

混沌粒子群优化的模糊神经PID控制器设计

赵俊, 陈建军

(西安电子科技大学机电工程学院, 陕西 西安 710071)

摘要: 针对常规PID控制的线性局限性及传统模糊控制和模糊PID控制中积分误差规则难以获取, 系统存在稳态误差的问题, 提出一类以模糊神经网络和PID神经网络组成的模糊神经PID控制器. 以整个神经网络的权值为优化参数, 利用基于混沌策略的粒子群全局优化算法离线优化和误差反传算法在线调整相结合的方法获得控制器参数, 并设计了混沌优化与粒子群结合的两步方案. 仿真结果表明: 与传统PID、模糊、模糊PID控制相比, 系统的瞬态和稳态性能有了明显提高, 且保持了一定的鲁棒性及高跟踪精度. 该方法有效地拓展了PID控制的使用范围, 并为智能方法与PID控制的结合提供了一种新的参考方案.

关键词: 稳态误差; PID神经网络; 模糊神经PID控制器; 混沌优化; 粒子群全局优化

中图分类号: TP273⁺.2 **文献标识码:** A **文章编号:** 1001-2400(2008)01-0054-06

Design of the fuzzy neural PID controller based on hybrid PSO

ZHAO Jun, CHEN Jian-jun

(School of Mechano-electronic Engineering, Xidian Univ., Xi'an 710071, China)

Abstract: Considering the problems of the linearity limit of PID control and the steady-state error in fuzzy, fuzzy PID control for it cannot easily obtain the control rules of the integral error, so a fuzzy neural PID controller which consists of a fuzzy neural network and a PID neural network is designed. The parameters of the controller are optimized by the mixed learning methods integrating the offline particle swarm optimization algorithm combined with chaos strategies of global searching ability, with the online BP algorithm of local searching ability. Simulation results show that the designed novel controller and the proposed optimization algorithm have obviously improved the performance of the transient state and steady state in the control processing. Compared with conventional PID, fuzzy control and fuzzy-PID control method, the new controller with the optimization method has good robustness and better performance. The new method breaks through the limit of linearity of PID control and expands its applications. It also provides a new reference for the combination of intelligent method and PID method.

Key Words: steady-state error; PID neural network; fuzzy neural PID controller; chaos optimization; particle swarm optimization

常规PID控制是目前工业过程控制中应用最为广泛的策略之一, 其控制结构简单、参数易于在线调整, 主要适用于线性系统控制. 而工业环境中被控对象往往机理复杂, 具有高度非线性、时变不确定性和纯滞后等特点, 常规PID控制器存在参数整定不良、工况适应性差等问题. 为此人们将模糊控制与PID控制结合, 利用模糊控制理论的非线性处理能力, 较好地解决了非线性系统的控制问题^[1,2]. 然而一般模糊控制器均表现为PD控制器的性能, 难以获得积分误差的规则, 故系统存在稳态误差. 且仅靠经验规则进行整定, 控制参数无法达到最优, 与PID结合后性能仍有待提高.

针对上述问题, 笔者提出一种由模糊神经网络(FNN)和PID神经网络(PIDNN)组成的模糊神经PID控制器(FNPIDC), 控制器待调整参数为神经网络权系. 利用FNN部分既能实现原有模糊控制器完成的模

收稿日期: 2007-03-29

基金项目: 国家部委预研基金资助(51421060505DZ0155); 陕西省自然科学基金资助(2005A009)

作者简介: 赵俊(1982-), 男, 西安电子科技大学博士研究生, E-mail: SEKEL@sohu.com.

糊推理,又能实时调整隶属函数,使控制器具备更强的学习和自适应能力;引入 PIDNN 能更好地改善系统的瞬态和稳态性能,获得更精确的跟踪效果.为获得更为接近最优解的控制参数值,笔者提出一种离线混沌粒子群优化和在线误差反传相结合的方法.即在标准粒子群算法(PSO)的基础上,通过结合两步混沌优化策略,在全局搜索范围内微调、局部范围内细搜索,使算法在初期尽快收敛的情况下,后期能够跳出局部极小的限制.离线取得近似最优解的参数,在线时利用误差反传(BP)快速进行调整,以获得更好的控制性能.通过在 Matlab 平台下的仿真实验,验证了控制器及其算法的有效性,取得了良好的控制效果.

1 PID、模糊、模糊 PID 控制原理

对于常规的 PID 控制器,其控制律为

$$U = K_P e(t) + K_I T \sum_{i=0}^t e(i) + (K_D/T) \Delta e(t) \quad (1)$$

其中 $e(t) = y_d(t) - y(t)$ 为误差,它是控制器的输入; $y_d(t)$ 为期望信号, $y(t)$ 为系统实际输出; K_P, K_I, K_D 分别为比例、积分、微分增益; T 为采样周期; $\Delta e(t) = e(t) - e(t-1)$ 为误差变化;而 $K_I = K_P/T_I, K_D = K_P \cdot T_D, T_I, T_D$ 分别为积分和微分时间常数.只要比例系数、积分时间、微分时间常数 3 个参数整定恰当,就可以避免调节过程过分震荡(P 作用),又能实现无差控制(I 作用),而且具有超调作用,有效克服动态误差和缩短调节过程时间(D 作用).

传统模糊控制器为二维,即误差 e 和误差变化 e_c 作为控制器的两个输入量.先对其进行模糊量化处理,得到模糊变量 E 和 E_c ,按模糊控制规则进行模糊推理,再由解模糊得到模糊控制量 U .最后经比例变换得到实际控制量 U_f .根据专家经验制定规则表后,模糊控制器对复杂和不确定系统能进行简单有效的控制.位置式的模糊控制器近似等价于一个位置式 PD 控制器,由于它不具有积分环节,故很难消除稳态误差^[3].

模糊 PID 控制器主要有两种,一是通过模糊规则直接调整 PID 参数,二是调整系统控制量.前者通过一套 e, e_c 和 K_p, K_i, K_d 的规则库直接调整参数;后者则是在大偏差范围内利用模糊推理的方法调整系统的控制量 U ,而在小偏差范围内转换成 PID 控制,两者的转换根据给定的偏差范围自动实现.图 1 为一典型调整系统控制量的模糊 PID 部分控制原理图.

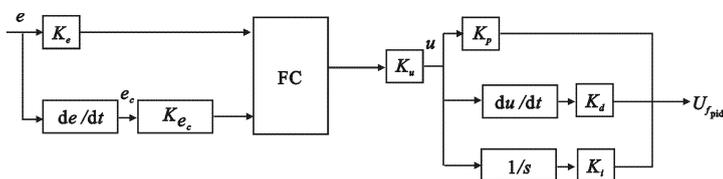


图 1 模糊 PID 控制原理图

与常规 PID 控制相比,模糊 PID 控制能使系统降低超调,提高响应速度,尤其对于时滞和不确定对象有更好的效果.然而模糊 PID 控制器的规则库一旦建立便无法更新,缺乏自适应性,模糊控制器部分又难以获得积分误差规则,势必影响到整个控制系统的精度.为了进一步提高模糊 PID 控制器的瞬态和稳态性能,下面提出一种新颖的模糊神经 PID 控制器,该控制器综合了模糊逻辑、神经网络、PID 控制 3 者的优点.

2 模糊神经 PID 控制器设计

FNPIDC 包含两个神经网络,前件为一个 FNN,主要用来代替原有的模糊控制器,起到自适应调整规则的作用;后件为一个 PIDNN,用来替代原有的 PID 参数部分,起到进一步提高系统瞬态和稳态性能,改善跟踪精度的作用.下面分别简单说明一下两种网络的独立结构.

FNN 为一个 4 层网络,分别为:输入层、模糊化层、模糊规则层、解模糊层.其输入量为 e 和 e_c ,输出为控制量 U .一般输入层为两个节点,模糊化层的节点数按 e 和 e_c 划分的模糊子集个数而定,激活函数即隶属函数用高斯函数表示为

$$\mu_{A_{ij}} = \exp\left[-(x_i - c_{ij})^2 / \sigma_{ij}^2\right] , \quad (2)$$

其中 x_i, c_{ij}, σ_{ij} 分别为输入变量、隶属函数中心、隶属函数宽度. 模糊规则层节点数为规则数, 激活函数为

$$O_k = \mu_{A_{1k}}(x_1) \cdot \mu_{A_{2k}}(x_2) , \quad (3)$$

O_k 为第 k 条规则的激活度. 解模糊层一般采用重心法解模糊.

PIDNN 由比例、积分、微分神经元构成, 它是一个 2-3-1 的 3 层前向型神经网络^[4]. 如图 2 所示.

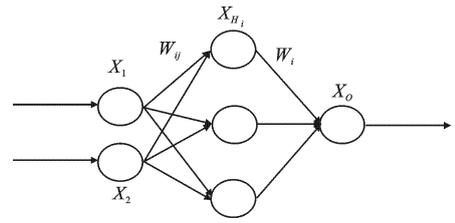


图 2 PID 神经网络拓扑图

X_1, X_2 为两个输入层节点, 当 PIDNN 单独用作控制器时, 输入层节点分别对应系统的输入和输出. X_{H1}, X_{H2}, X_{H3} 分别为比例、积分、微分神经元, X_O 为输出层节点. 其工作原理主要根据系统偏差 e , 通过控制算法, 如误差反传, 来不断调整权值, 对系统做出有效控制.

(1) 输入层神经元输入、输出 $I_1(k) = x(k) , \quad (4)$

$I_2(k) = y(k - 1) . \quad (5)$

(2) 隐含层神经元输入、输出

$$I_{H_i}(k) = \sum_{j=1}^2 \omega_{ij} I_j(k) , \quad i = 1, 2, 3 , \quad (6)$$

比例神经元: $O_{H_1}(k) = \begin{cases} -1 , & I_{H_1}(k) < -1 , \\ I_{H_1}(k) , & -1 \leq I_{H_1}(k) \leq 1 , \\ 1 , & I_{H_1}(k) > 1 , \end{cases} \quad (7)$

积分神经元: $O_{H_2}(k) = \begin{cases} -1 , & I_{H_2}(k) < -1 , \\ I_{H_2}(k) + I_{H_2}(k - 1) , & -1 \leq I_{H_2}(k) \leq 1 , \\ 1 , & I_{H_2}(k) > 1 , \end{cases} \quad (8)$

微分神经元: $O_{H_3}(k) = \begin{cases} -1 , & I_{H_3}(k) < -1 , \\ I_{H_3}(k) - I_{H_3}(k - 1) , & -1 \leq I_{H_3}(k) \leq 1 , \\ 1 , & I_{H_3}(k) > 1 . \end{cases} \quad (9)$

(3) 输出层神经元输入、输出: $O(k) = I(k) = \sum_{i=1}^3 \omega_i O_{H_i}(k) . \quad (10)$

PIDNN 中引入的 P, I, D 神经元使得网络成为一种动态前向网络, 增强了网络处理信息的能力, 更有利于对复杂非线性系统进行控制. 将 FNN 与 PIDNN 结合构成一种模糊神经 PID 控制器, 结构如图 3 所示. 这种改进是基于图 1 所示的调整控制量的模糊 PID 控制系统, 将 PIDNN 的输入变为一个节点, 其输入为 FNN 的输出, 它作为间接控制量加权后输入到 PIDNN 的隐层, 最后由隐层神经元的 P, I, D 作用产生 PID 控制效果, 输出系统控制量. 这样一来, 就把局限于规则得出的控制量 K_u 和静态参数的 K_p, K_i, K_d 都转化为动态调整形式, 融入了模糊推理、神经网络自适应调整模糊规则及 PID 简单控制 3 者的优点. 需要说明的是: 为使控制器参数在被控系统参数发生小范围变化时具有一定的鲁棒性, 控制器结构设计时需具备模糊子集划分和模糊规则的先验知识, 以此来确定第 2, 3 层神经元数, 利用 FNN 的自适应能力来对规则进行动态整定.

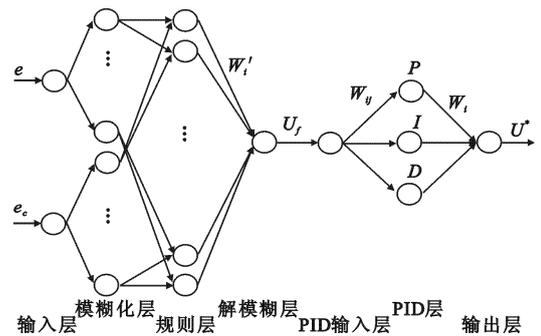


图 3 模糊神经 PID 拓扑结构图

FNPIDC 的控制律为

$$U = \omega_1 \cdot \omega_{11} \cdot U_f(k) + \omega_2 \cdot \omega_{21} \cdot [U_f(k) + U_f(k - 1)] + \omega_3 \cdot \omega_{31} \cdot [U_f(k) - U_f(k - 1)] , \quad (11)$$

其中 U 为控制器输出, $\omega_1, \omega_2, \omega_3$ 为 PID 层与输出层连接权, $\omega_{11}, \omega_{21}, \omega_{31}$ 为 PID 输入层与 PID 层连接权,

$U_f(k)$ 为模糊神经网络输出,它实际上是一个误差量的非线性映射.控制器需要学习的参数包括:FNN 中高斯函数的中心、宽度及规则层至解模糊层的权系;PIDNN 各层权系.FNPIDC 包含了模糊推理、神经网络自适应调整模糊规则及 PID 控制律 3 者的优点,从而为进一步加快系统响应速度、减小超调、提高稳态精度提供了有效支持.

3 混沌粒子群优化算法

对于控制器参数的离线优化,提出一种基于混沌策略与 PSO 结合的优化方案.

混沌是广泛存在于非线性系统中的一种现象,它具有随机性、遍历性的特点,在一定范围内能按照一定规律不重复地遍历所有的状态.基于这种特性,使得混沌搜索方法在智能控制领域的应用得到人们广泛的关注.混沌优化的基本思想是将变量的解空间转换到混沌空间.目前已有学者将其应用于神经网络权系的优化问题,并取得了良好的成效^[5].

这里采用 Logistic 映射产生混沌运动

$$x_{n+1} = \mu x_n (1 - x_n) \quad , \quad n = 0, 1, 2, \dots, M \quad , \quad 0 < x_0 < 1 \quad , \quad (12)$$

其中当 $\mu = 4$ 时,则完全处于混沌状态且混沌变量 x_n 在 $(0, 1)$ 范围内遍历.混沌运动对初始值的选取极其敏感,不同的初始值将得到不同的遍历轨迹.

PSO 算法是基于群智能的全局性搜索算法,它是 Kennedy 和 Eberhart 于 1995 年提出来的,其思想是受到鸟群、鱼群觅食的启发.算法采用速度-位置搜索模型,并通过惯性权重协调全局搜索和局部搜索,使算法能以较大概率收敛到最优解^[6].

标准 PSO 算法公式为

$$V_i = \omega \cdot V_i + c_1 \cdot \text{rand}() \cdot (p_{\text{best}} - X_i) + c_2 \cdot \text{rand}() \cdot (g_{\text{best}} - X_i) \quad , \quad (13)$$

$$X_i = X_i + V_i \quad , \quad (14)$$

式中 $i = 1, 2, \dots, n$, n 是群体中微粒的总数; V_i 是微粒的速度; ω 是惯性权重,通常取 $0.4 \sim 1.2$; p_{best} 和 g_{best} 分别表示粒子自身搜索到的最优值和当前种群中的最优值,称为个体极值和全局极值; $\text{rand}()$ 是介于 $(0, 1)$ 之间的随机数; X_i 是粒子的当前位置; c_1 和 c_2 是加速常数,也称学习因子,通常取 $c_1 = c_2 = 2$. 粒子在 N 维空间中的位置和速度可以表示为 $X_i = (x_1, x_2, \dots, x_N)$ 和 $V_i = (v_1, v_2, \dots, v_N)$. 每个粒子都有一个由自定义目标函数所确定的适应度值(fitness value),且每个粒子都存储了当前自身搜索到的最优值和当前种群中的最优值.并将这些信息作为经验进行动态调整.其中惯性权重一般按 shi 建议的线性递减权值算子为^[7]

$$\omega(i_{\text{ter}}) = \omega_{\text{max}} - (\omega_{\text{max}} - \omega_{\text{min}}) / i_{\text{max}} \cdot i_{\text{ter}} \quad , \quad (15)$$

其中 i_{max} 为最大进化代数; i_{ter} 为代数; ω_{max} 为最大惯性权值; ω_{min} 为最小惯性权值. ω 的引入使 PSO 算法性能有了很大提高,它可以调整全局和局部搜索能力,使 PSO 算法能成功应用于很多实际问题.

PSO 算法在处理函数优化问题上有着比遗传算法更优越的性能,但当变量维数增大,目标函数又较为复杂时,算法后期容易陷入局部极小.针对这个问题,大量学者对 PSO 算法进行了研究与改进,如带邻域操作的 PSO、带限制因子的 PSO、带变异算子的 PSO 等,使得 PSO 算法能尽可能地收敛到全局最优.但由于 PSO 算法缺乏数学理论的支持,故其寻优潜力还有待发掘.

文中算法具体流程如下:

Step 1 待优化参数编码.网络结构确定以后,将待优化参数——隐层中心 c 、标准化参数 σ 、输出层权值和 PIDNN 的权系 ω' 编码为向量 $c_1 \dots c_n, \sigma_1 \dots \sigma_n, \omega'_1 \dots \omega'_{(n/2)^2}, \omega_{11}, \omega_{21}, \omega_{31}, \omega_1, \omega_2, \omega_3$, 记为 P_{ara} . 编码方式为实数编码.其中 n 为模糊化层神经元个数,以输入变量模糊子集数相等为前提,则 $(n/2)^2$ 为规则数.

Step 2 定义适应度函数.这里采用 ITAE 指标,即误差绝对值乘时间积分准则,采用 ITAE 准则的系统一般具有快速、平稳、超调小等优点,ITAE 准则表达式为

$$J = \int_0^t |e(t)| dt \quad . \quad (16)$$

Step 3 初始化 PSO 算法参数并开始寻优. PSO 初始参数 $c_1 = c_2 = 2$, $\omega_{\text{max}} = 1.2$, $\omega_{\text{min}} = 0.4$, 将式

(16)作为 PSO 算法训练学习的目标函数,在全局范围内进行搜索寻优.

Step 4 第一步混沌辅助搜索. 设定一个混沌微调的触发条件:迭代次数 λ 和一个微小值 ξ . 当 PSO 精度在 λ 次迭代中变化小于 ξ 时,保存当前参数值为 P_{now} ,将其赋给混沌搜索的初始值 P_0 , $P_0 = P_{\text{now}}$. 在 $(0, 0.5)$ 区间随机初始化混沌变量 X_0 (由于 Logistic 映射为在 $(0, 1)$ 上对称),维数与 P_{now} 一致. 并定义 $i = 0$,利用下面两个式子产生新的参数值

$$P_0 = P_0 + \alpha(2X_i - 1) \quad , \quad (17)$$

$$X_{i+1} = 4X_i(1 - X_i) \quad , \quad i = i + 1 \quad , \quad (18)$$

其中 α 为搜索半径,调整它可以在较大范围内遍历搜索,但更耗时. 这一步骤的主要目的是在一定程度上避免粒子的速度和位置更新不合适,故 α 的作用只是微调,一般取较小的值. 记搜索到的更优值为 P_0 , 否则调整 α .

Step 5 第二步混沌策略(该操作作为 PSO 算法结束部分的扩展). 由于在较大范围内混沌优化非常费时,故第一步的主要目的是在全局范围内进行局部微调,起辅助作用. 第二步才是真正意义上的细搜索. 即在第一步混沌辅助下 PSO 搜索到一组参数值,记为 P_f . 再次利用混沌优化方法 $P_f = P_f + \beta(2X_i - 1)$, $X_{i+1} = 4X_i(1 - X_i)$, $i = i + 1$ 在小范围内遍历, β 为搜索半径. 将找到的更优值记为最终控制器的优化参数 P_{final} . 若精度不满足要求,返回调整 β 值,直到满足精度要求为止.

关于参数 $\lambda, \xi, \alpha, \beta$ 的探讨:参数 λ 可在一定程度上依据粒子群优化总的迭代次数来设,不宜间隔太密; ξ 一般取很小的值,在一定迭代次数后适应值几乎没有变化时,就利用混沌运动重新激活粒子,寻找更优的适应值; α 和 β 均为起辅助作用的调整因子,一般满足两个条件: α, β 的引入均应使寻优变量仍然在其解空间内,否则将耗费计算资源;它们都应该是较小的值,混沌优化在大范围内非常耗时,它们主要是起辅助的作用.

获得离线次优控制器参数后,在线仿真时利用 BP 算法进行调整. 由于 BP 算法在初始参数与最优值越接近时收敛到最优解的可能性越高,否则可能收敛到次优甚至发散,故离线获得的结果对于实时在线调整是很重要的. 在线 BP 算法调整按下面的式子进行调整

$$\begin{cases} c(n+1) = c(n) - x_{\text{ite}} \cdot (\partial E / \partial c) + \eta \cdot \Delta c(n) \quad , \\ \sigma(n+1) = \sigma(n) - x_{\text{ite}} \cdot (\partial E / \partial \sigma) + \eta \cdot \Delta \sigma(n) \quad , \\ \omega(n+1) = \omega(n) - x_{\text{ite}} \cdot (\partial E / \partial \omega) + \eta \cdot \Delta \omega(n) \quad . \end{cases} \quad (19)$$

4 仿真实验

选取一个典型非线性对象和一个一阶惯性大滞后对象分别进行仿真实验.

$$(1) \text{ 非线性对象模型 } y(n) = (2y(n-1) + u(n-1)) / (3 + y(n-1)^2) \quad . \quad (20)$$

期望信号:幅值为 1 的阶跃信号,采样时间为 0.001 s,进行信号跟踪控制仿真.

(2) 一阶惯性大滞后对象模型,延时 2 min:

$$G(s) = (a / (bs + 1)) \exp(-2s) \quad . \quad (21)$$

期望信号:幅值为 10 的阶跃信号,触发时间为 0. 分别用传统 PID、模糊、模糊 PID 及文中方法进行控制比较. 其中模糊、模糊 PID 控制的输入 E, E_c 及输出量 U 的模糊子集均划分为 5 个,隶属度函数为三角形,共 25 条规则. a, b 初始值取 $a = 1, b = 50$,同时考虑 a, b 小范围变化时的控制效果.

FNPIDC 的结构选取为:2-10-25-1-3-1,待训练参数为 51 个. 离线时混沌粒子群算法初始参数值为 $c_1 = c_2 = 2, \omega_{\text{max}} = 1.2, \omega_{\text{min}} = 0.4$,迭代次数 5 000 次. 混沌辅助参数 λ, ξ 的初始值为 $\lambda = 300, \xi = 0.0001$,当 PSO 在 300 代中适应度变化小于 0.0001 时,对于粒子进行混沌操作; PSO 寻优结束后再次由混沌运动进行遍历,求出更接近最优值的参数. 在线控制仿真采用离线时获得的控制器和误差反传调整,其仿真结果如图 4 和图 5 所示.

以惯性时滞系统为例,当对象参数发生小范围变化时,如取 $a = 1.2, b = 53$,控制器离线参数不变情况下进行在线控制发现其超调量仍然非常小且响应速度变化不大,说明系统能适应参数在小范围内变化,具有一定的鲁棒性.

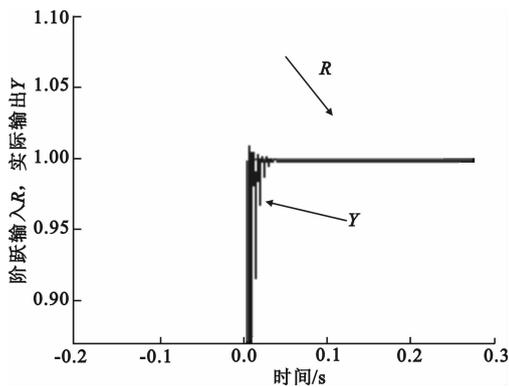


图 4 非线性系统阶跃信号跟踪

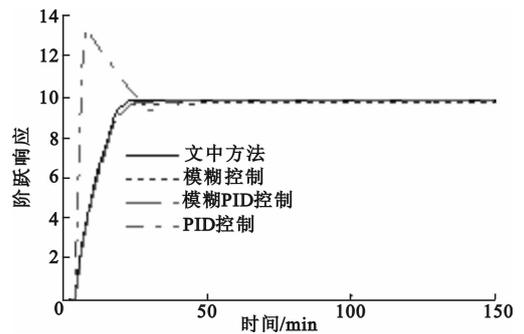


图 5 惯性时滞系统阶跃信号跟踪

从结果可以看出:对非线性、惯性大滞后对象,笔者设计的控制器及其优化算法能对其进行有效的控制,控制系统响应速度快、超调量小、稳态精度高,能快速进行信号跟踪并具有一定的鲁棒性,满足工况下实时控制的要求.在引入合适的系统辨识器(如动态 Elman 网或支持向量机)后,可以很方便地设计一个能处理具有不确定参数的时变、强非线性问题的控制系统,具有较好的应用价值.文中控制器的误差反传算法按前向型神经网络的推导方法得到,这里不做具体推导.

5 结束语

在继承 PID 简单控制的基础上合理地将其与各种智能方法相结合,设计了一种新型的模糊神经 PID 控制器.不仅融合了各种智能控制方法的优点,而且从本质上突破了 PID 控制的线性局限性并拓展了其使用范围.同时,在所设计的神经网络结构的控制器的基础上,利用混沌优化与 PSO 两步结合的混沌粒子群离线优化和误差反传在线调整相结合的方法来获得控制器的参数,有效地解决了控制器参数较多时不易于在线调整及难以获得近似最优解的问题.通过控制器与优化算法的有效结合,提高了控制系统的瞬态和稳态性能,对非线性、惯性滞后对象取得了良好的控制品质,且保持了一定的鲁棒性.仿真算例验证了控制器及其优化算法的有效性和合理性.仍需改进的是,在设计控制器时对网络结构进行优化,简化控制器结构,从而进一步提高控制系统的性能.

参考文献:

- [1] Chen G. Conventional and Fuzzy PID Controllers: an Overview[J]. Int J of Intelligent Control & Systems, 1996, 1(2): 235-246.
- [2] Mudi R K, Pal N R. A Self-tuning Fuzzy PI Controller[J]. Fuzzy Sets and Systems, 2000, 115(3): 327-338.
- [3] 谭冠政, 李安平, 王越超. 最优 Fuzzy-GA PID 控制器及其应用[J]. 中南工业大学学报, 2002, 33(4): 419-423.
- [4] Shu Huailin, Pi Youguo. PID Neural Networks for Time-delay Systems[J]. Computers and Chemical Engineering, 2000, 24(8): 859-862.
- [5] 李祥飞, 邹恩. 前馈神经网络的混沌学习方法研究[J]. 小型微型计算机系统, 2004, 25(2): 233-236.
- [6] Kennedy J, Eberhart R C. Particle Swarm Optimization[C]//Proceedings of IEEE International Conference on Neural Networks:04. Perth: IEEE, 1995: 1942-1948.
- [7] Shi Y, Eberhart R. A Modified Particle Swarm Optimizer[C]//Proc IEEE Int Conf, Evolution Computer: 05. Anchorage: IEEE, 1998: 69-73.

(编辑: 齐淑娟)