



PID 神经元网络多变量控制系统分析

舒怀林

(广州大学维修工程技术学部 广州 510091)

摘 要 PID 神经元网络是一种新的多层前向神经网络,其隐含层单元分别为比例(P)、积分(I)、微分(D)单元,各层神经元个数、连接方式、连接权初值是按 PID 控制规律的基本原则确定的,它可以用于多变量系统的解耦控制.给出了 PID 神经元网络的结构形式和计算方法,从理论上证明了 PID 神经元网络多变量控制系统的收敛性和稳定性,通过计算机仿真证明了 PID 神经元网络具有良好的自学习和自适应解耦控制性能.

关键词 神经网络控制,多变量系统,PID 控制,稳定性,收敛性.

ANALYSIS OF PID NEURAL NETWORK MULTIVARIABLE CONTROL SYSTEMS

SHU Huailin

(Tech. Faculty of Maintenance Eng., Guangzhou University, Guangzhou 510091)

Abstract PID neural network is a new kind of feedforward multi-layer network. Its hidden layer neurons are proportional neuron (P), integral neuron (I) and derivative (D) neuron. The numbers of the neurons, the connective forms and primary value of the weights are based on the rules of the PID control. The stabilization and convergence of the PID neural network multivariable control system are theoretically proved. Computer simulation displays the perfect self-study and adaptive decoupling control properties of the PID neural network.

Key words neurocontrol, multivariable system, PID control, stabilization, convergence.

1 引言

由于多变量系统内部的耦合作用以及参数的复杂性,使其控制问题成为控制理论和控制工程研究的重点和难点问题.将应用最广泛的 PID 控制器和具有学习功能的神经元

网络相结合,是智能控制研究的一个新的着眼点,已经有不少研究成果^[1-9].但目前的研究仅局限于采用神经网络辅助选取或修改传统 PID 控制器的 P,I,D 参数,且仅局限于在单变量系统的控制方面.

PID 神经网络(PID Neural Network)是将 PID 控制规律融进神经网络之中构成的. PID 神经网络属于多层前向神经网络的范畴,但它与一般多层前向神经网络有所不同,一般多层前向神经网络中的全部神经元的输入-输出特性都是相同的和静态的,而 PID 神经网络的隐含层是由作者提出的比例元、积分元和微分元组成的,是一种动态前向网络,更适合于系统控制;一般多层前向神经网络的隐层神经元个数、连接权初值是随机确定的,在其控制下的系统稳定性得不到保证,控制系统初始不稳定,网络的收敛性失去了基础,这是很多神经网络控制系统不能实用的原因之一.而 PID 神经网络的各层神经元个数、连接方式、连接权初值是按 PID 控制规律的基本原则和已有经验确定的,保证了系统稳定和参数的迅速收敛.

本文将在给出 PID 神经网络多变量控制器结构和算法的基础上,着重分析和证明 PID 神经网络多变量控制系统的收敛性和稳定性,最后给出其进行解耦控制的仿真实例和结论.

2 PID 神经网络多变量控制系统的结构形式和计算方法

如图1所示, PID 神经网络是三层前向网络,由并列的多个子网络组成,如果控制系统

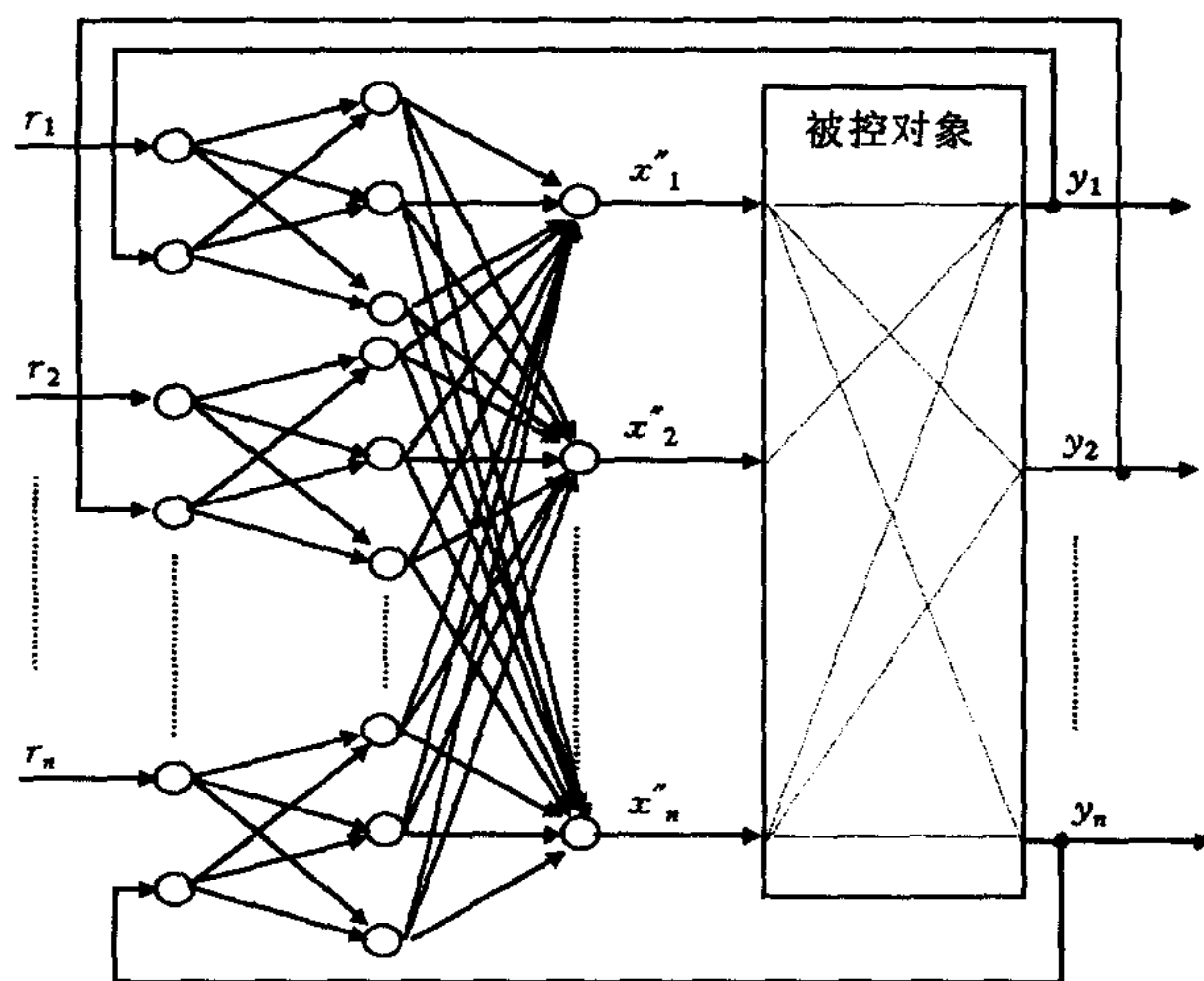


图1 PID 神经网络多变量控制系统结构

系统有 n 个被控变量,子网络就有 n 个.子网络的输入层至隐层是相互独立的,每个子网的输入层有两个神经元,分别输入系统一个变量的给定值和输出值;隐含层有3个神经元,其中包括1个比例元,1个积分元和1个微分元,它们的状态转换函数分别为上下限幅的比例、积分和微分函数;隐含层至输出层是互相交叉连接的,使整个 PID 神经网络结合为一体,网络的 n 个输出值为多变量对象的控制输入.

PID 神经网络的输入层和输出层神经元结构与一般多层前向神经网络类似,隐含层各神经元的输入函数为

$$net'_{sj}(k) = \sum_{i=1}^2 \omega_{sij} x_{si}(k). \quad (1)$$

上式中 ω_{sij} 为各子网输入层至隐含层的连接权值; x_{si} 为输入层神经元的输出值; $s=1,2,\dots,n$. 为子网序号; $i=1,2$, 为子网中输入层神经元序号; $j=1,2,3$, 为子网中隐含层神经元序号; k 为采样时刻;下同.

隐含层各神经元的状态函数有三种,分别为比例、积分和微分函数.比例元的状态为

$$u'_{s1}(k) = \begin{cases} 1, & u'_{s1}(k) > 1, \\ net'_{s1}(k), & -1 \leq u'_{s1}(k) \leq 1, \\ -1, & u'_{s1}(k) < -1; \end{cases} \quad (2)$$

积分元的状态为

$$u'_{s2}(k) = \begin{cases} 1, & u'_{s2}(k) > 1, \\ u'_{s2}(k-1) + net'_{s2}(k), & -1 \leq u'_{s2}(k) \leq 1, \\ -1, & u'_{s2}(k) < -1; \end{cases} \quad (3)$$

微分元的状态为

$$u'_{s3}(k) = \begin{cases} 1, & u'_{s3}(k) > 1, \\ u'_{s3}(k) - net'_{s3}(k-1), & -1 \leq u'_{s3}(k) < 1, \\ -1, & u'_{s3}(k) < -1; \end{cases} \quad (4)$$

隐含层各神经元的输出均为

$$x'_{sj}(k) = u'_{sj}(k). \quad (5)$$

3 PID 神经网络多变量控制系统的反传学习算法

包含 PID 神经网络和多变量对象在内的广义网络采用误差后向传播的批学习算法,学习的目标是使

$$J = \sum_{p=1}^n E_p = \frac{1}{m} \sum_{p=1}^n \sum_{k=1}^m [r_p(k) - y_p(k)]^2 = \frac{1}{m} \sum_{p=1}^n \sum_{k=1}^m e_p^2(k) \quad (6)$$

为最小. 式中 r_p 为系统给定值; y_p 为系统输出值; m 为每批采样点数, n 为被控变量个数.

按梯度法调节 PID 神经网络权值, 经 n_0 步训练和学习后, 隐含层至输出层的权值迭代公式为

$$w'_{sjh}(n_0 + 1) = w'_{sjh}(n_0) - \eta'_{sjh} \frac{\partial J}{\partial w'_{sjh}}. \quad (7)$$

上式中 η'_{sjh} 为该层的学习步长, w'_{sjh} 为隐含层至输出层的连接权值; $h=1, 2, \dots, n$ 为输出层神经元序号;

$$\frac{\partial J}{\partial w'_{sjh}} = - \frac{1}{m} \sum_{p=1}^n \sum_{k=1}^m \delta_{hp}(k) x'_{sj}(k), \quad (8)$$

其中

$$\delta_{hp} = 2[r_p(k) - y_p(k)] \operatorname{sgn} \frac{y_p(k+1) - y_p(k)}{x''_h(k) - x''_h(k-1)}, \quad (9)$$

式中 $x''_h(k)$ 为输出层神经元的输出值.

式(9)中用 $\operatorname{sgn}(\cdot)$ 项取代 $\frac{\partial y_p}{\partial x''_h}$, 以解决对象参数未知而造成的计算困难. 由式(8)可知, 该项仅为其中的一个相乘因子, 该项的正负决定收敛的方向, 其大小只决定收敛的速度.

输入层至隐含层的权值迭代公式为

$$w_{sij}(n_0 + 1) = w_{sij}(n_0) - \eta_{sij} \frac{\partial J}{\partial w_{sij}}. \quad (10)$$

上式中 η_{sij} 为该层的学习步长;

$$\frac{\partial J}{\partial w_{sij}} = -\frac{1}{m} \sum_{p=1}^n \sum_{h=1}^n \sum_{k=1}^m \delta_{sjh}(k) x_{si}(k); \quad (11)$$

其中
$$\delta_{sjh} = \delta_{hp} w_{sjh} \operatorname{sgn} \frac{u_{sj}(k) - u_{sj}(k-1)}{net_{sj}(k) - net_{sj}(k-1)}. \quad (12)$$

式(12)用 $\operatorname{sgn}(\cdot)$ 取代 $\frac{\partial u_{sj}}{\partial net_{sj}}$, 这种代替既不影响收敛方向, 又可简化和规范计算方法. 因为该项只是式(11)中的一个因子, 其符号决定收敛方向, 其值只影响收敛速度.

4 PID 神经元网络多变量控制系统的收敛性和稳定性分析

PID 神经元网络多变量控制系统的收敛性取决于学习步长的选取, 以下利用 Lyapunov 稳定性原理确定保证系统收敛的学习步长范围.

定理1. 当 PID 神经元网络的学习步长 η 满足

$$0 < \eta < \frac{1}{\epsilon^2}, \quad (13)$$

其中
$$\epsilon = -\frac{1}{2\sqrt{J}} \frac{\partial J}{\partial W}, \quad (14)$$

权值调整算法
$$W(n_0+1) = W(n_0) - \eta \frac{\partial J}{\partial W} \quad (15)$$

保证控制系统在学习过程中收敛. 其中 W 表示 PID 神经元网络的连接权值 w_{sij} 和 w'_{sjh} , J 由式(6)确定.

证明. 定义 Lyapunov 函数

$$L(n_0) = \frac{1}{2} \bar{E}^2(n_0), \quad (16)$$

其中
$$\bar{E}(n_0) = \sqrt{\frac{1}{m} \sum_{p=1}^n \sum_{k=1}^m [r_p(k) - y_p(k)]^2} = \sqrt{J}, \quad (17)$$

其中 $r_p(k)$ 为系统被控变量给定值, $y_p(k)$ 为系统被控变量实际值.

令
$$\Delta L(n_0) = L(n_0+1) - L(n_0) = \frac{1}{2} [\bar{E}^2(n_0+1) - \bar{E}^2(n_0)], \quad (18)$$

其中
$$\bar{E}(n_0+1) = \bar{E}(n_0) + \Delta \bar{E}(n_0), \quad (19)$$

误差变化量
$$\Delta \bar{E}(n_0) = \frac{\partial \bar{E}(n_0)}{\partial W} \cdot \Delta W, \quad (20)$$

网络权值变化量

$$\Delta W = -\eta \frac{\partial J}{\partial W} = -\eta \frac{\partial}{\partial W} [\bar{E}^2(n_0)] = -2\eta \bar{E}(n_0) \frac{\partial \bar{E}(n_0)}{\partial W}. \quad (21)$$

将式(19)~式(21)代入式(18), 则

$$\begin{aligned} \Delta L(k) &= \frac{1}{2} [\bar{E}^2(n_0+1) - \bar{E}^2(n_0)] = \frac{1}{2} \{ [\bar{E}(n_0) + \Delta \bar{E}(n_0)]^2 - \bar{E}^2(n_0) \} = \\ &= \frac{1}{2} \{ 2\bar{E}(n_0) \Delta \bar{E}(n_0) + [\Delta \bar{E}(n_0)]^2 \} = \\ &= \frac{1}{2} \left\{ 2\bar{E}(n_0) \frac{\partial \bar{E}(n_0)}{\partial W} \left[-2\eta \bar{E}(n_0) \frac{\partial \bar{E}(n_0)}{\partial W} \right] + \left[\frac{\partial \bar{E}(n_0)}{\partial W} \left(-2\eta \bar{E}(n_0) \frac{\partial \bar{E}(n_0)}{\partial W} \right) \right]^2 \right\} = \end{aligned}$$

$$2\left[\bar{E}(n_0) \frac{\partial \bar{E}(n_0)}{\partial W}\right]^2 \left[-\eta + \left(\eta \frac{\partial \bar{E}(n_0)}{\partial W}\right)^2\right]. \quad (22)$$

为保证系统的收敛性,必须满足 $\Delta L(n_0) < 0$, 即

$$\eta \left[\eta \left(\frac{\partial \bar{E}(n_0)}{\partial W} \right) - 1 \right] < 0. \quad (23)$$

解以上不等式,得

$$0 < \eta < \frac{1}{\left(\frac{\partial \bar{E}(n_0)}{\partial W} \right)^2}. \quad (24)$$

而由式(17)可求出

$$\frac{\partial \bar{E}(n_0)}{\partial W} = -\frac{1}{2\sqrt{J}} \frac{\partial J}{\partial W}. \quad (25)$$

定理1证毕.

推论1. 如果一个 PID 神经元网络多变量控制系统是收敛的,则其隐含层至输出层权值调整算法

$$w'_{sjh}(n_0 + 1) = w'_{sjh}(n_0) - \eta'_{sjh} \frac{\partial J}{\partial w'_{sjh}} \quad (26)$$

中的学习步长 η'_{sjh} 必然满足

$$0 < \eta'_{sjh} < \frac{1}{\epsilon_{sjh}^2}, \quad (27)$$

其中

$$\epsilon_{sjh} = -\frac{\sum_{p=1}^n \sum_{k=1}^m \delta'_{hp}(k) x'_{sj}(k)}{2\sqrt{m \sum_{p=1}^n \sum_{k=1}^m [r_p(k) - y_p(k)]^2}}. \quad (28)$$

证明略.

推论2. 如果一个 PID 神经元网络多变量控制系统是收敛的,则其输入层至隐含层的权值调整算法

$$w_{sij}(n_0 + 1) = w_{sij}(n_0) - \eta_{sij} \frac{\partial J}{\partial w_{sij}} \quad (29)$$

中的权值调整学习步长 η_{sij} 必须满足

$$0 < \eta_{sij} < \frac{1}{\epsilon_{sij}^2}, \quad (30)$$

其中

$$\epsilon_{sij} = -\frac{\sum_{p=1}^n \sum_{h=1}^n \sum_{k=1}^m \delta_{sjh}(k) x_{si}(k)}{2\sqrt{m \sum_{p=1}^n \sum_{k=1}^m [r_p(k) - y_p(k)]^2}}. \quad (31)$$

证明略.

推论1和推论2给出了 PID 神经元网络各层权值学习步长的取值范围,只要步长在此范围内,就可以保证系统的收敛性.在实际应用时,为计算简便起见,可先取步长为一个足够小的正数,然后根据推论2和推论3进行适当调整.

定理2. 一个 PID 神经元网络的连接权取初值时,如果其控制的多变量系统稳定,且其学习步长满足定理1条件,则这个 PID 神经元网络多变量控制系统必然是稳定的.

证明. 由定理2的条件,系统在PID神经元网络连接权取初值时稳定,再由定理1可知系统是收敛的,所以此PID神经元网络多变量系统在学习中和学习后必然稳定.定理2证毕.

5 PID 神经元网络多变量控制系统仿真实例

为了检验PID神经元网络多变量控制系统的性能,作者进行了大量的仿真研究,以下为一个仿真实例.设一个强耦合的非线性对象由以下方程描述:

$$\begin{cases} y_1(k+1) = 0.4y_1(k) + f_1[v_1(k), v_2(k)], \\ y_2(k+1) = 0.2y_2(k) + f_2[v_1(k), v_2(k)], \end{cases}$$

其中非线性函数为

$$\begin{cases} f_1[v_1(k), v_2(k)] = \frac{v_1(k)}{1 + v_1^2(k)} + 0.2v_1^3(k) + 0.5v_2(k), \\ f_2[v_1(k), v_2(k)] = \frac{v_2(k)}{1 + v_2^2(k)} + 0.4v_2^3(k) + 0.2v_1(k). \end{cases}$$

给定输入激励为

$$R_{10} = \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, R_{01} = \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

采用二输出的PID神经元网络对其进行控制,学习步长 $\eta=0.1$,每批的采样点数 $m=100$.用输入函数 R_{10}, R_{01} 分别训练网络100次后,系统对以上输入的输出响应如图2的(a)和(b)所示,学习的前30步中的系统误差平方均值衰减曲线如图2(c)所示.由仿真结果可知,PID神经元网络多变量控制器可以对这一类强耦合的非线性多变量对象实现有效的解耦控制,系统的动态性能很好,调节时间仅为一个采样周期,稳态误差为零;网络学习的收敛速度很快,学习过程中的误差单调递减,未陷入局部极小点.

6 结论

PID神经元网络是一种内含动、静特性的多层前向神经网络,网络的收敛速度很快且保证了控制系统的稳定性.PID神经元网络中比例元、积分元和微分元的存在,使PID神经元网络控制系统的响应

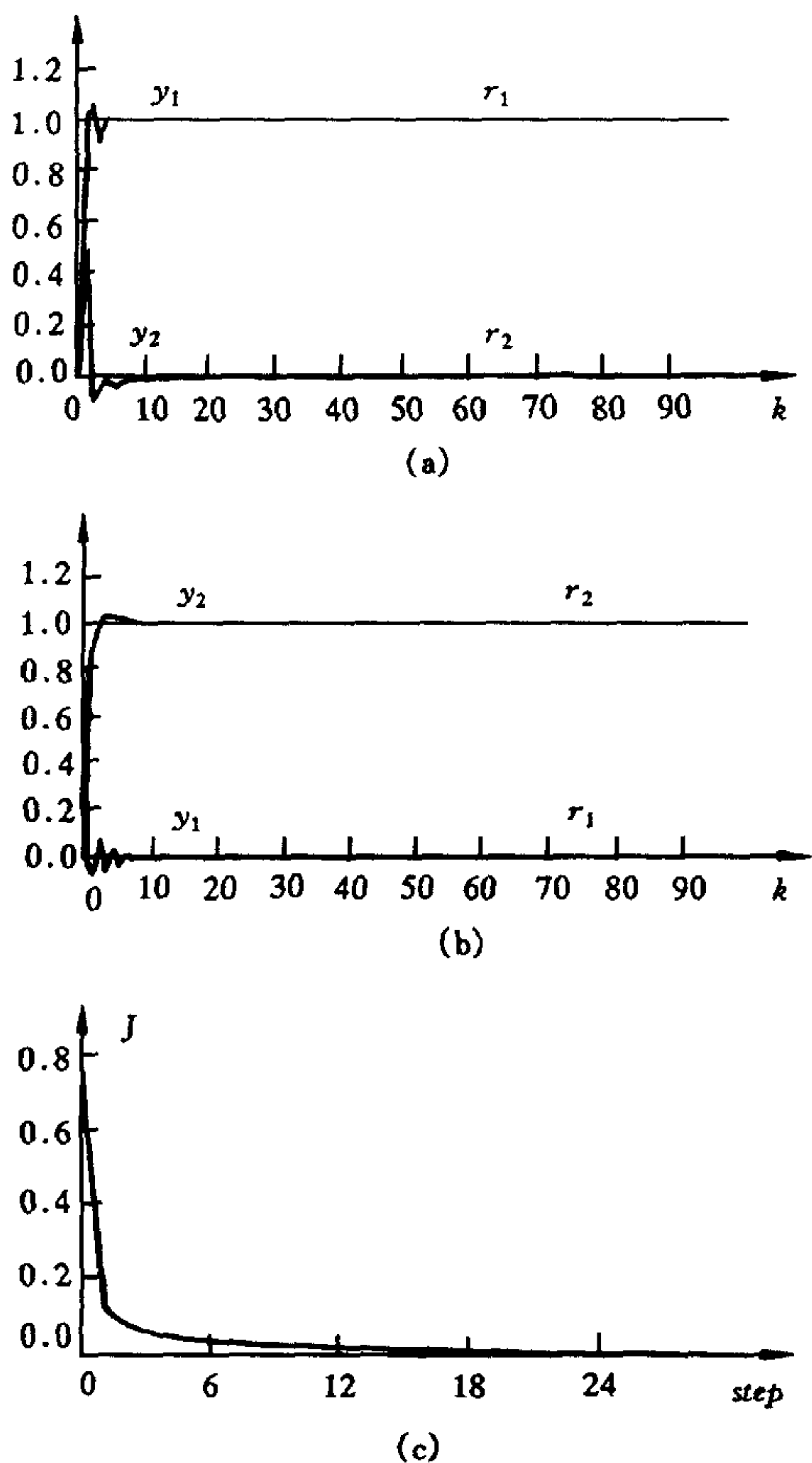


图2 PID 神经元网络多变量控制系统响应曲线

快、超调小、无静差. PID 神经元网络可以适用于多变量系统的解耦控制, 无需测量或辨识被控多变量对象的内部结构和参数, 可以得到良好的解耦控制效果.

参 考 文 献

- 1 Willis M J, Montague G A. Auto-tuning PI(D) controllers with artificial neural networks. In: Proc. 12th WCI-FAC. Sydney, Australia:1993,4. 61—64
- 2 Chan K C, Lenong S S. A neural network PI controller tuner. *Artif. Intell. Eng. (UK)*,1995,9(3):167—176
- 3 Ma X Loh N K. Tuning PID controllers by neural networks. In: Proc. Artificial Neural Network in Engineering Conference. St Louis: 1991. 625—630
- 4 Chen C L, Chang F Y. Design and analysis of neural/fuzzy variable structural PID control systems. *IEE Pro-Control Theory Appl.* 1996,143(2):200—208
- 5 Bahrami M, Tait K E. Design of direct controller of PID type by receptive field neural networks. *IJCNN'93-Nagoya*:1993. 1809—1912.
- 6 Akhyar S, Omatu S. Self-tuning PID control by neural-networks. *IJCNN '93-Nagoya*, 1993,3:2749—1752
- 7 李卓, 萧德云, 何世忠. 基于神经网络的模糊自适应 PID 控制方法. *控制与决策*,1996,11(3):340—345
- 8 王耀南, 蔡自兴. 基于神经元网络的智能 PID 控制及应用. *信息与控制*,1994,23(3):185—189
- 9 谭永红. 神经网络自适应 PID 控制及其应用. *模式识别与人工智能*,1993,6(1):81—85

舒怀林 男, 1954年11月生. 1982年7月在西安交通大学获学士学位, 1989年7月在西安矿业学院获硕士学位, 1997年12月在华中理工大学获博士学位. 主要研究领域为智能仪表、电力拖动、模糊控制、神经网络控制.