

文章编号:1001-9081(2006)06-1398-03

基于递归神经网络模型预测控制的模型平稳切换

杜福银,徐 扬,陈树伟

(西南交通大学 智能控制开发中心,四川 成都 610031)

(dufynew@sohu.com)

摘 要:不同生产条件下的控制系统可视多模型控制系统,但多模型控制在模型切换时会引起系统的瞬态响应。采用递归神经网络建立系统的多个模型,基于模型预测控制进行控制模型切换,克服了模型切换时引起的系统瞬态响应,实现系统的平稳切换。并通过仿真表明这种切换策略明显改善了模型切换过程的动态性能。

关键词:比例积分微分控制器;递归神经网络;模型预测控制;模型切换;平稳

中图分类号: TP18 **文献标识码:** A

Smooth model switching scheme based on recurrent neural network model predictive control

DU Fu-yin, XU Yang, CHEN Shu-wei

(Intelligence Control Development Center, Southwest Jiaotong University, Chengdu Sichuan 610031, China)

Abstract: The control system of differ product conditions could be regarded as a multi-model control system. But model switching will result in transient response of the plant controlled. A method to establish multiples models of system using recurrent neural network and complete switch process based on model predictive control was proposed. Which avoid transient response of switch process and achieve model switching smoothly. Simulation performances indicate that the model switching scheme improve dynamic quality of switch process greatly.

Key words: Proportional-Integral-Differential controller(PID); recurrent neural network; model predictive control; model switching; smooth

0 引言

参数时变或结构未知系统的控制器的设计,长时间以来一直是从事控制的工作者所关心的事情,显然,一个固定不变的控制器是无法适应这种环境的。Narendra 等提出的多模型自适应控制是解决这类问题的一个新的途径,如何实现模型的平稳切换是多模型自适应控制的一个关键的问题^[1~3]。

基于模型的预测控制(MPC)是 20 世纪 70 年代后期发展起来的一类新型计算机控制方法,由于其所具有的预测模型、滚动优化和反馈校正三个特征,已在工业实时控制中得到广泛关注^[4~7]。

本文提出基于系统递归神经网络模型的平稳切换策略,即当系统需要切换时,先基于即将切入的控制器和相应模型以及实际被控对象的输入输出优化出一个能使系统输出满足控制要求的控制量,将此控制量和即将切入的控制器切入实际系统,来实现模型的平稳切换。

1 系统递归神经网络模型建立

1.1 网络模型结构及输出计算

改进的 Elman 网络是一种典型的动态递归神经网络,其反馈连接由一组“结构单元”组成,用于记忆隐层过去的状态,并且在下一时刻连同网络输入一起作为隐层单元的输入,这一性质使得递归神经网络具有动态记忆功能。网络结构如

图 1 所示,改进的 Elman 网络与基本 Elman 网络的不同之处在于,改进的 Elman 网络的结构单元中引入了固定的增益 α ($0 \leq \alpha \leq 1$) 的自反馈连接, α 越接近于 1,考虑的时刻越远, $\alpha = 0$ 时,改进的 Elman 网络就退化为基本的 Elman 网络,因而改进的 Elman 网络可以很容易模拟高阶系统。

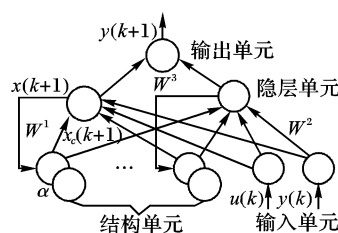


图 1 改进的 Elman 神经网络结构示意图

网络的外部输入为 $u(k)$, $y(k)$, 输出为 $y(k+1)$, 若记隐层的输出为 $x(k+1)$, 结构单元层的输出为 $x_c(k)$, 则有如下非线性状态空间表达式成立:

$$\begin{cases} x(k+1) = f(w^1 x_c(k) + w^2 u(k)) \\ x_c(k) = x(k) + \alpha x_c(k-1) \\ y(k+1) = g(w^3 x(k+1)) \end{cases} \quad (1)$$

其中 w^1, w^2, w^3 分别是结构单元到隐层、输入层到隐层、以及隐层到输出层的连接权矩阵, $f(\cdot)$ 和 $g(\cdot)$ 分别为隐层单元和输出单元激活函数所组成的非线性向量函数。

1.2 系统网络模型的学习算法

收稿日期:2005-12-26;修订日期:2006-02-22 基金项目:国家自然科学基金资助项目(60474022)

作者简介:杜福银(1969-),男,陕西宝鸡人,博士研究生,主要研究方向:智能控制;徐扬(1956-),男,河南新乡人,教授,博士,主要研究方向:智能信息处理;陈树伟(1977-),男,河南新乡人,博士研究生,主要研究方向:智能信息处理。

改进的 Elman 网络学习算法可归纳如下(这里网络结构采用 $r \times n \times m$, 不考虑 $x_{c,i}(k)$ 与 w_{ji}^1 之间的依赖关系):

$$\Delta w_{ij}^3 = \eta \delta_i^o x_j(k), i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad (2)$$

$$\Delta w_{jq}^2 = \eta \delta_j^h u_q(k), q = 1, 2, \dots, r; j = 1, 2, \dots, n \quad (3)$$

$$\Delta w_{jl}^1 = \eta \sum_{i=1}^m (\delta_i^o w_{ij}^3 \frac{\partial x_j(k)}{\partial w_{jl}^1}), j = l = 1, 2, \dots, n \quad (4)$$

$$\frac{\partial x_j(k)}{\partial w_{jl}^1} = f'_j(\cdot) x_i(k-1) + \alpha \frac{\partial x_j(k-1)}{\partial w_{jl}^1} \quad (5)$$

$$\delta_i^o = y_{d,i}(k) - y_i(k) g'_i(\cdot) \quad (6)$$

$$\delta_j^h = \sum_{i=1}^m (\delta_i^o w_{ij}^3) f'_j(\cdot) \quad (7)$$

式(6)中 $y_{d,i}(k)$, $y_i(k)$ 分别为系统期望输出和实际输出。

2 PID 递归神经网络控制器的设计

2.1 PID 递归神经网络控制器的结构及输出计算

PID 递归神经网络控制器的结构如图 2 所示, 为 $1 \times 3 \times 1$, 其输入是系统的误差 $e(k) = r(k) - y(k)$, 其中 $r(k)$ 是系统期望输出, $y(k)$ 是系统实际输出, $u(k)$ 是递归 PID 神经网络控制器在采样点 k 的输出, 即控制量。

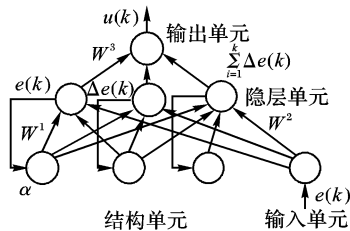


图2 PID 递归神经网络控制器结构示意图

若记隐层的输出为:

$$x(k) = [e(k) \quad \Delta e(k) \quad \sum_{i=1}^k e(i)]$$

结构单元层的输出为:

$$x_c(k) = [e(k-1) \quad \Delta e(k-1) \quad \sum_{i=1}^{k-1} e(i)]$$

则有以下非线性状态空间的表达式成立:

$$\begin{cases} x(k)^T = w^1 x_c(k-1)^T + w^2 e(k) \\ x_c(k) = x(k) \\ u(k) = w^3 x(k)^T \end{cases} \quad (8)$$

依据 PID 控制的特点, 对式(2)中的连接权矩阵作如下规定:

$$w^1 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, w^2 = [1 \quad 1 \quad 1]$$

$$w^3 = [K_p \quad K_d \quad K_i]$$

其中 w^1, w^2, w^3 分别是结构单元到隐层、输入层到隐层、以及隐层到输出层的连接权矩阵。

2.2 PID 递归神经网络控制器的学习算法

根据 PID 递归神经网络控制器的特点, 即 w^1, w^2 固定, w^3 可调, 所以只学习 w^3 , 这里递归神经网络控制器网络模型采用 $1 \times 3 \times 1$, 即 $u(k), y(k)$ 为单变量, 准则函数为:

$$E(k) = \frac{1}{2} [y_d(k) - y(k)]^2 \quad (9)$$

下面是基于系统 Elman 神经网络模型的 PID 递归神经网络控制器的学习算法:

$$\begin{aligned} \Delta K_p(k) &= \frac{\partial E(k+1)}{\partial K_p(k)} = \frac{\partial E(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial K_p(k)} \\ &= \frac{\partial E(k+1)}{\partial u(k)} e(k) \end{aligned} \quad (10)$$

$$\begin{aligned} \Delta K_d(k) &= \frac{\partial E(k+1)}{\partial K_d(k)} = \frac{\partial E(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial K_d(k)} \\ &= \frac{\partial E(k+1)}{\partial u(k)} \Delta e(k) \end{aligned} \quad (11)$$

$$\begin{aligned} \Delta K_i(k) &= \frac{\partial E(k+1)}{\partial K_i(k)} = \frac{\partial E(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial K_i(k)} \\ &= \frac{\partial E(k+1)}{\partial u(k)} \sum_{i=1}^k e(i) \end{aligned} \quad (12)$$

式(10)~(12)中, $\frac{\partial E(k+1)}{\partial u(k)}$ 由被控对象的 Elman 网络模型获得。

3 基于模型预测控制的平稳切换

被控对象所有的模型及相应控制器采用并行结构, 与实际系统的误差最小的模型才会被确定为参与实际控制的预测模型。

基于选中的 Elman 网络模型和相应 PID 神经网络控制器及系统实际输入输出, 优化出切换时的控制量 $\Delta u(k)$ 以使性能指标 J 最小, J 取二次型性能函数:

$$J = \frac{1}{2} \sum_{j=1}^M e^2(k+j) \quad (13)$$

其中 M 为预测步长, $e(k+j) = y_d(k+j) - \hat{y}(k+j)$, 可以由 $\frac{\partial J}{\partial \Delta u(k)} = 0$, 求得 $\Delta u(k)$ 。下面给出 $M=3$, 被控对象网络模型采用 $2 \times 6 \times 1$ 结构时, $\Delta u(k)$ 的优化算法, 参照图 3。

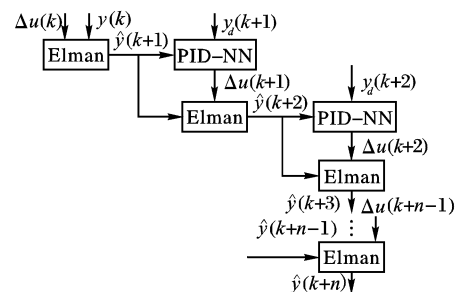


图3 当前控制信号产生以后各时刻预测输出路线示意图

1) 计算预测误差 $e(k+j)$, $j = 1, 2, 3$;

2) 计算 $\frac{\partial \hat{y}(k+1)}{\partial \Delta u(k)}$;

3) 计算:

$$\begin{aligned} \frac{\partial \hat{y}(k+2)}{\partial \Delta u(k)} &= \frac{\partial \hat{y}(k+2)}{\partial \Delta u(k+1)} \frac{\partial \Delta u(k+1)}{\partial \hat{y}(k+1)} \frac{\partial \hat{y}(k+1)}{\partial \Delta u(k)} + \\ &\quad \frac{\partial \hat{y}(k+2)}{\partial \hat{y}(k+1)} \frac{\partial \hat{y}(k+1)}{\partial \Delta u(k)} \end{aligned}$$

4) 计算:

$$\begin{aligned} \frac{\partial \hat{y}(k+3)}{\partial \Delta u(k)} &= \frac{\partial \hat{y}(k+3)}{\partial \Delta u(k+2)} \frac{\partial \Delta u(k+2)}{\partial \hat{y}(k+2)} \frac{\partial \hat{y}(k+2)}{\partial \Delta u(k)} + \\ &\quad \frac{\partial \hat{y}(k+3)}{\partial \hat{y}(k+2)} \frac{\partial \hat{y}(k+2)}{\partial \Delta u(k)} \end{aligned}$$

以上算法中 $\frac{\partial \hat{y}(k+1)}{\partial \Delta u(k)}$, $\frac{\partial \hat{y}(k+2)}{\partial \hat{y}(k+1)}$, $\frac{\partial \hat{y}(k+2)}{\partial \Delta u(k+1)}$,

$\frac{\partial \hat{y}(k+3)}{\partial \hat{y}(k+2)}$, $\frac{\partial \hat{y}(k+3)}{\partial \Delta u(k+2)}$ 可以通过被控对象的 Elman 网络求得,

$\frac{\partial \Delta u(k+2)}{\partial \hat{y}(k+2)}$, $\frac{\partial \Delta u(k+1)}{\partial \hat{y}(k+1)}$ 可以通过 PID 神经网络控制器求得。

5) 令: $\frac{\partial J}{\partial \Delta u(k)} = \sum_{j=1}^3 e(k+j) \frac{\partial \hat{y}(k+j)}{\partial \Delta u(k)} = 0$; 获得 $\Delta u(k)$ 。

将此控制量 $\Delta u(k)$ 和相应 PID 控制器同时切入实际系统, 就能改善系统切换过程的动态性能。

4 仿真示例

用提出的模型切换策略对以下两个模型的进行仿真:

模型 1: $\frac{2}{s^2 + 0.7s + 4}$, 相应的控制器为 PID1

模型 2: $\frac{2}{s^2 + 0.7s + 2}$, 相应的控制器为 PID2

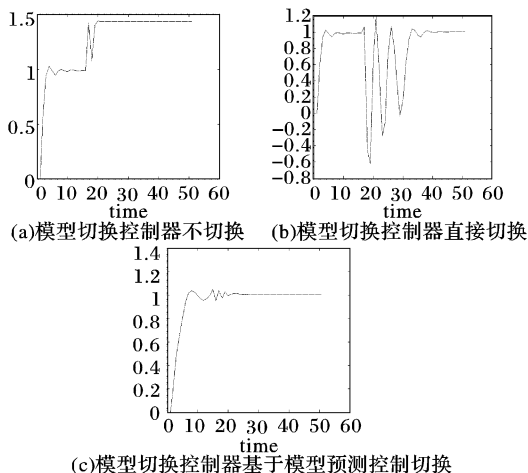


图4 实验结果

系统在加单位阶跃输入信号后的第 15 s 时, 模型 1 切换到模型 2, 图 4(a)、(b)、(c) 分别为控制器不切换, 直接切换和基于模型预测控制的切换的仿真结果。从仿真结果看, 不切换, 系统最终输出不能满足控制要求; 直接切换, 虽然系统最终输出能满足控制要求, 但切换过程动态性能很差; 基于模型预测控制切换, 不仅最终输出能满足控制要求, 而且切换过程动态性能得到了有效地改善, 切换过程的超调量在 $\pm 5\%$ 以

内, 切换过程的调节时间由直接切换的 17 s 缩短到 3 s。

5 结语

当被控对象的结构或参数随生产条件的变化而变化时, 该过程控制模型可视为多模型系统, 多模型自适应控制策略是控制这些过程的常用方式, 但实施多模控制时, 控制器时常需要在已建立控制器模型集中的控制器间切换, 这种切换会引起系统的瞬态响应, 而本文提出的基于模型预测的平稳切换策略, 有效改善了切换过程的系统动态性能, 并通过仿真表明了这种切换策略的有效性。

参考文献:

- [1] NARENDRA KS, BALAKRISHNAN J. Performance improvement in adaptive control systems using multiple models and switching[A]. In: Proc. of Seventh Yale Wrkshp. Adaptive Learning Syst[C]. New Haven: CT, 1992. 27-33.
- [2] NARENDRA KS, BALAKRISHNAN J. Improving Transient Response of Adaptive Control Systems using Multiple Models and Switching[J]. IEEE Trans. on Automatic Control. 1994, 39(9): 1861-1866.
- [3] NARENDRA KS, BALAKRISHNAN J. Adaptive Control Using Multiple Models and Switching [J]. IEEE Trans. on Automatic Control. 1997, 42(2): 171-187.
- [4] RECHALET J, et al. Model Predictive heuristic Control; Applications to Industry Processes [J]. Automatica. 1978, 17(5): 413-428.
- [5] 张志君, 于海晨. 基于神经网络的广义预测控制及在化学反应器中的应用[A]. 第二十三届中国控制会议论文集[C]. 上海: 华东理工大学出版社, 2004. 896-899.
- [6] 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. 199-200.
- [7] 宋宜斌, 王培进. 预测控制中的神经网络模型及其应用研究 [A]. 第二十三届中国控制会议论文集[C]. 上海: 华东理工大学出版社, 2004. 687-790.

(上接第 1397 页)

表2 结果正确率比较

类别	词频	互信息	T 检验	χ^2 统计	汉字关联度
环境	0.69	0.26	0.23	0.28	0.70
计算机	1	0.95	1	1	1
交通	0.48	0.69	0.62	0.45	0.83
教育	0.98	1	1	0.95	1
经济	0.59	0.7	0.7	0.3	0.73
军事	0.55	0.53	0.53	0.55	0.52
体育	0.75	0.89	0.89	0.34	0.86
医药	0.59	0.7	0.7	0.75	0.8
艺术	0.42	0.51	0.51	0.3	0.56
政治	0.48	0.65	0.65	0.25	0.95

从表 2 可以看出, 本文提出的汉字关联度 (CCAM) 在大多数类别的相似性分析与聚类中优于其他统计量。

文献[4]曾经分别使用基于 bi-gram、基于关键词、基于字词混合等三种方式进行文本索引、查询和检索的试验, 在 TREC-5 测试集上, 其结果正确率分别为 36.8%、33.2% 和 36.8%; 在 TREC-6 测试集上, 其结果正确率分别为 56.5%、47.1% 和 49.0%。试验结果表明基于 bi-gram 方式的正确率最高。由于本试验测试结果中基于汉字关联度的结果正确率

高于基于 bi-gram 词频的结果正确率, 因此, 说明了该算法的结果优于现有的基于关键词的同类算法的结果。

4 结语

本文通过构建文本的汉字关联度矩阵并以此设计文本相似度算法进行文本相似性分析, 为解决传统的基于“特征词”的相关技术方法引起的高计算费用问题探索了一种新的研究思路。该文提出的基于汉字与汉字之间关系的文本相似度算法用 Java 实现, 在 Linux 平台上通过了大规模语料的测试与应用。

参考文献:

- [1] 李庆虎, 陈玉健, 孙家广. 一种中文分词词典新机制—双字哈希机制[J]. 中文信息学报, 2003(4): 13-18.
- [2] MANNING D. Foundations of Statistical Natural Language Processing[M]. Publishing House of Electronics Industry. 2005.
- [3] 孙茂松, 肖明, 邹嘉彦. 基于无指导学习策略的无词表条件下的汉语自动分词[J]. 计算机学报, 2004(6): 736-742.
- [4] LEONG M, ZHOU H. Preliminary qualitative analysis of segmentation vs bi-gram indexing in Chinese[A]. In TREC-6, Gaithersburg [C]. Maryland, 1997. 551-558.