

文章编号: 1002-0411(2000)01-070-06

# 采用遗传算法训练对角递归神经网络预测控制器

张阿卜 黄伟斌

(厦门大学自动化系 厦门 361005)

**摘要:** 本文提出了一种基于广义预测控制的神经网络预测控制方案。预测控制器由对角递归神经网络预测控制器和前向神经网络静态补偿器组成。两种神经网络均采用遗传算法进行训练。仿真实验表明,对于带纯时延的非线性被控对象,采用遗传算法设计的对角递归神经网络预测控制器具有令人满意的控制性能。

**关键词:** 广义预测控制, 神经网络, 遗传算法

**中图分类号:** TP13

**文献标识码:** B

## 1 引言

广义预测控制(GPC)具有优良的控制性能,在工业过程控制中已经得到广泛应用<sup>[1]</sup>。传统的广义预测控制算法只适用于线性被控对象。本文基于 GPC 的思想,采用对角递归神经网络作为控制系统中的预测器和预测控制器。对角递归神经网络具有全递归神经网络的动态特性,同时只需较少的神经元和权值个数,较适合于动态系统的实时控制<sup>[2]</sup>,控制系统中静态补偿器则采用前向神经网络。神经网络权值的训练可以采用 BP 算法,但算法的训练时间往往较长,在训练预测控制器时需要计算目标函数的梯度,当预测长度比较大时,计算梯度的工作量显著增加,而且容易陷入局部极小值。本文则采用了具有全局最优搜索能力的遗传算法<sup>[3]</sup>对其进行训练。遗传算法在训练神经网络权值时具有速度快、不需求目标函数的梯度、较不易陷入局部极小值等特点。所设计的神经网络预测器既适用于线性被控对象,也适用于非线性被控对象。

## 2 基于 GPC 的神经网络预测控制系统

### 2.1 对角递归神经网络(DRNN)

对角递归神经网络的结构如图 1。它反映的输入-输出关系为

$$O(k) = \sum_j W_j^o X_j(k) \quad (1)$$

$$X_j(k) = F(S_j(k)) \quad (2)$$

$$S_j(k) = W_j^p X_j(k-1) + \sum_i W_{ij}^l I_i(k) \quad (3)$$

其中  $I_i(k)$  是网络的输入,  $S_j(k)$  是隐含层第  $j$  个节点的输入,  $X_j(k)$  是隐含层第  $j$  个节点的输出,  $O(k)$  是网络的输出, 可取  $F(x) = 1/(1+e^{-x})$ 。

设 DRNN 的输入为  $u(k)$ ,  $y(k)$ , 则其输出为

$$O(k) = h(u(l), y(l); l = k) \quad (4)$$

其中  $h$  为非线性函数,  $k, l$  为非负整数

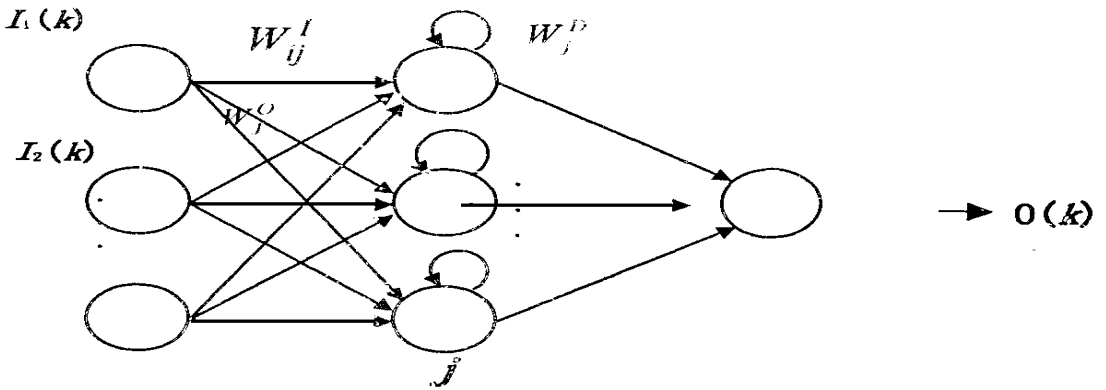


图 1 对角递归神经网络结构

### 2.2 控制系统的组成

设被控对象为

$$y(k+1) = f(y(k), y(k-1), \dots, y(k-n_y), u(k), u(k-1), \dots, u(k-n_u)) \quad (5)$$

系统的给定值为  $r$ , 则参考轨迹为

$$y_r(k+1) = \alpha y_r(k) + (1-\alpha)r, y_r(0) = y(0) \quad (6)$$

其中  $0 < \alpha < 1$

基于 GPC 的策略, 设预测长度为  $N$ , 控制长度为 1, 控制系统的性能指标为

$$J_c = \frac{1}{2N} \sum_{j=1}^N (y_r(k+j) - y(k+j))^2 + \lambda u^2(k) \quad (7)$$

其中  $y(k+j)$  为输出的预测值,  $\lambda$  为控制量的加权系数 由于取控制长度为 1, 令  $u(k+j) = u(k), j = 1, 2, \dots, N$ , 并由下式对输出进行预测:

$$y(k+j) = f(y(k+j-1), \dots, y(k+j-n_y-1), u(k+j-1), \dots, u(k+j-n_u-1)), j = 1, 2, \dots, N \quad (8)$$

对性能指标进行优化计算, 使  $J_c$  最小, 可得  $u(k)$ . 由式(5)、(7)、(8)可以看出预测控制器可表示为

$$u(k) = g(y_r(k+1), \dots, y_r(k+N), y(k), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)) \quad (9)$$

根据式(9), 同时为了消除系统的静态误差, 本文采用的控制器由两部分组成: 预测控制器和静态补偿器, 它们的输出分别为  $u_b$  和  $u_f$ , 这时有  $u = u_b + u_f$ . 预测控制器和静态补偿器分别表示为

$$u_b = g_b(y_r(k+1), \dots, y_r(k+N), y(k), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)) \quad (10)$$

$$u_f = g_f(r) \quad (11)$$

而  $u_f$  满足被控对象稳态时的输入输出关系:

$$y_s = f(r, r, \dots, r, u_f, u_f, \dots, u_f) \quad (12)$$

其中  $u_f$  和  $y_s$  分别为被控对象稳态时的输入和输出

今采用 DRNN 构成辨识器、预测器、预测控制器, 根据 DRNN 的性质(式(4)), 有如下关

$$\text{辨识器 (DRN I)} \quad \hat{y}(k+1) = f(\hat{y}(k), u(k), W) \quad (13)$$

$$\begin{aligned} \text{预测器 (DRN P)} \quad y_p(k+j) &= f(y_p(k+j-1), u(k), \hat{W}), \\ j &= 2, 3, \dots, N; \quad y_p(k+1) = y(k+1) \end{aligned} \quad (14)$$

$$\text{预测控制器 (DRN PC)} \quad u_b(k) = g_b(y_r(k+1), \dots, (y_r(k+N), y(k), u(k-1), V_1). \quad (15)$$

静态补偿器采用前向神经网络构成:

$$\text{静态补偿器 (FNC)} \quad u_f = g_f(r, V_2) \quad (16)$$

式中  $W, V_1, V_2$  是网络的权值

由此可知, 预测器和辨识器具有相同的结构. 神经网络DRNC的输入为  $y_r(k+1), \dots, y_r(k+N), y(k), u(k-1)$ , 输入节点数为  $I_s = N + 2$  个.

整个系统的结构图为:

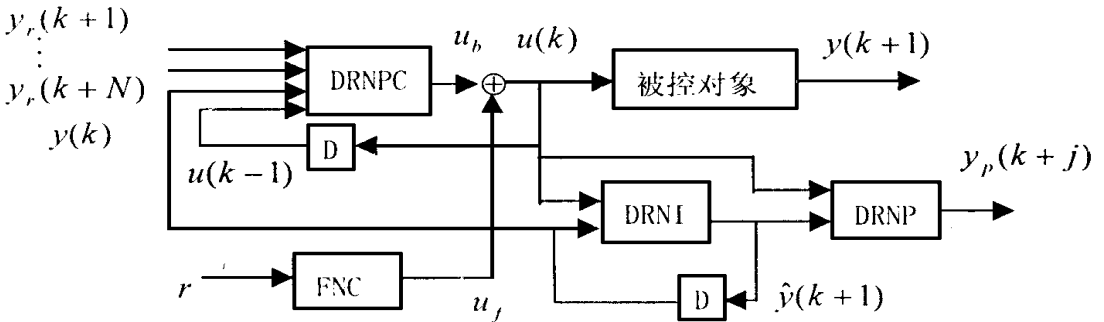


图2 神经网络控制系统方框图

图2中,  $D$  表示时间滞后一拍, 辨识器DRNI和预测器DRNP 仅在训练DRNPC和FNC时使用

### 3 采用遗传算法训练神经网络预测控制器

遗传算法 (Genetic Algorithm 以下简称 GA) 来源于生物学中达尔文的进化论 它是建立在自然选择和种群遗传的基础上, 模拟自然界“物竞天择, 适者生存”的进化过程, 在问题空间进行全面并行的, 随机的搜索优化, 使种群朝全局最优的方向收敛

#### 3.1 基于遗传算法的网络权值编码和解码

采用遗传算法对神经网络的权值进行训练, 首先必须对权值进行相应的编码, 同样在求每个个体的适应值时, 必须将每个个体转化成相应的网络权值, 计算出网络的输出, 并通过控制系统的性能指标对其进行评估, 取得相应的适应值 这里我们采用简单的编码方式, 即将网络的所有权值按一定的顺序排列在一起, 每个权值相应的是一定长度 (不妨设为  $l$ ) 的二进制数, 另设构成控制器的神经网络的输入节点数为  $I_s$ , 中间节点数为  $J_s$ , 输出节点数为 1, 则权值的个数  $m = (I_s + 2) \times J_s$ , 个体的串长为

$$L = (I_s + 2) \times J_s \times l$$

这样对于拥有  $m$  个权值的神经网络的个体的编码就可用  $L$  位的二进制串长表示 第  $t$  代中的第  $i$  个个体  $x_t^i$  可以表示为  $m \times l$  维的行向量, 即  $x_t^i = [x_t^{i(1)} \dots x_t^{i(l+1)} \dots x_t^{i(2l)} \dots x_t^{i(m-l+1)} \dots x_t^{i(m)}]$  第  $k$  个长度为  $l$  的二进制码串就代表着该神经网络的第  $k$  个权的值, 其转化为实数的解码函数  $\Gamma$  为

$$\Gamma(x^i, k) = u_k + \frac{v_k - u_k}{2^l - 1} \left( \prod_{j=1}^l x_i^{i(k+l-j)} \times 2^{j-1} \right)$$

其中  $v_k$  和  $u_k$  是该权值范围的上下限

这样在完成编码和解码的设计之后, 可以根据具体的网络控制器产生一定群体数的串长为  $L$  的二进制数, 然后通过解码函数, 获得每个个体相应的神经网络的全部权值, 组成神经网络。这样, 就可以用遗传算法对该群体进行繁殖、交叉和变异。

### 3.2 基于遗传算法的DRN I网络的权值训练

采用遗传算法对DRN I网络的权值进行训练。其中DRN I网络有两个输入: 一为对象输入区间的任意一个随机数  $u(k)$ , 另一个输入为  $y(k)$ 。利用上述网络权值的编码方式产生一定数量的个体, 每个个体即可形成一个DRN I网络, 通过个体的解码函数获得网络的权值, 计算得出网络输出  $\hat{y}(k+1)$ 。通过遗传算法对网络的最优解进行搜索。具体步骤如下:

1)  $k=0$  时, 随机产生一定群体数的个体, 作为遗传算子的初始群体

2) 取  $u(k), u(k+1), \dots, u(k+N_p-1)$  为任意随机数, 将它们分  $N_p$  次分别作用于被控对象和DRN I, 测量对象的输出  $y(k+j)$  和每个个体所构成网络的输出

$$y(k+j), \quad j = 1, 2, 3, \dots, N_p$$

3) 利用下式计算每个个体所对应的  $J_p$ , 把它的倒数作为个体的初始适应值 (Raw Fitness), 然后进行遗传操作, 即复制、交叉、变异等, 产生下一代的新个体

$$J_p = \frac{1}{N_p} \sum_{j=1}^{N_p} (\hat{y}(k+j) - y(k+j))^2$$

4)  $k = k+1$ , 如果  $k$  小于设定的代数, 返回 2)。

5) 保存搜索后的最优个体

### 3.3 基于遗传算法的前馈神经网络静态补偿器的权值训练

采用遗传算法对静态补偿器的网络权值进行训练。训练时需用已训练过的DRN I。这时DRN I的输入为静态补偿器的输出  $u_f$  和给定值  $r$ , 即这时DRN I的输入输出的关系为  $y = \hat{f}(r, u_f, W)$ 。编码方式与训练DRN I时相同, 具体的训练步骤如下:

1)  $\text{gen} = 0$  时, 随机产生一定群体数的个体, 作为遗传算子的初始群体

2) 随机产生  $N_f$  个给定信号:  $r(i), \quad i = 1, 2, \dots, N_f$ 。

3) 计算相应每个个体所构成的前向神经网络的输出  $u_f(i)$ , 并将  $u_f$  和  $r(i)$  作用于DRN I得到  $\hat{y}(i)$ 。

4) 用下式计算每个个体所对应的  $E$ , 并把它倒数作为个体的适应值, 然后进行遗传操作, 即复制、交叉、变异等, 产生下一代的新个体

$$E = \sum_{i=0}^{N_f} (\hat{y}(i) - r(i))^2$$

5)  $\text{gen} = \text{gen} + 1$ , 如果  $\text{gen}$  小于设定的数, 则返回 2)。

6) 保存搜索后的最优个体

### 3.4 基于遗传算法的神经网络预测控制器的权值训练

采用遗传算法对反馈控制器的网络权值进行训练。编码方式也与训练DRN I时相同, 具体的训练步骤如下:

1)  $\text{gen} = 0$  时, 设定每一代的训练次数  $N_t$ , 随机产生一定群体数的个体, 作为遗传算法的

### 初始群体

2)  $k=0$  时, 随机产生给定信号  $r$ , 计算相应的参考轨迹  $y_r(k+1), y_r(k+2), \dots, y_r(k+N)$ , 以及已训练过 FNC 的输出  $u_f$ .

3) 计算每个个体所构成的网络的输出  $u_b(k)$ , 并计算  $u(k) = u_b(k) + u_f$ .

4) 由训练后的 DRN I 和 DRN P 计算预测输出  $y_p(k+j), j=1, 2, \dots, N$ .

5) 当  $k < N_i$  时,  $k = k+1$ , 产生参考轨迹  $y_r(k+j), j=1, 2, \dots, N$ , 返回 3).

当  $k = N_i$  时, 对每个个体计算

$$E_c = \sum_{k=0}^{N_r} \left( \frac{1}{2N} \sum_{j=1}^N (y_r(k+j) - y_p(k+j))^2 + \lambda u^2(k) \right)$$

并把它倒数作为个体的适应值, 然后进行遗传操作, 即复制、交叉、变异等, 产生下一代的新个体

6)  $\text{gen} = \text{gen} + 1$ , 如果  $\text{gen}$  小于设定的数, 则返回 2)

7) 保存搜索后的最优个体

## 4 仿真实验及结论

仿真对象为带有纯时延的非线性被控对象:

$$y(k+1) = \frac{y(k)y(k-1)[y(k)+2.5]}{1+y^2(k)+y^2(k-1)} + 1.25u(k-2)$$

即  $n_y = 1, n_u = 2$ , 取  $N = 4$  参考模型:  $y_r(k+1) = 0.8y_r(k) + 0.2r(k+1)$ . 其中  $r(k)$ , 是幅度为 5 的正负跳变的方波, 方波的周期为 200

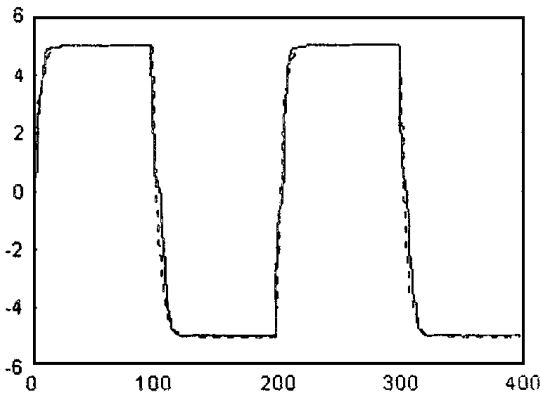


图3 带纯时延的非线性被控对象的仿真结果

先用遗传算法对 DRN I, FNC 和 DRN PC 网络进行训练 其中, 在训练 DRN I 时,  $u(k)$  取  $[-4, 2]$  之间的随机数 在训练 FNC, DRN PC 时, 取  $[-5, 5]$  之间的随机数 DRN I, FNC 和 DRN PC 的网络结构分别取 2-5-1, 1-4-1 和 8-24-1. 个体的群体数取 26, 交叉率取 0.8, 变异率取 0.1, 分别进行训练到 500 代, 保存最优个体 图 3 是 DRN PC 和 FNC 经遗传算法训练后系统的仿真结果

仿真结果表明, 经过遗传算法训练后的神经网络预测控制器对于带纯时延被控对象有很好的性能 本文所研究的是控制长度为 1 的神经网络预测控制器, 但本文采用的 GA 训练神经网络预测控制器的方法不难推广到控制长度

大于 1 的一般的广义预测控制器

## 参 考 文 献

1 Clarke D W, Mohtadi C and Tuffs P S. Generalized Predictive Control Part1 and Part2 Automatica, 1987, 23(2): 137 ~ 160

2 Ku Chao-chee and Lee Kwang Y. Diagonal Recurrent Neural Networks for Dynamic Systems Control IEEE Trans Neur-

ral Networks, 1995, 6(1): 144~ 155

3 Goldberg D E. Genetic Algorithms in Search, Optimization & Machine Learning. Addison Wesley, 1989

## TRAINING OF DIAGONAL RECURRENT NEURAL NETWORK PREDICTIVE CONTROLLER USING GENETIC ALGORITHMS

ZHANG A-bu HUANG Wei-bin

(Dept. of Automation, Xiamen University, Xiamen 361005)

**Abstract** This paper proposes a neural network predictive control scheme based on generalized predictive control (GPC). The predictive controller is made of diagonal recurrent neural network predictive controller (DRNPC) and feed forward neural network steady-state compensator (FNC). Two kinds of neural networks are trained using genetic algorithms. The simulation results show satisfactory performance of the neural network predictive controller for nonlinear plants with dead time.

**Key words:** generalized predictive control, neural network, genetic algorithm

### 作者简介

张阿卜(1946-), 男, 副教授, 研究领域为预测控制, 模糊控制, 神经网络, 遗传算法等。

黄伟斌(1974-), 男, 硕士, 研究领域为神经网络, 遗传算法及其应用等。

(上接第 54 页)

## A HIERARCHICAL FSM METHOD FOR CONTROL FLOW OF MACHINE TOOL

LEI Wei-min<sup>1</sup> YU Dong<sup>2</sup> LI Ben-ren<sup>2</sup> TENG Hong-fei<sup>1</sup>

(1. Dalian University of Technology 116024; 2. Shenyang Institute of Computing Technology, Chinese Academy of Sciences 110003)

**Abstract** The procedure-oriented IEC-1131-3 specification has poor capability to support the open architecture design and object-oriented implementation in representation of the control flow of machine tool. In order to meet the new requirement, we extended the concept of Finite State Machine, and proposed a new hierarchical FSM method for representing the control flow of machine tool. In this paper, the organization, properties and formal definition of hierarchical FSM are discussed in detail. A kind of general structure based on hierarchical FSM representation of machine tool controller is described. Moreover, the problems related to representing openness and system implementation are also discussed.

**Keywords:** machine tool controller, control flow representation, hierarchical FSM, IEC-1131-3 specification

### 作者简介

雷为民(1969-), 男, 助理研究员, 博士生, 研究方向为数控技术、计算机应用、智能控制。