

文章编号: 1001-0920(2001)03-347-04

动态模糊神经网络控制器在伺服系统中的应用

柳朝军, 廖晓钟, 张宇河
(北京理工大学 自动控制系, 北京 100081)

摘要: 通过在 ANFIS 的归一化层与输出层之间加入递归层, 提出一种新型的动态模糊神经网络 (DFNN), 将模糊推理系统、神经网络和 III 型控制有机地结合起来。给出了 DFNN 的网络结构, 为基于收缩间距隶属函数和 BP 算法提供了参数调整方法。系统实验表明, DFNN 控制器比 PID+ 前馈控制具有更好的动、静态响应, 尤其在前馈信号难以取得的情况下具有更明显的优势。

关键词: 模糊神经网络; 学习算法; 复合控制

中图分类号: TP 18 文献标识码: A

Application in Servo System of Dynamical Neuro-fuzzy Network Controller

LIU Chao-jun, LIAO Xiao-zhong, ZHANG Yu-he

(Department of Automatic Control, Beijing Institute of Technology, Beijing 100081, China)

Abstract: A novel dynamical neuro-fuzzy network (DFNN) is proposed by adding a recurrent layer between the normalized layer and output layer of the forward neuro-fuzzy network ANFIS. DFNN combines the advantages of fuzzy system, neural network and type-III controller. The structure of DFNN and a parameter-regulating method which is based on the shrinking-span membership functions and BP algorithm are proposed. The experiment results show that DFNN has a better response than the traditional PID+ forward controller especially in the situation when the forward signal is difficult to obtain.

Key words: neuro-fuzzy network; learning algorithm; combined control

1 引言

在传统的伺服系统中, 前馈加 PID 反馈的复合控制器已得到广泛的应用。然而, 在某些特殊的控制系统中(如视频跟踪系统), 伺服系统的输入为偏差量, 前馈信号不易得到, 系统的精度难以提高。近年来, 模糊神经网络已成为研究的热点。因此, 构造同时具有模糊逻辑、神经网络和 PID 优点的模糊神经网络 PID 控制器, 具有非常实用的价值。

文献[1]构造出基于自适应网络的模糊推理系统(ANFIS)。本文则在 ANFIS 的归一化层与输出层之间加入局部递归层, 由于局部递归元的动态效果, 构成了动态模糊神经网络(DFNN)。在递归层中使用两个局部递归元, 并将其与误差及误差的变化为小和零(模糊语言变量)时的规则相关联。当误差较大时, DFNN 仍采用 ANFIS 形式实现 PD 控制; 当误差及误差的变化为小和零(模糊语言变量)时,

收稿日期: 2000-03-16; 修回日期: 2000-08-08

基金项目: 国家“九五”重点预研项目(34.2.1)

作者简介: 柳朝军(1968—), 男, 河北石家庄人, 博士生, 从事模糊系统和神经网络等研究; 张宇河(1940—), 男, 福建漳州人, 教授, 博士生导师, 从事机器人和运动控制等研究。

由于局部递归元的二次积分作用而实现 型控制。实际伺服系统的实验结果表明,DFNN 比复合控制具有更好的动、静态响应。

2 动态模糊神经网络结构

动态模糊神经网络结构如图 1 所示,它由 6 部分组成:输入层(L1),模糊化层(L2),规则层(L3),归一化层(L4),递归层(L5)和输出层(L6)。前 4 层做输入空间的模糊划分,递归层内含有两个回归神经元,并与误差及误差的变化为小和零的规则输出相连。

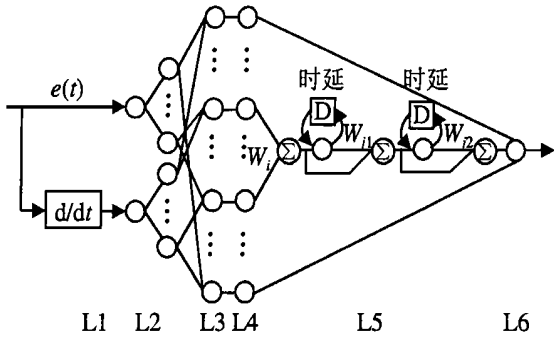


图 1 DFNN 的结构

1) 输入层 L1: 输入层将输入向量 $X(t) = [x_1(t) \ x_2(t) \ \dots \ x_n(t)]^T$ 传到下一层,将第 i 个神经元连接到第 i 个 X 变量。其中, $t_0 < t < t_f, i = 1, 2, \dots, n, n$ 为输入变量个数。

2) 模糊化层 L2: 将 $n \times N_i$ 个神经元分成 n 组,每个神经元只与一个输入变量相连。输入连接权是每个隶属函数的中心和宽度 $\{c_{ij}, \sigma_{ij}\}$, 采用三角型隶属函数,输出隶属度

$$\mu_{ij}(t) = 1 - \left| \frac{x(t) - c_{ij}}{\sigma_{ij}} \right| \quad (1)$$

其中, $i = 1, 2, \dots, n, j = 1, 2, \dots, N_i$ 。每个神经元对应一个模糊子集或语言变量,如 NL, NM, NS, NZ, PZ, PS, PM 和 PL 等。

3) 规则层 L3: 计算模糊子集的隶属度及每条规则的激活度 $a_s(t)$, 每个神经元与 n 个输入变量相连, 每个输入连接权为 1。神经元完成模糊与操作, 以获得相应的激活度

$$a_s(t) = \mu_{i1}(t) \dots \mu_{kl}(t) \quad (2)$$

其中, $s = 1, 2, \dots, N_A; i, k = 1, 2, \dots, n (i \neq k); j = 1, 2, \dots, N_j; l = 1, 2, \dots, N_l; N_A = \prod_{i=1}^n N_i$ 。

4) 归一化层 L4: 对规则层的激活度输出做归

一化处理,即

$$\overline{a_s(t)} = \frac{a_s(t)}{\sum_{l=1}^{N_A} a_l(t)}, \quad s = 1, 2, \dots, N_A \quad (3)$$

至此,便实现了输入空间的模糊划分和模糊推理。值得注意的是:对任意输入向量 $X(t)$, 任意时刻只有部分模糊输入子空间被激活,非零的规则激活度只在相应的激活子空间存在。

5) 递归层 L5: 递归层结构如图 1 所示。它包含两个具有内反馈的回归神经元和两个线性求和神经元,仅与误差及误差的变化为小和零的 12 条规则输出相连,其余规则输出则直接与输出层 L6 相连。反馈层输入为

$$I(t) = \sum_{i=s-p}^{p+k} \overline{a_s(t)} W_i$$

其中, p 为误差及误差的变化为小和零的第 1 条规则, $k = 12$ 为误差及误差的变化为小和零的规则总数。由文献[2]可知,当采用三角形隶属函数时,product-sum 型模糊控制器表现为 PD 控制器,反馈层输入

$$I(t) = A + Pe(t) + De(t) \quad (4)$$

其中 A, P 和 D 为时变参数。

第 1 个求和神经元输出

$$\begin{aligned} X(t) &= I(t) + W_{j1} \int I(t) dt = \\ &= (A + W_{j1}A t) + (P + W_{j1}D) e(t) + \\ &= De(t) + W_{j1}P \int e(t) dt \end{aligned} \quad (5)$$

第 2 个线性求和神经元输出

$$\begin{aligned} O(t) &= X(t) + W_{j2} \int X(t) dt = \\ &= \left[A + W_{j1}A t + W_{j2}A t + \frac{1}{2} W_{j1}W_{j2}A t^2 \right] + \\ &= (P + W_{j1}D + W_{j2}D) e(t) + De(t) + \\ &= (W_{j1}P + W_{j2}P + W_{j1}W_{j2}D) \int e(t) dt + \\ &= W_{j1}W_{j2}P \int \int e(t) dt \end{aligned} \quad (6)$$

由于递归层输出 $O(t)$ 为误差 $e(t)$ 的二次积分函数,因此系统在误差及误差的变化为小和零时具有型控制的特点;而当误差较大时,激活的规则仍采用 ANFIS 形式后件输出形式。此时模糊神经网络表现为 PD 控制器,从而用模糊神经网络实现了 PD+型控制。

6) 输出层 L6: 以输出层神经元对递归层的输出求和,作为模糊神经网络的输出

$$u(t) = O(t) + \sum_{i=s=1}^{N_A^*} W_i \overline{a_s(t)} \quad (7)$$

其中 N_A^* 为非零规则的总数, $N_A^* = N_A - k$, k 是误差为零的规则总数。

3 学习算法

3.1 收缩间距隶属函数

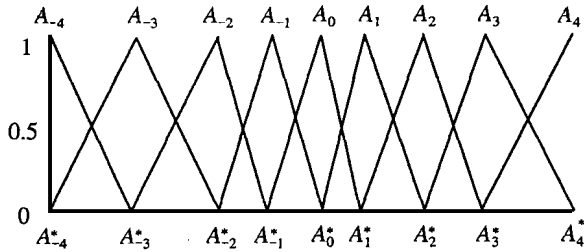


图 2 收缩间距隶属函数

收缩间距隶属函数如图 2 所示。其中, A_l 为语言变量, A_l^* 为隶属函数的中心, $l = -4, \dots, 0, \dots, 4$ 。假设有 M 个输入变量和一个输出变量, 每个输入变量有 $2m_i + 1$ 个语言值 $A_l(x_i)$ ($l \in l_{m_i} = \{0, \pm 1, \dots, \pm m_i\}$, $i = 1 \sim M$) 和一个收缩因子 s_i ($i = 1 \sim M$)。收缩间距的三角隶属函数为

$$\mu_{A_l}(x_i) = \begin{cases} 0, & x_i < A_{l-1}^* \\ \frac{x_i - A_{l-1}^*}{A_l^* - A_{l-1}^*}, & A_{l-1}^* \leq x_i < A_l^* \\ \frac{A_{l+1}^* - x_i}{A_{l+1}^* - A_l^*}, & A_l^* \leq x_i < A_{l+1}^* \\ 0, & x_i \geq A_{l+1}^* \end{cases} \quad (8)$$

隶属函数的中心定义为

$$A_l^* = \frac{L}{m_i} s_i^{m_i - |l|} \quad (9)$$

式中 $l \in l_{m_i}$, $s_i \in [0, 1]$ 为语言变量 x_i 的收缩因子。当 $s_i = 1$ 时, 收缩间距隶属函数 (SSMFs) 便等效为等间距隶属函数。由于输入变量的个数和每个变量的语言值数可事先确定, 可调因子只有输入变量的收缩因子 s_i , 这便大大降低了学习强度。

3.2 网络后件参数学习

由于积分系数 W_{j1} 和 W_{j2} 在等速和等加速跟踪时起主要作用, 而在阶跃信号时则希望其影响很小, 因此令 $W_{j1} = W_{j2} = 0$, 在阶跃状态下训练权值 W_i 。

如果 $r(t)$ 为控制系统的参考输入, $y(t)$ 为系统的实际输出, 则系统的误差定义为 $e(t) = r(t) - y(t)$, t_0 和 t_f 为系统运行的初始和终止时间。如果 T_s 为采样时间, 则采样个数为 $N_f = (t_f -$

$t_0) / T_s$ 。

训练目标性能指标为

$$J = \frac{1}{N_f} \sum_{k=1}^{N_f} (r(k) - y(k))^2 \quad (10)$$

由梯度法得

$$\frac{\partial J}{\partial y(k)} = - \frac{2}{N_f} (r(k) - y(k)) \quad (11)$$

由于模糊神经网络被用作直接自适应控制器, 无法利用对象的 Jacobian 阵信息, 因此本文利用对象的定性信息, 即用 Jacobian 阵符号代替 Jacobian 阵进行反传权系数修正, 即

$$\frac{\partial y(k)}{\partial u(k)} \text{sign} \left[\frac{y(k) - y(k-1)}{u(k) - u(k-1)} \right] \quad (12)$$

由式 (7) 得

$$\frac{\partial u(k)}{\partial W_i(k)} = \frac{\partial u(k)}{\partial O(k)} \frac{\partial O(k)}{\partial W_i(k)} = \overline{a_s(k)} \quad (13)$$

由式 (11) ~ (13) 得

$$\frac{\partial J}{\partial W_i(k)} = \frac{\partial J}{\partial y(k)} \frac{\partial y(k)}{\partial u(k)} \frac{\partial u(k)}{\partial W_i(k)} = - \frac{2}{N_f} (r(k) - y(k)) \text{sign} \left[\frac{y(k) - y(k-1)}{u(k) - u(k-1)} \right] \overline{a_s(k)} \quad (14)$$

因此

$$W_i(k+1) = W_i(k) + \eta \frac{\partial J}{\partial W_i(k)} \quad (15)$$

其中, $0 < \eta < 1$ 为学习率, $k = 0, 1, \dots$ 为学习步数。

当 W_i 整定后, 积分系数 W_{j1} 和 W_{j2} 通过简单的手工选择便可得到满意的参数值, 获得理想的阶跃、等速及正弦响应。

如果 $W_{j1} = W_{j2} = 0$, 则动态模糊神经网络简化为 ANFIS。由此可见, 动态模糊神经网络是 ANFIS 的扩展。另外, 由于模糊神经网络的参数 W_i , W_{j1} 和 W_{j2} 均有明确的物理意义, 因此其初始值可按专家经验进行选择, 而不象神经网络的权值需随机确定。这不仅加快了网络的收敛速度, 而且具有将符号主义和连接主义相互融合的优点。

4 实验结果

实验系统原理如图 3 所示。

电机输出功率 0.4 kW, 额定转矩 2.9 N·m, 最大转矩 26N·m, 最高转速 2 000 rpm, 转动惯量 0.002 2 kgm², 机械时间常数 0.013 s, 电气时间常数 0.007 s; 光电码盘 2 000 脉冲 / 转, 减速比 69.5, 负载转动惯量 27 kg·m², 采样周期 0.001 s。

DFNN 控制参数: 收缩因子 $s_1 = 0.37, s_2 = 1,$

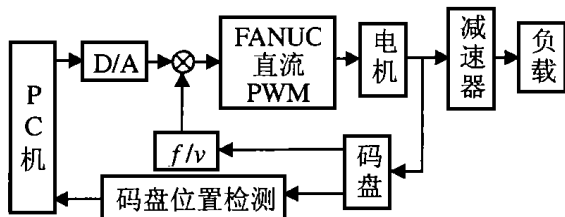


图3 实验系统原理

$W_{j1} = 0.003, W_{j2} = 0.01, W_i$ 按上述BP学习算法进行训练,其中学习率 $\eta = 0.02$ 。

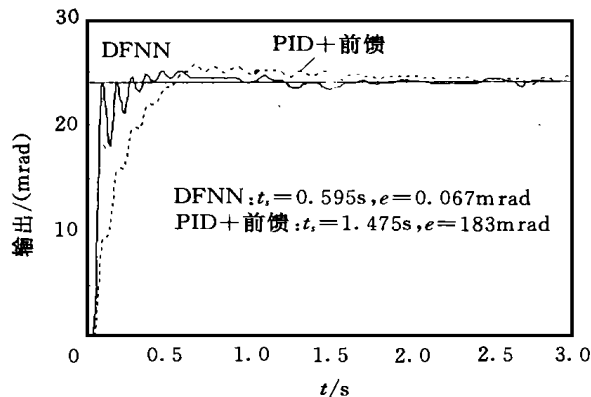


图4 阶跃输入两种控制效果比较

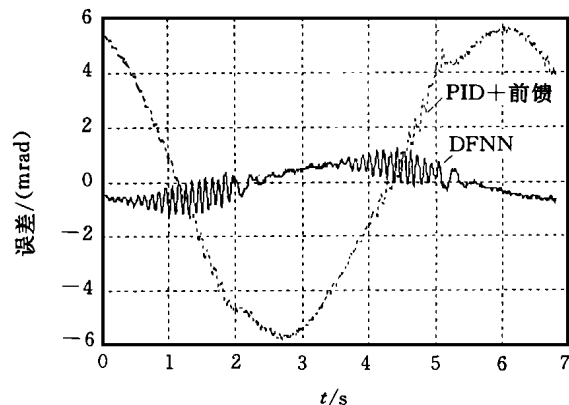


图5 正弦输入两种控制误差比较

24mrad的阶跃响应如图4实线所示,跟踪速度 $60^\circ/s$, 加速度 $60^\circ/s^2$ 的等效正弦振幅为 60° 周期为6.28s,DFNN的跟踪误差如图5实线所示,最大误

差为1.283mrad。

PID + 前馈控制参数: $K_P = 1.875, K_I = 1/256, K_D = 1.875$, 前馈系数 $K_q = -32.9.24\text{mrad}$ 的阶跃响应如图4虚线所示; 正弦信号跟踪误差如图5虚线所示, 最大误差为5.676 mrad。

实验结果表明, 阶跃输入时, DFNN 具有更快的响应速度; 正弦信号跟踪时, DFNN 明显地减小了系统误差, 提高了系统的控制精度。

5 结 论

本文提出一种新型的动态模糊神经网络。DFNN 将模糊逻辑、神经网络和 型控制器融为一体, 具有如下优点:

- 1) 输入空间被划分为一组模糊子空间, 每个模糊子空间为一个动态子系统, 可表达更复杂的动态行为;
- 2) 不需要对被控对象先验知识的了解, 适合于复杂对象和非精确建模对象的控制;
- 3) 网络参数具有明确的物理意义, 初始权值的选择可按经验整定, 大大加快了收敛速度;
- 4) 网络表现为 PD+ 型控制器, 由于隶属函数覆盖整个输入空间, PD 逐渐切换到 型控制上, 因而避免了固定点切换带来的抖动现象;
- 5) 不需要前馈信号, 既具有快速平稳的过渡过程响应, 又能提高系统的跟踪精度, 较好地解决了稳定性与准确性之间的矛盾。

参考文献:

- [1] J S R Jang. ANFIS: A adaptive-network-based fuzzy inference system[J]. IEEE Trans on System, Man and Cybernetics, 1993, 23(3): 665-685.
- [2] Wu Zhiqiao, M Mizumoto. PID type fuzzy controller and parameters adaptive method[J]. Fuzzy Sets and Systems, 1996, 78(1): 23-35.
- [3] 孙增圻, 张再兴, 邓志东. 智能控制理论与技术[M]. 北京: 清华大学出版社, 1997.