

文章编号: 1671-7848(2014)01-0111-05

一种自构建神经网络的内模控制方法

赵志诚^{1,2}, 王元元², 刘志远¹, 张井岗²

(1. 哈尔滨工业大学 控制科学与工程系, 黑龙江 哈尔滨 150001; 2. 太原科技大学 电子信息工程学院, 山西 太原 030024)



摘 要: 针对非线性过程, 提出了一种基于自构建神经网络的内模控制方法(Internal Model Control, IMC)。采用自构建算法实现神经网络的结构学习和参数学习, 在被控过程内部模型和控制器模型的辨识过程中, 该网络能够根据给定的判定条件自动增加神经元节点, 以满足辨识精度的要求; 为了防止网络学习过拟合, 基于灵敏度方法对神经网络隐层节点进行修剪删除; 网络的参数学习采用梯度下降法。自构建算法可以有效地避免普通神经网络内模控制方案中网络结构难以确定的问题, 仿真结果表明, 该控制系统有良好的跟踪性、鲁棒性和抗干扰性。

关键词: 自构建; 神经网络; 灵敏度; 内模控制

中图分类号: TP 273

文献标志码: A

An Internal Model Control Method Based on Self-Constructing Neural Network

ZHAO Zhi-cheng^{1,2}, WANG Yuan-yuan², LIU Zhi-yuan¹, ZHANG Jing-gang²

(1. Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, China;

2. School of Electronic Information Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China)

Abstract: A novel algorithm on internal model control (IMC) based on self-constructing neural network (NN) is proposed for the non-linear process in this paper. The structure learning and parameters learning of the neural network were realized by self-constructing algorithm. In the identification process of the internal model and the controller, the neural network can automatically increase the nodes to meet the identification accuracy requirements. Moreover, in order to prevent the over-fitting of neural network learning, the hidden layer nodes can be pruned based on the sensitivity method. In addition, parameters learning adopt the gradient descending method. Compared with conventional NN-IMC method, this scheme can effectively avoid the problem of network structure is difficult to determine. The simulation result shows that the control system has a good target tracking performance, disturbance rejection properties and robustness simultaneously.

Key words: self-constructing; neural network; sensitivity; internal model control

1 引言

内模控制是一种基于过程数学模型的控制方法, 过程模型与逆模型的获取是内模控制设计的关键。由于实际工业过程往往存在非线性特性, 难以获得精确的数学模型。神经网络因其具有任意逼近非线性的能力而成为了一种有效的建模方法, 且与内模控制相结合, 进一步丰富了非线性系统的控制方法^[1-3]。然而神经网络建模仍存在许多不足之处, 如网络结构难以确定, 复杂的网络结构将使学习收敛速度变得缓慢, 归纳性变差; 过简的结构又易造成学习能力降低, 这些不足最终会影响控制系统的性能。

针对神经网络结构难以确定的不足, 一些学者提出了关于 BP 神经网络^[4]、小波神经网络^[5]、模

糊神经网络^[6-9]、模糊小脑神经网络^[10]的结构改进方法, 采用动态变结构的思想, 有效地改善了网络结构和系统辨识、控制性能。

本文针对非线性过程提出一种基于自构建神经网络的内模控制方法, 利用自构建神经网络建立和自适应调整非线性过程模型和控制器模型, 实现了辨识网络和控制器网络结构的在线调整。理论分析和仿真结果表明, 所提出的方法具有快速跟踪输入、无超调等良好特性, 并且能在系统受到干扰或被控过程参数发生变化时, 仍然具有较好的自适应性和鲁棒稳定性。

2 自构建神经网络原理

2.1 增加神经元的数学描述

神经网络隐层节点的连接权表示该节点的连接

收稿日期: 2012-10-22; 收修定稿日期: 2012-11-20

基金项目: 山西省自然科学基金资助项目(2012011027-4)

作者简介: 赵志诚(1970-), 男, 山西临猗人, 教授, 主要从事先进控制技术等方面的教学与科研工作。

强度, 权值为正表示激励, 权值为负表示抑制。隐层第 r 个节点对神经网络的作用强度可表示为

$$u_r = \sum_{i=1}^{m_1} w_{ri} + \sum_{j=1}^{m_3} v_{rj} \quad (1)$$

式中: w_{ri} 为隐层第 r 个节点对第 i 个输入的连接权; v_{rj} 为隐层第 r 个节点对第 j 个输出的连接权; m_1 为输入层节点个数; m_3 为输出层节点个数。神经网络隐层节点的作用强度拟概率表示为

$$p_r = u_r^2 / \sum_{i=1}^{m_2} u_i^2 \quad (2)$$

式中: p_r 为隐层第 r 个节点作用强度的拟概率; m_2 为隐层节点个数。增加节点判定条件:

$$\begin{cases} p > G_{th}; \\ |e| > e_0 \end{cases} \quad (3)$$

式中: $p = \min(p_r)$; G_{th} 为拟概率阈值; e_0 为不同训练阶段的期望精度且 $e_0 = \mu^n$, $\mu \in (0, 1)$ 为收敛常数, 本文中 μ 取 0.5; n 为训练次数; e 为神经网络期望输出与实际输出之差。式(3)表示的意义为, 当所有神经元的作用强度都比较大且网络收敛精度低时, 说明网络欠拟合需要增加新的神经元。此时, 将激励强度最大的神经元 r 分化为 2 个神经元。新神经元 r' 的权值定义如下:

$$v_{r'j}(n+1) = v_{rj}(n), \quad (4)$$

$$w_{r'i}(n+1) = \alpha w_{ri}(n) \quad (5)$$

式中, α 为小于 1 的正常数。神经元 r 的权值如下:

$$v_{rj}(n+1) = v_{rj}(n), \quad (6)$$

$$w_{ri}(n+1) = (1 - \alpha)w_{ri}(n) \quad (7)$$

2.2 修剪神经元的数学描述

如果来自某节点的轻微扰动使得网络的输出产生很大的变化, 说明网络输出对该节点的变化非常敏感, 即该节点对网络起着不可忽视的作用, 反之则影响较小, 该节点可以删除。灵敏度算法就是通过计算网络各节点权值对网络误差函数的灵敏度, 即该节点权值对误差函数的重要程度, 来删除那些灵敏度值较小的节点。

Mozer 和 Skelensky 对神经网络的节点引入一个重视程度系数^[11], 假定第 r 个节点的重视程度系数为 α_r , 则 $\alpha_r = 0$ 相当于删除该节点, $\alpha_r = 1$ 表示该节点为一传统节点, 单输出神经网络的输入输出关系可表示为

$$o_r = f\left(\sum_i w_{ri}x_i\right); y = \sum_r v_r o_r \alpha_r$$

式中: x_i 为第 i 个输入; w_{ri} 为第 i 个输入与第 r 个隐节点的连接权; o_r 为第 r 个隐节点的输出值; v_r 为第 r 个隐节点与输出节点的连接权; f 为 sigmoid 激活函数。

节点灵敏度的定义为

$$\rho_r = E_{\alpha_r=0} - E_{\alpha_r=1}$$

式中: $E_{\alpha_r=0}$ 为该节点不存在时系统目标函数; $E_{\alpha_r=1}$ 为该节点存在时系统目标函数。

利用目标函数对 α_r 的导数信息来逼近 ρ_r 即:

$$\lim_{\gamma \rightarrow 1} \frac{E_{\alpha_r=\gamma} - E_{\alpha_r=1}}{\gamma - 1} = \left. \frac{\partial E}{\partial \alpha_r} \right|_{\alpha_r=1}, \text{ 对上式进行工程近似, 即}$$

$$\text{当 } \gamma = 0 \text{ 时, 上式依然成立, } \frac{E_{\alpha_r=0} - E_{\alpha_r=1}}{-1} \approx \left. \frac{\partial E}{\partial \alpha_r} \right|_{\alpha_r=1},$$

所以得到灵敏度的逼近表达式 $\rho_r \approx - \left. \frac{\partial E}{\partial \alpha_r} \right|_{\alpha_r=1}$, 计算

ρ_r 时, $\alpha_r = 1$ 。当隐层某节点的灵敏度小于灵敏度阈值 M_{th} 时, 表明此节点对网络的影响较小, 可以忽略, 故删除该节点。

3 自构建神经网络内模控制

3.1 神经网络内模控制

在模型可逆的前提下, 神经网络内模控制系统结构如图 1 所示, 系统的内部模型和控制器均由神经网络实现。

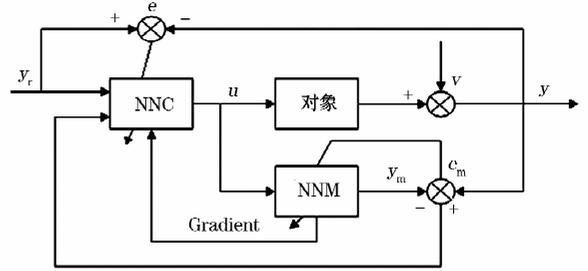


图 1 神经网络内模控制

Fig. 1 Neural network internal model control

图中: NNC 为神经网络内模控制器, u 为控制信号, NNM 为神经网络内部模型, y 为被控过程的输出, y_m 为模型的输出, $e_m = y - y_m$ 为过程输出与模型输出之差, y_r 为参考输入, $e = y_r - y$ 为参考输入与过程输出之差, v 为系统的不可测干扰。神经网络内模控制的关键在于辨识被控过程的模型 NNM 和逆模型 NNC, 可以在控制系统运行时在线建立 NNM 和 NNC, 也可以预先采样系统的输入输出数据, 建立 NNM 和 NNC, 再投入控制系统运行。在建立 NNC 和 NNM 时, 根据 e_m 来训练 NNM 的可调参数, 根据 e 和经 NNM 通道反传的梯度信号来训练 NNC 的可调参数。

3.2 过程模型的建立

单输入单输出非线性被控过程可以用如下的离散时间非线性系统表示:

$$y(k+1) = f\{y(k), \dots, y[k-(n-1)], u(k), \dots, u[k-(m-1)]\} + v(k) \quad (8)$$

式中: $f(\cdot)$ 为非线性函数; $y(k)$, $u(k)$ 和 $v(k)$ 分别为过程的输出、输入和扰动; n , m 为输出和输入的滞后时间。用神经网络建模时, 将过程的输入和

输出的各次滞后作为网络的输入, 通过网络的非线性映射可得模型的输出。

采用三层结构的 BP 网络实现式(8)所表示的非线性函数 $f(\cdot)$, 网络输入层和隐含层的神经元个数分别用 m_1 和 m_2 表示, 各层的净输入用 net 表示, 输出用 o 表示, o_{ij} 表示第 i 层的第 j 个神经元的输出, 则过程模型建立如下:

输入层描述:

$$o_{ij}(k) = \begin{cases} y(k-j), & 0 \leq j \leq n-1; \\ u(k-j+n), & n \leq j \leq n+m-1 \end{cases} \quad (9)$$

隐含层描述:

$$\text{net}_{2l}(k) = \sum_{j=1}^{m_1} w_{lj} o_{ij}(k) - \theta_{2l}, \quad l = 1, 2, \dots, m_2 \quad (10)$$

$$o_{2l}(k) = g[\text{net}_{2l}(k)], \quad l = 1, 2, \dots, m_2 \quad (11)$$

输出层描述:

$$\text{net}_3(k) = \sum_{l=1}^{m_2} v_l o_{2l}(k) - \theta_3, \quad (12)$$

$$y_m(k+1) = \text{net}_3(k) \quad (13)$$

式中: 神经元的激活函数 $g(\text{net}) = \frac{1}{1 + e^{-\text{net}}}$, w_{lj} 和 v_l

分别为输入层与隐层和隐层与输出层之间的连接权值; θ_{2l} 和 θ_3 为隐层和输出层神经元的阈值。

网络学习的目标函数为

$$E_{\text{NNM}}(k) = \frac{1}{2} [y(k) - y_m(k)]^2 \quad (14)$$

则该网络隐层第 h 个节点的灵敏度为

$$\rho_h = [y(k) - y_m(k)] v_h(k) o_{2h}(k) \quad (15)$$

用梯度下降法调整参数, 学习算法如下:

$$\Delta v_l(k+1) = -\eta \frac{\partial E_{\text{NNM}}}{\partial y_m} \frac{\partial y_m}{\partial \text{net}_3} \frac{\partial \text{net}_3}{\partial v_l} = \eta [y(k) - y_m(k)] o_{2l}(k) \quad (16)$$

$$\Delta w_{lj}(k+1) = -\eta \frac{\partial E_{\text{NNM}}}{\partial y_m} \frac{\partial y_m}{\partial \text{net}_3} \frac{\partial \text{net}_3}{\partial o_{2l}} \frac{\partial o_{2l}}{\partial \text{net}_{2l}} \frac{\partial \text{net}_{2l}}{\partial w_{lj}} = \eta [y(k) - y_m(k)] v_l(k) g'[\text{net}_{2l}(k)] o_{ij}(k) \quad (17)$$

$$\Delta \theta_3(k+1) = -\eta [y(k) - y_m(k)] \quad (18)$$

$$\Delta \theta_{2l}(k+1) = -\eta [y(k) - y_m(k)] v_l(k) g'[\text{net}_{2l}(k)] \quad (19)$$

自构建神经网络的学习算法如下:

①初始化网络;

②输入训练样本并计算网络各层的输出;

③根据式(1), 式(2), 式(3)计算节点的权值拟概率及当前训练的期望误差;

④根据式(3)判断是否增加隐层节点, 若增加则将对对应节点进行分化, 新增节点的权值根据式(4)~(7)得出, 初始阈值为0, 否则至⑤;

⑤根据式(15)计算隐层节点灵敏度, 判断各节点灵敏度是否小于灵敏度阈值, 若小于则删除对应节点并返回②, 否则至⑥;

⑥若网络结构无变化, 则用梯度下降法修正参数。如果完成了设定的训练次数, 返回②, 否则结束训练。

3.3 过程逆模型的建立

将被控过程的逆动态模型用离散时间非线性系统表示为

$$u(k) = f^{-1}\{y(k+1), y(k), \dots, y[k-(n-1)], u(k-1), \dots, u[k-(m-1)]\} \quad (20)$$

在控制系统中总是希望输出无偏差跟踪参考输入, 因此 $y(k) = y_r(k)$, 式(20)可表示为

$$u(k) = f^{-1}\{y_r(k+1), y_r(k), \dots, y_r[k-(n-1)], u(k-1), \dots, u[k-(m-1)]\} \quad (21)$$

式中, $y_r(k+1)$ 为下一采样时刻系统的参考输入值。一般通过对当前及以前时刻参考输入采样值的线性化预测可以得到 $y_r(k+1)$ 的预测值, 如采用 k 时刻和 $k-1$ 时刻的采样值, 则得到 $y_r(k+1) = 2y_r(k) - y_r(k-1)$ 。因此, 式(21)可表示为

$$u(k) = f^{-1}\{[2y_r(k) - y_r(k-1)], y_r(k), \dots, y_r[k-(n-1)], u(k-1), \dots, u[k-(m-1)]\} \quad (22)$$

在内模控制系统中, 还需加入反馈量 $e_m(k) = y(k) - y_m(k)$ 作为过程逆模型的输入, 所以神经网络内模控制器的输入输出关系可用下式表示:

$$u(k) = \psi\{y_r(k), \dots, y_r[k-(n-1)], u(k-1), \dots, u[k-(m-1)], e_m(k)\} \quad (23)$$

同样用三层 BP 神经网络来实现式(23)所示的非线性函数 $\psi(\cdot)$, 设输入层、隐层的神经元个数分别为 n_1 和 n_2 。

输入层描述:

$$o_{ij}(k) = \begin{cases} y_r(k-j), & 0 \leq j \leq n-1; \\ u(k-j+n-1), & n \leq j \leq n+m-2 \\ e_m(k) \end{cases} \quad (24)$$

隐含层描述:

$$\text{net}_{2l}(k) = \sum_{j=1}^{n_1} w_{lj} o_{ij}(k) - \theta_{2l}, \quad l = 1, 2, \dots, n_2 \quad (25)$$

$$o_{2l}(k) = g[\text{net}_{2l}(k)], \quad l = 1, 2, \dots, n_2 \quad (26)$$

输出层描述:

$$\text{net}_3(k) = \sum_{l=1}^{n_2} v_l o_{2l}(k) - \theta_3 \quad (27)$$

$$u(k) = \text{net}_3(k) \quad (28)$$

网络学习的目标函数为

$$E_{\text{NNC}}(k) = \frac{1}{2} [y_r(k) - y(k)]^2 \quad (29)$$

该网络隐层第 h 个节点的灵敏度为

$$\rho_h = [y_r(k) - y(k)] \frac{\partial y}{\partial u} v_h(k) o_{2h}(k) \quad (30)$$

神经元激活函数与模型网络中的相同, 神经元学习率为 η 。参数修正算法如下:

$$\Delta v_l(k+1) = -\eta \frac{\partial E_{\text{NNC}}}{\partial y} \cdot \frac{\partial y}{\partial u} \cdot \frac{\partial u}{\partial \text{net}_3} \cdot \frac{\partial \text{net}_3}{\partial v_l} = \eta [y_r(k) - y(k)] \frac{\partial y}{\partial u} o_{2l}(k) \quad (31)$$

$$\Delta w_{lj}(k+1) = -\eta \frac{\partial E_{\text{NNC}}}{\partial y} \cdot \frac{\partial y}{\partial u} \cdot \frac{\partial u}{\partial \text{net}_3} \cdot \frac{\partial \text{net}_3}{\partial o_{2l}} \cdot \frac{\partial o_{2l}}{\partial \text{net}_{2l}} \cdot \frac{\partial \text{net}_{2l}}{\partial w_{lj}} =$$

$$\eta [y_r(k) - y(k)] \frac{\partial y}{\partial u} v_l(k) g'[\text{net}_{2l}(k)] o_{lj}(k) \quad (32)$$

$$\Delta \theta_3(k+1) = -\eta [y_r(k) - y(k)] \frac{\partial y}{\partial u} \quad (33)$$

$$\Delta \theta_{2l}(k+1) = -\eta [y_r(k) - y(k)] \frac{\partial y}{\partial u} v_l(k) \cdot g'[\text{net}_{2l}(k)] \quad (34)$$

对于非线性模型来说, $\frac{\partial y(k)}{\partial u(k)}$ 不容易求得, 且经过有限次训练后 $y_m(k)$ 能够以一定精度逼近 $y(k)$, 所以用 $\frac{\partial y_m(k)}{\partial u(k)}$ 近似代替 $\frac{\partial y(k)}{\partial u(k)}$ 。

4 仿真及结果分析

为验证本算法的有效性, 考察式(35)所示非线性被控过程, 依本文方法采用在线训练方式进行过程建模和控制器设计, 并与常规神经网络方法的控制结果进行比较。

$$y(k+1) = \frac{ay(k) + bu(k)}{1 + y^2(k)} + u^3(k) \quad (35)$$

式中: 取 $a = 1.0, b = 1.2$ 。

仿真中常规神经网络隐层神经元个数根据经验公式 $q = \sqrt{r+c} + d$ 选取, 其中: r, c 分别为神经网络的输入、输出个数, 通过试凑取 $d = 4$ 。为了体现神经网络的自构建特性, 本文方法初始时取 $d = 0$ 。

在网络的训练过程中, 拟概率阈值 G_{th} 和灵敏度阈值 S_{th} 的取值将直接影响网络的训练效果, 经过仿真测试可得出 NNM 的 $G_{\text{th}} = [0.25, 0.35]$, $S_{\text{th}}[0.05e_0, 0.25e_0]$, NNC 的 $G_{\text{th}} = [0.35, 0.5]$, $S_{\text{th}}[0.05e_0, 0.3e_0]$ 。当 NNM 的 $G_{\text{th}} = 0.28$, $S_{\text{th}} = 0.25e_0$, NNC 的 $G_{\text{th}} = 0.35, S_{\text{th}} = 0.12e_0$, 隐层神经元个数的变化如图 2 所示。可见网络隐层神经元的个数能根据判定条件逐步调整, 当辨识精度达到要求时, 网络结构也随之稳定。

当网络训练的拟概率阈值 G_{th} 和灵敏度阈值 S_{th} 在不同范围取值时, 系统的单位阶跃响应如图 3 所示。

由图可见, 当网络训练的相关参数取值在设定范围时, 系统能合理调节隐层神经元个数来保证控制性能。当相关参数取值大于或小于设定范围时均

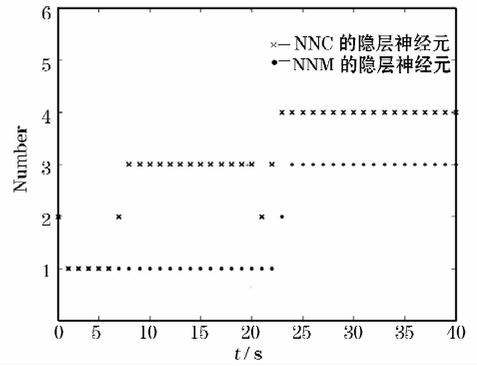


图 2 训练过程中神经网络隐层神经元数

Fig. 2 Number of hidden neuron in the training process

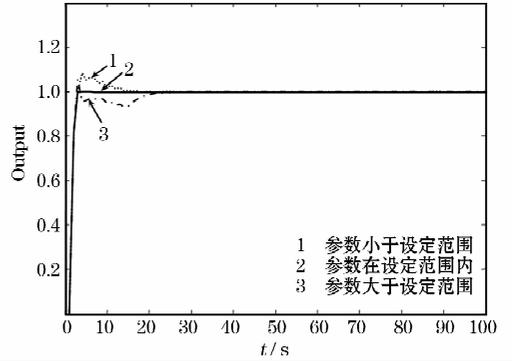


图 3 不同参数下系统的单位阶跃响应

Fig. 3 Step response of the system in the different parameters

会使网络结构变差, 最终影响系统的控制性能。

另外, 与常规的神经网络相比, 相同训练次数时的跟踪误差如表 1 所示, 可见自构建方法具有较好的收敛性。

表 1 两种方法的跟踪误差比较

Table 1 Comparison of tracking errors of the two methods

训练次数	常规方法跟踪误差	本文方法跟踪误差
5	-1.2×10^{-1}	6.4×10^{-4}
10	-4.3×10^{-2}	9.8×10^{-5}
15	-1.8×10^{-2}	6.5×10^{-8}
20	-8.2×10^{-3}	1.6×10^{-8}

常规神经网络内模控制与本文方法的阶跃响应结果如图 4 所示。

在采样点 $k = 50$ 时加入幅值为 0.1 的阶跃扰动 $v(k)$, 结果显示本文方法的响应速度更快, 超调量小, 抗干扰能力更好。

为验证控制系统的鲁棒性, 在采样点 $k = 50$ 时改变被控过程参数 $a = 0.8, b = 1.0$, 其仿真结果如图 5 所示。结果表明, 本文方法有更强的跟踪性和鲁棒性。

表 2 给出了两种方法的控制系统性能指标, 其中 IAE 为偏差绝对值积分, $\sigma\%$ 为超调量, k 为扰动后恢复稳态的训练次数, 较小的指标意味着较好的性能。由表 2 可见, 本文方法有更好的控制性能。

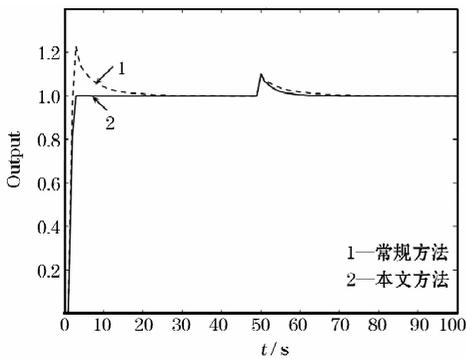


图 4 系统的单位阶跃响应

Fig. 4 Step response of the system

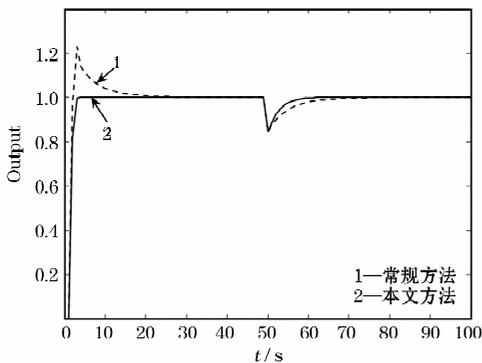


图 5 参数摄动时系统的阶跃响应

Fig. 5 Step response of the system with parameter perturbation

表 2 两种方法中控制系统的性能指标

Table 2 Performance indicators of the two methods

方法	标称情况			模型失配		
	$\sigma/\%$	IAE	k	$\sigma/\%$	IAE	k
常规	22.5	2.7	20	22.5	2.9	20
本文	0	1.5	10	0	1.7	10

5 结 论

针对复杂的非线性被控过程，本文提出了一种基于自构建神经网络的内模控制方法。实现了网络结构和参数同时在线学习，使得辨识网络和控制

网络的结构可以根据需要在线变化，既不会欠拟合也不会过拟合。理论分析和仿真结果表明，所提出的方法具有快速跟踪输入、无超调等良好特性，并且能在系统受到干扰或被控过程参数发生变化时，仍然具有良好的自适应性和鲁棒稳定性。

参考文献 (References) :

- [1] Haber R E, Alique J R. Nonlinear internal model control using neural networks: an application for machining processes [J]. Neural Comput & Applic, 2004, 13(1): 47-55.
- [2] 张伟, 李大字. 改进的一般化学习网络内模控制在 CSTR 中的应用[J]. 北京化工大学学报: 自然科学版, 2009, 36(4): 100-104. (Zhang Wei, Li Dazi. Application of an improved universal learning network internal model control to a continuous stirred tank reactor system [J]. Journal of Beijing University of Chemical Technology: Natural Science, 2009, 36(4): 100-104)
- [3] 陈高华, 张井岗, 赵志诚. 基于模糊神经网络的二自由度内模控制[J]. 电气自动化, 2011, 33(1): 1-3. (Chen Gaohua, Zhang Jinggang, Zhao Zhicheng. A two-degrees-of-freedom internal model control based on fuzzy neural network [J]. Electrical Automation, 2011, 33(1): 1-3.)
- [4] Li H X, Deng H. An approximate internal model-based neural control for unknown nonlinear discrete processes [J]. IEEE Transactions on Neural Networks, 2006, 17(5): 659-670.
- [5] Hsu C F. Adaptive growing-and-pruning neural network control for a linear piezoelectric ceramic motor [J]. Engineering Applications of Artificial Intelligence, 2008, 21(8): 1153-1163.
- [6] Wu C F, Lin C J, Lee C Y. A functional neural fuzzy network for classification applications [J]. Expert Systems with Applications, 2010, 49(11): 1-7.
- [7] Lin C J. Nonlinear systems control using self-constructing wavelet networks [J]. Applied Soft Computing, 2009, 9(1): 71-79.
- [8] Cheng K H. Auto- structuring fuzzy neural system for intelligent control [J]. Journal of the Franklin Institute, 2009, 346(3): 267-288.
- [9] Phan P A, Gale T J. Direct adaptive fuzzy control with a self-structuring algorithm [J]. Fuzzy Sets and Systems, 2008, 159(8): 871-899.
- [10] Lee C Y, Lin C J, Chen H J. A self-constructing fuzzy CMAC model and its applications [J]. Information Sciences, 2007, 177(1): 264-280.
- [11] Rementeria S. Dynamic schedule for effective on-line connection pruning [J]. Neural Processing Letters, 2001, 14(1): 1-14.