文章编号: 1000-8152(2007)06-0949-05

基于神经网络的非线性系统预测函数控制

张日东1,2, 王树青1

(1. 浙江大学 先进控制技术研究所 工业控制技术国家重点实验室, 浙江 杭州 310027;

2. 杭州电子科技大学 信息与控制研究所, 浙江 杭州 310018)

摘要: 针对离散非线性系统, 利用神经网络非线性激励函数的局部线性表示, 提出一种可用于非线性过程的神经网络预测函数控制方法并给出了控制律的收敛性分析. 该方法将复杂的神经网络非线性预测方程转化成直观而有效的线性形式, 同时利用线性预测函数方法求得解析的控制律, 避免了复杂的非线性优化求解, 仿真结果表明了算法的有效性.

关键词: 神经网络模型; 预测函数控制; 非线性过程

中图分类号: TP273 文献标识码: A

Neural network based predictive functional control for nonlinear systems

ZHANG Ri-dong^{1,2}, WANG Shu-qing¹

- (1. Institute of Advanced Process Control, National Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou Zhejiang 310027, China;
- 2. Information and Control Institute, Hangzhou Dianzi University, Hangzhou Zhejiang 310018, China)

Abstract: A neural network predictive functional control method and its convergence analysis for discrete nonlinear systems are presented by using the local linearization of nonlinear activation functions. The method gives a direct and effective multi-step predicting method and uses linear methods to obtain the control law and avoids the complicated nonlinear optimization. Simulation results are presented in the article showing the efficiency of this method.

Key words: neural network model; predictive functional control; nonlinear process

1 引言(Introduction)

预测函数控制(PFC)^[1]最早由法国ADERSA公司的Richalet与德国IITB公司的Kuntze等人于1986年共同提出,并在机器人跟踪控制、导弹跟踪^[2]、热焓控制^[3]等得到了成功的应用. 它在保持模型预测控制优点的同时,还使得所产生的控制输入更具有规律性,并且可有效地减少计算工作量,从而适应了快变过程对控制算法的要求. 关于线性系统的预测函数控制,人们已经做了许多工作,建立了比较完善的理论体系,并且日趋成熟. 而对于非线性系统的预测函数控制. 由于缺乏有效的数学分析工具及建立精确模型的困难,使得目前的非线性预测函数控制成果不多. 近年来,人们用神经网络对非线性系统进行控制做了大量研究,也出现了许多神经网络预测及预测函数控制算法^[4~9],这些方法为神经网络非线性预测控制的分析与综合问题提供了一个基本思

路. 但目前这些方法都还涉及到结构复杂、计算量大以及需要非线性搜索求解等缺点. 研究表明, 如何利用较少神经网络设计能提供直观有效预测方程和避免非线性求解的预测函数控制器仍然是一项十分有意义的工作.

本文提出一种神经网络非线性预测函数控制方法,该方法仅用一个神经网络来综合非线性系统的预测函数控制,充分考虑降低系统结构的复杂性和减轻运算负担的需要.主要思路是利用非线性激励函数的局部线性表示,将非线性预测方程转化为一系列简单直观的线性预测方程,并利用线性预测函数控制方法求取控制律,从而不需要复杂的非线性优化搜索等求解方法并使控制律具有解析形式.

2 非线性系统的模型及其表示(Nonlinear model and its treatment)

$$y(t) = f[y(t-1), \dots, y(t-n), u(t-d-1), \dots, u(t-d-m)].$$
 (1)

其中: n, m分别为时间序列 $\{y(t)\}, \{u(t)\}$ 的阶次, d+1为系统的时滞.

该系统可以用3层BP神经网络表示为

$$y(t) = g\{\sum_{i=1}^{I} w_3(i)g[\sum_{j=1}^{m} w_2(i,j)u(t-d-j) + \sum_{j=m+1}^{m+n} w_2(i,j)y(t+m-j)]\}.$$
 (2)

其中: $w_2(i, j), w_3(i) (i = 1, \dots, I, j = 1, \dots, m + n)$ 为层与层之间的连接权, 输入结点为n + m个, 隐层结点为1个, g为激励函数:

$$g(x) = \frac{1}{1 + e^{-x}}. (3)$$

命题 非线性系统(2)可以表示为如下的线性部分与非线性部分组合的形式:

$$y(t) = a_1 y(t-1) + \dots + a_n y(t-n) + b_0 u(t-d-1) + \dots + b_{m-1} u(t-d-m) + N + \varepsilon(\theta(t)).$$
(4)

其中:

$$\begin{aligned} a_1 &= \sum_{i=1}^I M_i w_2(i,m+1) b_0 = \sum_{i=1}^I M_i w_2(i,1), \\ a_2 &= \sum_{i=1}^I M_i w_2(i,m+2) b_1 = \sum_{i=1}^I M_i w_2(i,2), \\ &\vdots \\ a_n &= \sum_{i=1}^I M_i w_2(i,m+n) b_{m-1} = \sum_{i=1}^I M_i w_2(i,m), \\ N &= \sum_{i=1}^I w_3(i) g'(s_{31}) [g(s_{2i}) - g'(s_{2i}) s_{2i}] + \\ g(s_{31}) - g'(s_{31}) s_{31}, \\ \varepsilon(\theta(t)) &= \sum_{i=1}^I w_3(i) g'(s_{31}) \varepsilon_{2i}(\theta(t)) + \varepsilon_1(\theta(t)), \\ M_i &= w_3(i) g'(s_{31}) g'(s_{2i}), \end{aligned}$$

 s_{31}, s_{2i} $(i = 1, 2, \dots, I)$ 为非线性函数g的中心点. $\theta(t)$ 代表向量 $(y(t-1), \dots, y(t-n), u(t-d-1), \dots, u(t-d-m))$.

证 对式(2)中的"q"进行处理,令

$$s_3(t) = \sum_{i=1}^{I} w_3(i)g[\sum_{j=1}^{m} w_2(i,j)u(t-d-j) + \sum_{j=m+1}^{m+n} w_2(i,j)y(t+m-j)],$$
 (5)

$$s_{2i}(t) = \sum_{j=1}^{m} w_2(i,j)u(t-d-j) + \sum_{j=m+1}^{m+n} w_2(i,j)y(t+m-j),$$
(6)

则有 $y(t) = g[s_3(t)]$,利用泰勒展开,其中心点设为 s_{31} ,得到下式:

$$y(t) = g(s_{31}) + g'(s_{31})[s_3(t) - s_{31}] + \varepsilon_1(\theta(t)) = g'(s_{31})s_3(t) + g(s_{31}) - g'(s_{31})s_{31} + \varepsilon_1(\theta(t)) = g'(s_{31}) \sum_{i=1}^{I} w_3(i)g[s_{2i}(t)] + g(s_{31}) - g'(s_{31})s_{31} + \varepsilon_1(\theta(t)).$$

$$(7)$$

这里 ε_1 是函数符号. 并且定义 $\theta(t+i)$ 为 $(y(t+i-1), \cdots, y(t+i-n), u(t+i-d-1), \cdots, u(t+i-d-m))$, 同理,对 $g[s_{2i}(t)]$ 也类似处理,设中心点为 s_{2i} $(i=1,2,\cdots,I)$, 则有

$$g[s_{2i}(t)] = g(s_{2i}) + g'(s_{2i})[s_{2i}(t) - s_{2i}] + \varepsilon_{2i}(\theta(t)).$$
(8)

其中 $\varepsilon_{2i}(i=1,2,\cdots,I)$ 也是函数符号.

将式(8)代入式(7), 并将非线性部分 $\varepsilon_{2i}(\theta(t))$ ($i=1,2,\cdots,I$)合并成 $\varepsilon(\theta(t))$ 从而有

$$y(t) = \sum_{i=1}^{I} w_{3}(i)g'(s_{31})\{g(s_{2i}) + g'(s_{2i})[s_{2i}(t) - s_{2i}]\} + g(s_{31}) - g'(s_{31})s_{31} + \varepsilon(\theta(t)) =$$

$$\sum_{i=1}^{I} w_{3}(i)g'(s_{31})g'(s_{2i})s_{2i}(t) +$$

$$\{\sum_{i=1}^{I} w_{3}(i)g'(s_{31})[g(s_{2i}) - g'(s_{2i})s_{2i}] +$$

$$g(s_{31}) - g'(s_{31})s_{31}\} + \varepsilon(\theta(t)), \tag{9}$$

其中 $\varepsilon(\theta(t))$ 为非线性部分:

$$\varepsilon(\theta(t)) = \sum_{i=1}^{I} w_3(i)g'(s_{31})\varepsilon_{2i}(\theta(t)) + \varepsilon_1(\theta(t)).$$
(10)

为书写简便, 令式(9)中的第2项为N, 第1项中的 $w3(i)g'(s_{31})g'(s_{2i})$ 为 M_i , 则可得到式(9)的等价式:

$$y(t) = \sum_{i=1}^{I} M_{i} s_{2i}(t) + N + \varepsilon(\theta(t)) =$$

$$\sum_{i=1}^{I} M_{i} \left[\sum_{j=1}^{m} w_{2}(i,j) u(t-d-j) + \frac{\sum_{j=m+1}^{m+n} w_{2}(i,j) y(t-j+m) \right] + N + \varepsilon(\theta(t)). \quad (11)$$

把y(t)写成离散差分方程形式:

$$y(t) = a_1 y(t-1) + \dots + a_n y(t-n) + b_0 u(t-d-1) + \dots +$$

$$b_{m-1}u(t-d-m)+N+\varepsilon(\theta(t)). \tag{12}$$

式(11)与式(12)对比系数得
$$\begin{cases}
a_1 = \sum_{i=1}^{I} M_i w_2(i, m+1) b_0 = \sum_{i=1}^{I} M_i w_2(i, 1), \\
a_2 = \sum_{i=1}^{I} M_i w_2(i, m+2) b_1 = \sum_{i=1}^{I} M_i w_2(i, 2), \\
\vdots \\
a_n = \sum_{i=1}^{I} M_i w_2(i, m+n) b_{m-1} = \\
\sum_{i=1}^{I} M_i w_2(i, m).
\end{cases}$$
(13)

证毕.

3 控制系统设计(Controller design)

3.1 基于模型的预测输出(Predicted output)

注意到N在每个控制周期为某常数,式(4)可写为

$$y(t) = A_1 y(t-1) + \dots + A_{n+1} y(t-n-1) + B_{1,0} \Delta u(t-d-1) + \dots + B_{1,m-1} \Delta u(t-d-m) + \Delta \varepsilon(\theta(t)).$$
 (14)

其中: $A_1 = 1 + a_1$, $A_i = a_i - a_{i-1}$ $(i = 2, 3, \dots, n)$, $A_{n+1} = -a_n$, $B_{1,i} = b_i$ $(i = 0, 1, \dots, m-1)$.

将最优预测输出分成3部分: 一部分取决于过去的输入和输出,用 Y_{past} 表示; 另一部分取决于现在和未来的输入,表为GU; 第3部分为预测误差,包括非线性误差 E_1 和外界干扰引起的误差 E_2 . 这里预测时域和控制时域均为p, 其中 E_2 也就是预测控制中的反馈校正项,它包含 $p \cap y(t) - \hat{y}(t)$, 因此得到

$$Y = Y_{\text{past}} + GU + E_1 + E_2. \tag{15}$$

其中:

$$Y = (y(t+d+1)/t, \dots, y(t+d+p)/t))^{T},$$

$$Y_{\text{past}} = (y_{\text{past}}(t+d+1), \dots, y_{\text{past}}(t+d+p))^{T},$$

$$U = (\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+p-1))^{T},$$

$$E_{1} = (\varepsilon(\theta(t+1)), \varepsilon(\theta(t+2)), \dots, \varepsilon(\theta(t+p)))^{T},$$

$$E_{2} = (y(t) - \hat{y}(t), y(t) - \hat{y}(t), \dots, y(t) - \hat{y}(t))^{T},$$

$$G = \begin{bmatrix} B_{1,0} \\ B_{2,0} & B_{1,0} & \mathbf{0} \\ \vdots & \vdots & \vdots \\ B_{p,0} & B_{p-1,0} & \cdots & B_{1,0} \end{bmatrix}.$$

其中: y(t)为实际输出, $\hat{y}(t)$ 为神经网络模型输出. 由于 Y_{past} 是系统的自由输出, 因此它可由其神经网络模型计算. 而G 中的元素计算如下:

$$B_{1,0} = b_0$$
,

$$B_{k,0} = b_{k-1} + \sum_{j=1}^{k-1} A_j B_{k-j,0}, \ k = 2, \dots, p.$$
 (16)

3.2 系统控制律求取(Control law)

与其他的预测控制算法不同的是, 预测函数控制 认为控制量与一组相应于过程特性和跟踪设定值的 基函数有关, 是基函数的线性组合, 即

$$u(t+i) = \sum_{j=1}^{M} \mu_j(k) f_j(i).$$
 (17)

其中: $\mu_j(k)$ 为基函数线性组合的权系数, $f_j(i)$ 为基函数在第i个采样周期的取值, M为基函数的阶数, 依据跟踪设定值和过程模型确定.

取参考轨迹为

$$\begin{cases} y_r(t+d) = y(t), \\ y_r(t+d+k) = \alpha^k y(t) + (1-\alpha^k) y_s, \\ k = 1, 2, \dots, p-1. \end{cases}$$
 (18)

其中: α 为输出柔化系数, y_s 为设定值. 另外, 在多步预测情况下, 分别设参考轨迹向量和目标函数为

$$\begin{cases}
Y_r = (y_r(t+d+1), \dots, y_r(t+d+p))^{\mathrm{T}}, \\
J = \min\{(Y_r - Y)^{\mathrm{T}}(Y_r - Y) + \beta^2 U^{\mathrm{T}}U\}.
\end{cases} (19)$$

其中 β^2 为加权项. 为适应多步预测控制算法的要求, 当预测步长取为p, 并且有 $u(t) = u(t+1) = \cdots = u(t+p-1)$ 时, 结合式(17)求解性能指标(19)可得控制律:

$$\Delta u(t) = G'^{\mathrm{T}}(Y_r - Y_{\mathrm{past}} - E_1 - E_2) / (G'^{\mathrm{T}}G' + \beta^2).$$

其中 $G' = [B_{1,0}, B_{2,0}, \cdots, B_{p,0}]^{\mathrm{T}}$, 不失一般性并为后续分析简便, 本文选取式(19)的简化形式如下:

$$J' = \min(y_r(t+d+p) - y(t+d+p)/t)^2. (20)$$

结合式(17)求解性能指标(20)可得控制律

$$\Delta u(t) - (y_r(t+d+p) - y_{past}(t+d+p) - \varepsilon(\theta(t+p)) - y(t) + \hat{y}(t)/B_{p,0}),$$

$$u(t) = \mu_1 = u(t-1) + \Delta u(t).$$

由于系统为非线性的,因此 $\varepsilon(\theta(t+p))$ 是未知的,本文采取迭代递推求取控制律,算法为

$$\begin{cases} \varepsilon(\theta(t+p))^{0} = 0, \\ \Delta u(t)^{0} = (y_{r}(t+d+p) - y_{\text{past}}(t+d+p) - \varepsilon(\theta(t+p))^{0} - y(t) + \hat{y}(t))/B_{p,0}, \\ \varepsilon(\theta(t+p))^{k+1} = \varepsilon(\theta(t+p))^{k} + \delta(k)(y_{m}^{k}(t+d+p) - y^{k}(t+d+p)), \\ \Delta u(t)^{k+1} = (y_{r}(t+d+p) - y_{\text{past}}(t+d+p) - \varepsilon(\theta(t+p))^{k+1} - y(t) + \hat{y}(t))/B_{p,0}. \end{cases}$$
(21)

其中: $\varepsilon(\theta(t+p))^0$ 为 $\varepsilon(\theta(t+p))$ 的初始值, $\Delta u(t)^0$ 为 $\Delta u(t)$ 的初始值, 上标k代表各分量第k步的值. 而 $y_m^k(t+d+p)$ 为求出的 $\Delta u(t)^k$ 代入神经网络模型而求出的输出, $\delta(k)$ 为收敛因子, 它是正标量递减序列, $\delta(k) \in (0,1)$, $\lim_{k\to\infty}\delta(k)=0$. 适当选取 $\delta(k)$ 上述递推可以很快收敛, 通常取 $\delta(k)=1/k$. 当 $y_m^k(t+d+p)=y(t+d+p)$ 时, 求出的 $\Delta u(t)^k$ 即为最优控制律. 实际上若 $\|\varepsilon(\theta(t+p))^{k+1}-\varepsilon(\theta(t+p))^k\|$ 足够小,则即可认为 $\Delta u(t)^k$ 为控制律.

现将本文算法归纳如下:

- 1) 用输入输出样本对神经网络进行一段时间的 离线训练, 得到权值的初值.
 - 2) 读取系统输出y(t), 辨识网络的权值.
 - 3) 利用第2节的方法重新表示系统.
 - 4) 计算系统的自由输出 Y_{past} .
 - 5) 用式(18)计算参考轨迹.
 - 6) 用式(21)求取控制量,进行控制.
 - 7) 返回到第2)步.

4 收敛性分析(Convergence analysis)

由式(21), 对充分大的k来说, $\delta(k)$ 将任意小, 为此, $\varepsilon(\theta(t+p))^k$, $\Delta u(t)^k$ 的变化都将越来越小, 这可从式(21)的第3,4式看出, 当 $\varepsilon(\theta(t+p))^k$ 收敛于真值 ε 时, $y^k(t+d+p)$ 将收敛于 $y_m^k(t+d+p)$, 从而得到最优控制律. 因此本节从分析 $\varepsilon(\theta(t+p))^k$ 的收敛性入手, 若 $\varepsilon(\theta(t+p))^k$ 收敛于真值 ε , 则由上面的分析可知, 最优控制律的求解是收敛的. 本文控制律的收敛性可以借助伴随微分方程的稳定性质来确定.

引理[10] 对于以下递推算法

$$\begin{cases} \alpha(k) = a(k-1) + \delta(k)Q(k, \alpha(k-1), \Phi(k)), \\ \Phi(k) = A(\alpha(k-1))\Phi(k-1) + B(\alpha(k-1))e(k). \end{cases}$$

式中: $\alpha(k)$ 称为"估计量",并定义 $D_s = \{\alpha | A(\alpha)\}$ 的所有特征值都严格在单位圆内, $D_R \not= D_s$ 的一个连通开子集, 在子集 D_R 上, 式(22)中的函数满足以下正则条件C1)~C5):

- C1) 函数 $Q(k, \alpha, \Phi)$ 在 $(\bar{\alpha}, \bar{\Phi})$ 附近关于 α 和 Φ 是 Lipschitz连续的, 其中 $\bar{\alpha} \in D_R$, $\bar{\Phi}$ 是任意的; 且关于 α 和 Φ 是连续可微的.
- C2) 对 $\bar{\alpha} \in D_R$, 矩阵函数 $A(\alpha)$, $B(\alpha)$ 关于 α 是 Lipschitz连续的.
- C3) $\{e(k)\}$ 是个独立的随机向量序列, 它使得 $\lim_{k\to\infty} \mathrm{E}\{Q(k,\bar{\alpha},\Phi(k,\bar{\alpha}))\} \to f^*(\alpha); \forall \bar{\alpha}\in D_R.$ 其中 $\Phi(k,\bar{\alpha})$ 由下式定义:

$$\begin{cases} \Phi(k,\bar{\alpha}) = A(\bar{\alpha})\Phi(k-1,\bar{\alpha}) + B(\bar{\alpha})e(k), \\ \Phi(0,\bar{\alpha}) = 0. \end{cases}$$

C4)
$$\sum_{k=1}^{\infty} \delta(k) = \infty, \sum_{k=1}^{\infty} [\delta(k)]^p < \infty, (p > 1).$$

C5) $\delta(k)$ 是正标量递减序列, $\lim_{k\to\infty}\delta(k)=0$,

$$\lim_{k\to\infty}\sup[\frac{1}{\delta(k-1)}-\frac{1}{\delta(k)}]<\infty.$$

如果对估计量 $\bar{\alpha} \in D_R$ 的任何可能值能够定义平稳随机函数 $Q(k, \bar{\alpha}, \Phi(k, \bar{\alpha}))$,则式(22)算法的伴随微分方程为

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\alpha_D(\tau) = f^*(\alpha_D(\tau)). \tag{23}$$

其中 τ 为伪时间, $\tau = \sum_{i=1}^{k} \delta(k)$, 而

$$f = \lim_{k \to \infty} E\{Q(k, \bar{\delta}, \Phi(k, \bar{\alpha}))\}.$$
 (24)

若方程(23)存在一个正函数 $V(\alpha_D)$, 使得

$$\begin{cases}
\frac{\mathrm{d}}{\mathrm{d}\tau}V(\alpha_D) \leq 0, \forall \alpha_D \in D_A, \\
\frac{\mathrm{d}}{\mathrm{d}\tau}V(\alpha_D) = 0, \forall \alpha_D \in D_C, D_C \in D_A,
\end{cases} (25)$$

则当 $k \to \infty$ 时, $\alpha(k)$ 以概率1趋于不变集 D_C ,即 $\alpha(k) \stackrel{w.p.1}{\underset{k\to\infty}{\longrightarrow}} D_C$. 同时,如果 α^* 是微分方程(23)的整体渐近稳定平衡点,则当 $k \to \infty$ 时, $\alpha(k)$ 又以概率1收敛于 α^* ,也就是 $\alpha(k) \stackrel{w.p.1}{\underset{k\to\infty}{\longrightarrow}} \alpha^*$.

定理 1 对于非线性系统(1), 如果将其神经网络模型式(2)表示为式(4)的形式, 控制器设计为式(21)的形式, 控制器设计参数 $\delta(k)$ 是正标量递减序列, 且 $\delta(k) \in (0,1)$, $\lim_{k \to \infty} \delta(k) = 0$. 则系统的预测函数控制律是收敛的.

证 1) 本文的递推算法.

由式(21)及式(15)得

$$\varepsilon(\theta(t+p))^{k+1} = \varepsilon(\theta(t+p))^k + \delta(k)(y_m^k(t+d+p) - y_{\text{past}}(t+d+p) - B_{p,0}\Delta u(t)^k - \varepsilon(\theta(t+p)) - y(t) + \hat{y}(t)).$$
(26)

考虑到在当前递推步下: $y_m^k(t+d+p)$, $y_{past}(t+d+p)$, $\Delta u(t)^k$, $\hat{y}(t)-y(t)$ 均为常量, 并合并其中的 $\varepsilon(\theta(t+p))^k$ 得到

$$\varepsilon(\theta(t+p))^{k+1} =$$

$$F\varepsilon(\theta(t+p))^{k} + \delta(k)(y_{m}^{k}(t+d+p) - y_{\text{past}}(t+d+p) - B_{p,0}\Delta u(t)^{k} - y(t) + \hat{y}(t)) = F\varepsilon(\theta(t+p))^{k} + D,$$
(27)

其中D, F为常数:

$$D = \delta(k)(y_m^k(t+d+p) - y_{\text{past}}(t+d+p) - B_{p,0}\Delta u(t)^k - y(t) + \hat{y}(t)),$$

$$F = 1 - \delta(k).$$
(28)

而当拟合点为多个时, D, F分别变为相应拟合点个数的矩阵.

2) 正则条件分析.

首先定义 D_s ={F|F的所有特征值都严格在单位 圆内}. 根据F的特点可知

$$\lambda_k(F) = 1 - \delta(k) < 1. \tag{29}$$

可见 D_S 为全平面, 即 $D_R = D_S = R^n$. 其次, 由式(27)及 $\delta(k) = 1/k$, 条件C1)~C5)均可满足. 于是,式(27)的伴随微分方程为

$$\begin{split} \frac{\mathrm{d}\varepsilon_{D}(\tau)}{\mathrm{d}\tau} &= f^{*}(\varepsilon_{D}(\tau)) = \\ \lim_{k \to \infty} \mathrm{E}\{y_{m}^{k}(t+d+p) - y_{\mathrm{past}}(t+d+p) - \\ B_{p,0}\Delta u(t)^{k} - \varepsilon_{D}(\tau) - y(t) + \hat{y}(t)\}, \ (30) \end{split}$$

而

$$y_m^k(t+d+p) =$$

$$y_{\text{past}}(t+d+p) + B_{p,0}\Delta u(t)^k + \bar{\varepsilon} + y(t) - \hat{y}(t),$$

进而有

$$f^*(\varepsilon_D(\tau)) = \bar{\varepsilon} - \varepsilon_D(\tau).$$
 (31)

式中 $\bar{\epsilon}$ 为 $\epsilon(\theta(t+p))$ 的真值.

设 ε_{D}^{*} 是微分方程(31)的平衡点,则有

$$\varepsilon_D^* = \bar{\varepsilon}. \tag{32}$$

3) 稳定性分析.

取微分方程(31)的Lyapunov函数

$$V(\varepsilon_D) = \frac{1}{2}(\bar{\varepsilon} - \varepsilon_D)^2 > 0, \tag{33}$$

则

$$\frac{\mathrm{d}}{\mathrm{d}\tau}V(\varepsilon_D) = (\bar{\varepsilon} - \varepsilon_D)\frac{\mathrm{d}}{\mathrm{d}\tau}(\bar{\varepsilon} - \varepsilon_D) = -(\bar{\varepsilon} - \varepsilon_D)^2 \le 0, \ \forall \varepsilon_D,$$
(34)

目.

$$\frac{\mathrm{d}}{\mathrm{d}\tau}V(\varepsilon_D) = 0, \ \varepsilon_D = \varepsilon_D^* = \bar{\varepsilon}. \tag{35}$$

可见, 微分方程存在不变集 $D_C = \{\bar{\varepsilon}\}$, 而吸收域为全平面, 则由引理知 $\varepsilon(\theta(t+p))^k$ 收敛于 $\bar{\varepsilon}$. 这就说明了控制律是收敛的.

证毕.

5 仿真研究(Simulation study)

设被控对象的真实模型为

$$y(t) = \frac{0.91y(t-1) + u(t-5)}{1 + u(t-6)} + e(t)/\Delta.$$

仿真是在MATLAB6.5环境下编程实现,整个仿真过程依上面的7个步骤进行:首先用输入输出样本对神经网络进行一段时间的离线训练,得到权值的初值.然后依次执行步骤2)~7).仿真研究给出了本文方法与一般PFC方法控制该对象的对比

结果, 图1为一般PFC的仿真结果, 图2为本文方法的结果. 其中: 设定值取为 $y_s=0.5$, 控制参数选为p=5, $\beta^2=1$, $\alpha=0.65$, e(t)是-0.01至0.01均匀分布的独立的随机数. 神经网络结构取为3-5-1. 而一般PFC采用模型

$$y(t) = a_1y(t-1) + b_1u(t-5) + b_2u(t-6)$$

来描述该非线性对象, 用递推最小二乘法来辨识模型的参数 a_1, b_1, b_2 . 可见, 一般的PFC方法没有实现很好的控制, 控制过程中震荡相对剧烈、超调量大一些, 最终有一定的余差. 而本文方法的效果不错.

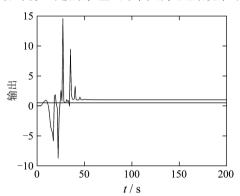


图 1 一般PFC的仿真结果

Fig. 1 Output of predictive functional control

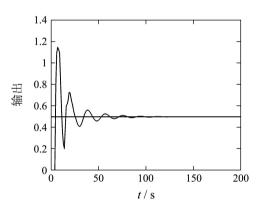


图 2 本文方法的仿真结果

Fig. 2 Output of the proposed method

6 结论(Conclusions)

本文将神经网络模型应用于预测函数控制,提出了一种非线性预测函数控制算法,该方法充分考虑了降低结构复杂性、减轻运算负担的要求,用一个神经网络给出了直观有效的线性预测方程形式,不需要非线性求解控制律.经仿真证明该方法是有效的.

参考文献(References):

[1] KUNTZE H B, JACUBASCH A, RICHALET J, et al. On the predictive functional control of elastic industrial robot[C]//Proc of the 25th IEEE Conference on Decision and Control. [S.l.]: [s.n.], 1986: 1877 – 1881.

(下转第958页)