|, \alpha=0.75, K=0.00001, \beta=0.2, \zeta=0.3, \eta_{ku}(0) = 0.0001, \eta_k(0) = 0.0001, \eta_{ij}(0) = 0.0001, \eta_{ke}(0) = 0.0003, \eta_{kc}(0) = 0.0003$. 比例因子、权值、阈值、量化因子的初值取 1.2 中的训练结果.

则经过 10 次、100 次学习, 基于神经网络的控制系统的响应曲线分别为图 6 的曲线②、③. 这两条曲线表明, 自学习使控制系统适应了模型的变化, 响应速度加快, 超调量和调节时间减小.

3 控制器的在线自学习

上述离线学习的方法可以改善控制系统的响应特性. 事实上, 不采用离线学习, 而采用直接让控制器边运行边学习的在线自学习方法也能够使控制器在运行过程中逐步改善控制性能. 只需把离线学习算法中的 n 换成 t 并使 $N=1$ 就可得到在线学习算法.

在线学习时控制系统的输入信号取周期是 200s 的方波信号.

神经网络的在线调整采用一步反向学习算法, 在每个控制周期实时修正神经网络权值、阈值以及量化因子 k_e 和 k_c 、比例因子 k_u . 由于采用一步算法, 学习率自适应调整的 $\bar{\text{delta}}-\text{delta}$ 规则不再适用.

相关参数为: $\gamma=9, \lambda(t) = 1, \alpha=0.1, \eta_{ku}(0) = 0.00001, \eta_k(0) = 0.00001, \eta_{ij}(0) = 0.00001, \eta_{ke}(0) = 0.00003, \eta_{kc}(0) = 0.00003$. 比例因子、权值、阈值、量化因子的初值取 1.2 中的训练结果.

图 8 为在线自学习控制与模糊控制的比较, 其中曲线①是原模糊控制系统的响应曲线, 曲线

②是在线自学习控制的响应曲线.

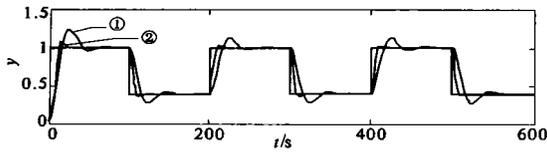


图8 基于神经网络的在线自学习控制与模糊控制的响应曲线

Fig.8 Comparison of responses of self-learning control based on neural net work and fuzzy control

4 结论及讨论

以上内容说明,用神经网络记忆模糊控制器运行时输入输出数据的映射关系不仅可以实现与模糊控制相同的控制性能,而且可以利用神经网络的学习能力通过离线或在线自学习使控制器适应过程的变化,改善控制性能.

在大量的仿真中我们也看到,神经网络的规模、结构、初值、训练学习参数以及目标函数的形

式和参数对训练结果和控制效果有很大影响;同样,过程模型的结构、参数和特性对神经网络的选择和训练学习也有很大影响.在没有成熟的方法可供选择的情况下,只能依靠经验和仿真试验来解决这些问题.对这些问题进行深入研究应是神经网络应用研究的重要内容.

参考文献:

- [1] 冯冬青,谢宋和.模糊智能控制[M].北京:化学工业出版社,1998.
- [2] 李士勇.模糊控制、神经控制和智能控制[M].哈尔滨:哈尔滨工业大学出版社,1998.
- [3] 王耀南.模糊神经网络在炉温控制中的应用[J].系统工程与电子技术,1997,19(12):72~77.
- [4] 冯冬青,李小雄.基于DSP的轧机调速系统模糊神经控制[J].郑州大学学报(工学版),2003,24(3):17~20.
- [5] 阎平凡,张长水.人工神经网络与模拟进化计算[M].北京:清华大学出版社,2000.

Self-learning Fuzzy Control Based on Neural Network

FENG Dong-qing, ZHANG Xi-ping

(College of Electrical Engineering, Zhengzhou University, Zhengzhou 450002, China)

Abstract: This paper puts forward a kind of self learning fuzzy control method based on neural network by means of the combination of fuzzy control and self learning ability of neural network offering algorithms for network training, controller off line and on line self learning. With this method, the controller can realize off line and on line self learning to improve its control performance when the controlled process varies. Simulation results indicate the effectiveness of this method.

Key words: fuzzy control; neural network; self learning; simulation