

基于合作粒子群算法的PID神经网络非线性控制系统

朴海国, 王志新, 张华强

(上海交通大学 电气工程系, 上海 200240)

摘要: PID神经网络(PIDNN)模型为一种新型的神经网络模型, 兼有PID与神经网络的共同优点, 应用于复杂的控制系统, 取得优良控制性能, 但其后向传播算法(BP)限制了该模型的应用范围. 为实现对非线性多变量系统的有效控制, 扩展神经网络的应有范围, 本文采用PIDNN神经网络设计了多变量PIDNN神经网络(MPIDNN)控制器, 并用本文作者提出的合作粒子群算法(CPSO)取代了传统BP后向传播算法, 通过比较MPIDNN_CPSO、MPIDNN_CRPSO、MPIDNN_PSO和MPIDNN_BP4种控制器的控制性能, 仿真结果表明, 基于CPSO算法的MPIDNN控制器实现了对非线性多变量不对称系统的有效控制. 与传统的BP算法相比, CPSO算法提高了控制系统的稳定性、精确性与鲁棒性.

关键词: PID神经网络; 粒子群算法; 非线性不对称控制; 稳定性; 鲁棒性; 合作粒子群最优算法

中图分类号: TP273 **文献标识码:** A

Nonlinear control system of PID neural network based on cooperated particle swarm optimization(PSO)

PIAO Hai-guo, WANG Zhi-xin, ZHANG Hua-qiang

(Department of Electrical Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: The PID neural network(PIDNN) model is a novel neural network model with the advantages of PID and artificial neuron network. This model has been used for complex control systems to achieve desirable control performances. However, the conventional backward-propagation(BP)algorithm restrains the model's wide applications to control field. To control the nonlinear MIMO system efficiently and to extend the application range of PIDNN, we develop the MIMO PID neural network(MPIDNN) controller based on PIDNN, and propose the cooperated PSO(CPSO) algorithm to take the place of BP algorithm. Simulation results of the MPIDNN controllers based on BP, PSO and CRPSO algorithms indicate that CPSO-based MPIDNN controller is more effective than the other three in controlling the MIMO systems. The CPSO algorithm makes MPIDNN controller better in performances than BP algorithm in accuracy, stability and robustness.

Key words: PID neural network; PSO algorithm; nonlinear dissymmetrical control; stability; robust; CPSO

1 引言(Introduction)

传统PID控制算法广泛应用于工业控制系统, 但不适合应用于控制复杂的非线性系统, PID的控制参数难以选择^[1]. 非线性MIMO系统常为复杂强耦合系统, 对此类复杂系统, 为实现有效控制, 获得优良控制性能, 神经网络被引入控制领域. 神经网络具有非线性特性, 可实现自适应控制, 适合解决非线性问题. 但直到今天, 人工神经网络未能广泛应用于控制领域, 对此, 文献[2]给出原因: 仿真训练时间长, 不能满足快速性要求; 连接权重随机初始化, 影响控制结果, 有时导致系统不稳定; 神经元函数为静态函数, 不适合控制动态系统. 此外, 为获得优良控制性能,

神经网络层数也难于抉择. 基于PID优良动态控制性能与人工神经网络优秀非线性特性, 文献[2]提出一种新型神经网络模型——PID神经网络(PIDNN)模型. 该神经网络为新型动态神经网络, 非PID与人工神经网络的简单混合. 在控制领域, 与传统的神经网络相比, 具有训练时间短, 动态性能好, 层数结构清晰等优点. 此神经网络采用传统BP后向传播算法——梯度下降法. BP算法全局搜索能力弱, 易陷入局部最优; 训练结果与初始权重值并学习步长选取密切相关, 为获得满意训练效果, 须仔细、多次试取, 但试取不能保证每次均能获得期望控制效果, 训练有时不收敛, 需用强迫方法提前终止以保证控制系统稳定性

与鲁棒性. 因此, BP后向传播算法限制了PIDNN神经网络的应用, 使其许多优点不能得到有效发挥.

PIDNN控制器设计作为一种最优问题, 其训练目的恰是为了在权值的解空间中发现最优值, 以期达到跟踪误差最小目的. 可供选择并常用于最优问题的、比较成熟的算法有免疫算法(AIA)、蚁群算法(ACO)、遗传算法(GA)和粒子群算法(PSO). 这些仿生计算模型在不同的领域都有成功的相应应用. 本文中选取PSO算法作为MPIDNN控制器训练算法.

粒子群算法是一种新型的、基于总群进化的全局搜索算法, 由Eberhart and Kennedy在1995年首次提出^[3], 在解决实值最优问题方面, 取得理想效果. 该算法能有效定位最优值所在盆地范围, 但不能快速有效搜索该盆地范围^[4,5]. 为此, Eberhart与Shi在文献^[6]中, 介绍了一种时间衰减惯性因数来平衡PSO算法的全局与局部搜索能力. 为克服PSO算法存在的缺陷, 文献^[7]提出了合作随机粒子群算法(CRPSO), 为PSO的改进算法, 该算法用于预测模型, 与BP、PSO和GA(遗传算法)比较, 取得较好预测结果. 为有效控制非线性不对称MIMO系统并设计MPIDNN控制器, 采用了本文作者提出的合作粒子群算法(CPSO), 此算法为CRPSO的改进算法, 依据此算法设计了MPIDNN控制器, 并形成了基于CPSO的PIDNN神经网络系统设计方法. 与BP、PSO和CRPSO算法比较, 在控制精度、稳定性、鲁棒性和解耦方面, CPSO使得MPIDNN控制器具有最好的控制性能表现, 有利于PIDNN在控制领域的广泛应用.

2 PIDNN神经网络结构与算法(The structure and algorithm of PIDNN neural network)

PIDNN基本结构由3部分构成: 输入层、隐含层和输出层. 输入层包含两个比例神经元, 一个用于输入系统给定, 一个用于输入系统输出反馈. 输出层则由一个比例神经元构成, 用于被控系统控制量的输出. 隐含层为PIDNN主要构成部分, 含有3种神经元: 比例神经元(P)、积分神经元(I)和微分神经元(D). 网络结构与单变量控制系统如图1所示:

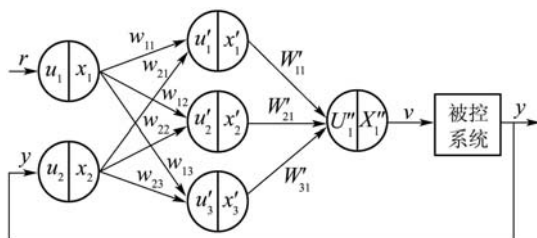


图1 PIDNN控制系统基本结构图

Fig. 1 The basic structure of PIDNN control system

2.1 前向传播算法(Forward algorithm)

由图1可知, PIDNN前向传播算法也是由3部分构成. 输入层中, 两个神经元的输入输出函数定义如下式:

$$x_i(k) = \begin{cases} q', & u_i(k) \geq q, \\ u_i(k), & -q < u_i(k) < q, \\ -q', & u_i(k) \leq -q. \end{cases} \quad (1)$$

式中: $i = 1, 2$, 为输入层比例神经元编号; k 为采样时刻; q 为输入最大限制值; q' 为超过最大限制值 q 时的实际输出值; $u_i(k)$ 为输入层第 i 个神经元在 k 采样时刻的输入值; $x_i(k)$ 为输入层第 i 个神经元在 k 采样时刻的输出值.

在隐含层中, 神经元的输入定义如下:

$$u'_j = \sum_{i=1}^2 w_{ij} \cdot x_i(k). \quad (2)$$

式中: $j = 1, 2, 3$, 为隐含层神经元编号; w_{ij} 为输入层与隐含层之间的连接权值. 隐含层各神经元的输入输出函数分别定义如下式:

$$x'_1(k) = \begin{cases} q', & u'_1(k) \geq q, \\ u'_1(k), & -q < u'_1(k) < q, \\ -q', & u'_1(k) \leq -q; \end{cases} \quad (3)$$

$$x'_2(k) = \begin{cases} q', & u'_2(k) \geq q, \\ x'_2(k-1) + u'_2(k), & -q < u'_2(k) < q, \\ -q', & u'_2(k) \leq -q; \end{cases} \quad (4)$$

$$x'_3(k) = \begin{cases} q', & u'_3(k) \geq q, \\ u'_3(k) + u'_3(k-1), & -q < u'_3(k) < q, \\ -q', & u'_3(k) \leq -q. \end{cases} \quad (5)$$

式(3)~(5)分别为比例神经元、积分神经元和微分神经元.

在输出层中, 唯一的比例神经元, 其输入定义为:

$$u''_h = \sum_{j=1}^3 w_{jh} \cdot x'_j(k). \quad (6)$$

式中: $h = 1$, 为输出神经元的编号; w_{jh} 为隐含层与输出层的连接权值. 输出层比例神经元的输入输出函数为下式:

$$x''_h(k) = \begin{cases} q', & u''_h(k) \geq q, \\ u''_h(k), & -q < u''_h(k) < q, \\ -q', & u''_h(k) \leq -q. \end{cases} \quad (7)$$

2.2 价值函数与后向传播(BP)算法(Cost function and BP algorithm)

价值函数为神经网络训练的依据, 作为评估方法, 在神经网络控制器设计中不可或缺. 文献^[2]采用了函数名为MSE的价值函数. 该函数根据误差平

方的平均值评估网络的性能表现. 基本PIDNN的价
值函数定义如下:

$$J = E_h = \frac{1}{m} \sum_{k=1}^m [r(k) - y(k)]^2. \quad (8)$$

$$\Delta w_{ij} = \eta \frac{dJ}{dw_{ij}} = \begin{cases} -\frac{2}{m} \sum_{k=1}^m [r(k) - y(k)] \cdot \frac{dy}{dx''} \cdot w_{jh}(k) \cdot 1 \cdot x_i, & P_{\text{neuron}}, \\ -\frac{2}{m} \sum_{k=1}^m [r(k) - y(k)] \cdot \frac{dy}{dx''} \cdot w_{jh}(k) \cdot u'_j \cdot x_i, & I_{\text{neuron}}, \\ -\frac{2}{m} \sum_{k=1}^m [r(k) - y(k)] \cdot \frac{dy}{dx''} \cdot w_{jh}(k) \cdot \frac{dx'_j}{du'_j} \cdot x_i, & D_{\text{neuron}}, \end{cases}$$

$$\Delta w_{jh} = \eta \frac{dJ}{dw_{jh}} = -\frac{2}{m} \sum_{k=1}^m [r(k) - y(k)] \cdot \frac{dy}{dx''} \cdot x'_j(k),$$

$$w_{ij}(n_0 + 1) = w_{ij}(n_0) - \Delta w_{ij}, \quad w_{jh}(n_0 + 1) = w_{jh}(n_0) - \Delta w_{jh}. \quad (9)$$

式(9)中: η 为学习步长; $\frac{dy}{dx''}$ 与 $\frac{dx'_j}{du'_j}$ 分别由

$$\text{sgn}\left(\frac{y(k+1)-y(k)}{x''(k)-x''(k-1)}\right) \text{和} \text{sgn}\left(\frac{x'_j(k)-x'_j(k-1)}{u'_j(k)-u'_j(k-1)}\right)$$

替代. 因为 $\frac{dy}{dx''}$ 与 $\frac{dx'_j}{du'_j}$ 为式(9)中的公共因子, 其符号的正负决定权值的变化方向, 其数值只影响权值的变化速度, 而权值变化速度可以通过学习步长加以调节, 因此, 这种替代是许可的.

3 CPSO合作粒子群算法与MPIDNN控制系统(CPSO algorithm and MPIDNN control system)

作为传统PSO粒子群的改进算法, CRPSO随机合作粒子群算法的特色在于该算法包含多个子群, 通过随机方式从所有子群中选择每个子群的最优位置 g_{best} , 并用于粒子的速度与位置更新. 该算法改善了传统PSO算法上的不足. 但由于 g_{best} 的随机选择机制, 致使各个子群的粒子更新过分依赖其他子群, 从而削弱了子群自身获得的 g_{best} 在子群内部粒子更新上的作用, 进而未能充分发挥子群自身的搜索能力. 为克服CRPSO算法的缺陷并利用其优点, 提出了修正算法, 即CPSO算法.

3.1 CPSO合作粒子群算法(CPSO algorithm)

3.1.1 传统PSO粒子群算法(Conventional PSO algorithm)

PSO为一种新颖的进化算法, 来源于对鸟群或鱼群寻索食物的模仿^[8,9]. PSO中的每个粒子代表着一只飞鸟或是一条鱼, 拥有一个位置和一个速度, 分别代表最优问题在解空间的解与搜寻方向.

式中: m 为采样点总的数目; $r(k)$ 、 $y(k)$ 分别为控制系统的输入与输出.

第 n_0 步后, BP算法的权值修改公式为:

每一个粒子根据自身在搜寻过程中获得的最好经验 p_{best} 与全群粒子在搜寻过程中的获得的最好经验 g_{best} 来调整自身的速度与位置. 粒子的速度与位置更新公式为:

$$\begin{cases} v(k+1) = \omega \cdot v(k) + c_1 \cdot r_1 \cdot [p_{\text{best}} - x(k)] + \\ \quad c_2 \cdot r_2 \cdot [g_{\text{best}} - x(k)], \\ x(k+1) = x(k) + v(k+1). \end{cases} \quad (10)$$

式中: v 与 x 代表粒子的速度与位置; k 为采样时刻; ω 为平衡PSO算法全局搜索能力与局部搜索能力的时间衰减因子; c_1 与 c_2 为正值的加速常数; r_1 与 r_2 为介于0与1之间的随机数值; p_{best} 代表粒子自身发现的最优位置; g_{best} 为全群粒子发现的最优位置.

3.1.2 CRPSO随机合作粒子群算法(CRPSO algorithm)

在合作搜寻模式下, 文献[7]提出了PSO的改进算法——CRPSO算法, 该算法的目标是使用多个子群来搜寻解空间的不同部分, 通过采用随机合作机制来平衡全局搜索与局部搜索能力.

在CRPSO算法中, 每一子群中的粒子通过随机方式选择其他子群发现的最优位置 g_{best} 用于更新粒子速度与最优位置. CRPSO的速度、位置更新公式如公式(11)所示:

$$\begin{cases} v_i(k+1) = \omega \cdot v_i(k) + c_1 \cdot r_1 \cdot [p_{\text{best}} - x_i(k)] + \\ \quad c_2 \cdot r_2 \cdot [g_{\text{best}}(r) - x_i(k)], \\ x(k+1) = x(k) + v(k+1). \end{cases} \quad (11)$$

式中: $i = 1, \dots, n$, 为算法中子群的数目; r 为介于1到 n 之间的随机整数, 用于选择不同子群间的

最优位置值 g_{best} , 以期达到信息共享的目的, 并提高算法的全局与局部搜索能力.

3.1.3 CPSO合作粒子群算法(CPSO algorithm)

在预测方面, CRPSO算法改善了PSO算法存在的不足, 提高了算法搜索能力^[7], 但在控制应用上, 这种改善并不十分明显, 相同粒子数目时, 搜索能力有时低于PSO算法. 为提高算法在控制上的全局与局部搜索能力, CPSO算法通过随机方式使用其他子群最优位置 g_{best} , 并保留每个子群自身最优位置 g_{best} , 同时将两种最优位置值用于子群中粒子的速度与最优位置更新. CPSO算法的更新公式定义为:

$$\begin{cases} v_i(k+1) = \omega \cdot v_i(k) + 0.5 \cdot c_1 \cdot r_1 \cdot [p_{best} - x_i(k)] + \\ \quad 0.5 \cdot c_2 \cdot r_2 \cdot [g_{best} - x_i(k)] + \\ \quad 0.5 \cdot c_2 \cdot r_2 \cdot [g_{best}(r) - x_i(k)], \\ x(k+1) = x(k) + v(k+1). \end{cases} \quad (12)$$

式中0.5用于平衡粒子最优位置 p_{best} , 子群自身最优位置 g_{best} 与其他子群最优位置 $g_{best}(r)$ 在更新公式中的作用.

3.2 多变量MPIDNN(MPIDNN)控制系统(MPIDNN control system)

3.2.1 MPIDNN控制系统结构(The structure of MPIDNN control system)

为实现非线性不对称MIMO系统的解耦控制,

$$\Delta w_{ij} = \eta \frac{\partial J}{\partial w_{sij}} = \begin{cases} -\frac{2}{m} \sum_{s=1}^2 \sum_{h=1}^3 \sum_{k=1}^m [r_s(k) - y_s(k)] \cdot \frac{\partial y_s}{\partial x_h''} \cdot w_{sjh}(k) \cdot 1 \cdot x_{si}, & P_{neuron}, \\ -\frac{2}{m} \sum_{s=1}^2 \sum_{h=1}^3 \sum_{k=1}^m [r_s(k) - y_s(k)] \cdot \frac{\partial y_s}{\partial x_h''} \cdot w_{sjh}(k) \cdot u'_{sj} \cdot x_{si}, & I_{neuron}, \\ -\frac{2}{m} \sum_{s=1}^2 \sum_{h=1}^3 \sum_{k=1}^m [r_s(k) - y_s(k)] \cdot \frac{\partial y_s}{\partial x_h''} \cdot w_{sjh}(k) \cdot \frac{\partial x'_{sj}}{\partial u'_{sj}} \cdot x_{si}, & D_{neuron}, \end{cases}$$

$$\Delta w_{jh} = \eta \frac{\partial J}{\partial w_{sjh}} = -\frac{2}{m} \sum_{s=1}^2 \sum_{k=1}^m [r_s(k) - y_s(k)] \cdot \frac{\partial y_s}{\partial x_h''} \cdot x'_{sj}(k). \quad (14)$$

式(14)中 $\frac{\partial y_s}{\partial x_h''}$ 与 $\frac{\partial x'_{sj}}{\partial u'_{sj}}$ 分别由

$$\text{sgn}\left(\frac{y_s(k+1) - y_s(k)}{x_h''(k) - x_h''(k-1)}\right) \text{和} \text{sgn}\left(\frac{x'_{sj}(k) - x'_{sj}(k-1)}{u'_{sj}(k) - u'_{sj}(k-1)}\right)$$

表示.

3.2.2 CPSO训练实现(The training of CPSO)

MPIDNN控制系统的控制训练过程可简单的归结为如下步骤:

Step 1 初始化MPIDNN与CPSO的全部参数; 定义速度、位置的维数与范围.

建立了MPIDNN控制系统. 结构图如图2所示, 由图2可知, 被控系统为3输入2输出多变量系统, 各层间的连接权值分别定义为 w_{sij} 与 w_{sjh} , 其中 s 为基本PIDNN网络结构的数目; i, j, h , 定义如前.

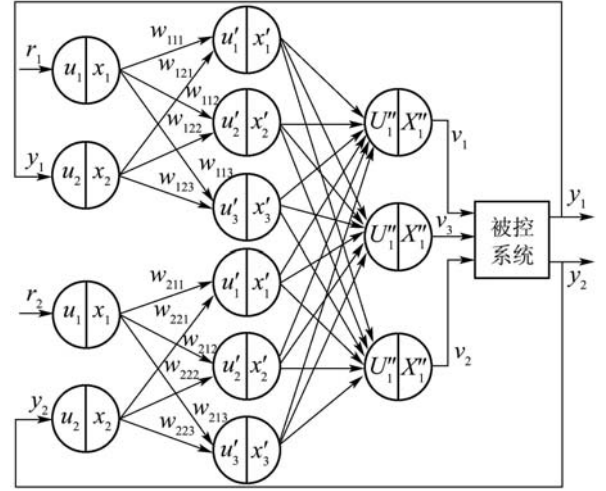


图2 MPIDNN控制系统结构图

Fig. 2 The structure of MPIDNN control system

与单变量PIDNN控制系统相比, MPIDNN控制系统中的输出层由基本PIDNN网络结构的输出共同决定. 同时, 价值函数与权值修改公式相应的改变为:

$$J = \sum_{s=1}^2 E_s = \frac{1}{m} \sum_{s=1}^2 \sum_{k=1}^m [r_s(k) - y_s(k)]^2, \quad (13)$$

Step 2 初始化所有子群粒子的速度与位置.

Step 3 计算第一步仿真步中每一个粒子的价值函数值, 即所有粒子的适应值; 初始化每一子群的最优位置 p_{best} 与 g_{best} , 并存储所有子群中最小的 g_{best} 于变量 g_{best} 中.

Step 4 随机产生1至 n 中的随机整数 r .

Step 5 根据公式(11)更新子群中第 i 个粒子的速度与位置.

Step 6 如果 $f(x_i) < f(p_{best})$, 则更新子群中第 i 个粒子的 p_{best} .

Step 7 如果 $f(p_{\text{best}}) < f(g_{\text{best}})$, 则更新子群的 g_{best} .

Step 8 保存所有子群中最小的 g_{best} 于 g_{best} 中.

Step 9 重复Step4~8, 直到设定的最大循环仿真步数.

Step 10 应用 g_{best} 中存储的权值最优解控制 MPIDNN 控制系统.

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} \frac{0.7547z^{-1}}{1-0.4125z^{-1}} & \frac{0.6546z^{-1}}{1-0.5724z^{-1}} & \frac{0.2458z^{-1}}{1-0.7412z^{-1}} \\ \frac{0.1526z^{-1}}{1-0.8125z^{-1}} & \frac{0.8545z^{-1}}{1-0.7125z^{-1}} & \frac{0.4489z^{-1}}{1-0.7581z^{-1}} \end{bmatrix} \cdot \begin{bmatrix} V_1^2(z) \\ V_2(z) \\ V_3^2(z) \end{bmatrix}. \quad (15)$$

由式(14)不难看出系统的非线性特性及其输入输出之间的强耦合特性.

MPIDNN_CPSO与MPIDNN_CRPSO控制系统的参数给定如下:

$s = 2, h = 3, q = 1, q' = 1$, 采样数 = 200, 最大循环步数 = 50, 子群数 = 3, 每一子群中粒子数 = 30; 权值解空间由两部分构成: w_{sij} 输入层与中间层的权值范围 = $[-1.5 \ 1.5]$, w_{sjh} 中间层与输出层的权值范围 = $[-2 \ 2]$, 权值解空间维数 = 30; 设定 $X_{\max sij} = -X_{\min sij} = 1.5$, $X_{\max sjh} = -X_{\min sjh} = 2$, 则解空间两部分速度的最大值与最小值可分别表示为: $V_{\max sij} = -V_{\min sij} = 0.1 \cdot (X_{\max sij} - X_{\min sij}) = 0.3$, $V_{\max sjh} = -V_{\min sjh} = 0.1 \cdot (X_{\max sjh} - X_{\min sjh}) = 0.4$; 每一子群的位置与速度初始化公式可表示为:

$$\begin{cases} \text{pos} = X_{\min} + (X_{\max} - X_{\min}) \cdot \text{rand}, \\ \text{vel} = V_{\min} + 2 \cdot V_{\max} \cdot \text{rand}. \end{cases} \quad (16)$$

式中rand为0至1之间的一随机数值.

MPIDNN_PSO控制系统的参数给定, 除粒子数 = 90的设定不同外, 其他均与MPIDNN_CPSO相同.

MPIDNN_BP控制系统的参数给定为: $s = 2, h = 3, q = 1, q' = 1$, 采样数 = 200, 最大循环步数 = 4500, 学习步长 $\eta = 0.01$; 各层之间的权值初始化值:

$$\begin{aligned} w_{sij}(1:s, 1, 1) &= 2, w_{sij}(1:s, 2, 1) = -2, \\ w_{sij}(1:s, 1, 2) &= 1, w_{sij}(1:s, 2, 2) = -1, \\ w_{sij}(1:s, 1, 3) &= 2, w_{sij}(1:s, 2, 3) = -3, \\ w_{sjh}(1:s, 1:3, 1:h) &= 0, w_{sjh}(1, 1:3, 1:3) = 0.1. \end{aligned}$$

此处初始化权值经多次仿真实验确定, 在接下的

其中 $f(x_i), f(p_{\text{best}}), f(g_{\text{best}})$ 分别代表第 i 个粒子的适应值、粒子最优位置 p_{best} 与子群最优位置 g_{best} .

4 仿真与分析(Simulation and analysis)

4.1 被控系统与控制系统参数给定(Object system and the parameters of control system)

被控制系统的离散方程如下式所示:

实验中不再发生变化.

4.2 仿真实验与结果分析(The experiment of simulation and the analysis of results)

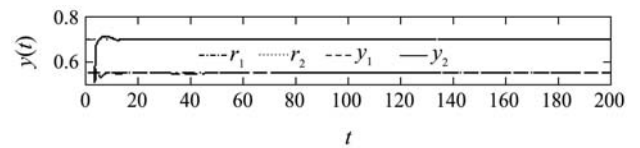
在3种典型输入下, 分别针对4种算法下的 MPIDNN 控制器的控制效果进行了MATLAB仿真实验.

4.2.1 恒定值系统输入(System input with constant values)

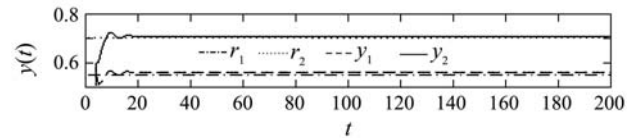
系统的输入给定为:

$$\begin{cases} r_1 = 0.55, \\ r_2 = 0.7. \end{cases} \quad (17)$$

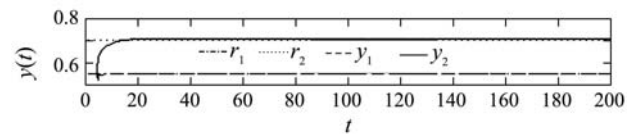
基于此给定输入的仿真结果如图3所示.



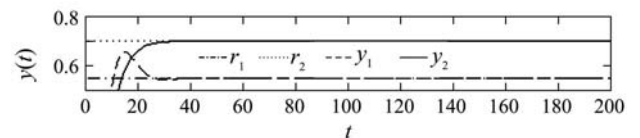
仿真步数: 200, 误差值: 1.2405e-005 CPSO



仿真步数: 200, 误差值: 0.000259 CRPSO



仿真步数: 200, 误差值: 0.000417 PSO



仿真步数: 200, 误差值: 0.013077 BP

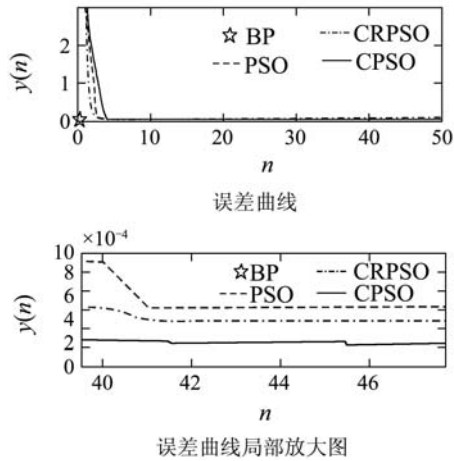


图3 恒定输入值下的MPIDNN控制系统仿真曲线图
Fig. 3 The simulation curves of MPIDNN control system with constant system inputs

由图3可知, 对被控系统, 4种算法下的MPIDNN控制系统均成功进行了解耦控制, 获得了良好控制效果. 此处粒子群算法的仿真步数为50步, 与文献[7]中的神经网络千步训练相比, 训练步数大大缩短. 比较跟踪曲线、误差曲线和误差值可见, 粒子群PSO、CRPSO与CPSO算法误差曲线的衰减, 表明搜索一直向着最优方向进行, 粒子群及其改进算法均有较强的全局与局部搜索能力; 相同粒子数, CRPSO算法局部搜索能力弱于PSO算法, CPSO算法局部搜索能力最强; 4种算法中BP算法误差值最大, 相对于CPSO算法, 相差3个数量级, BP算法的全局搜索能力弱于粒子群算法; BP误差曲线只有1个点, 表明其以后的训练过程为发散过程, 系统训练被提前终止; CPSO算法收敛于最小误差值, 提高了MPIDNN控制系统的控制精度与跟踪速度, 取得了最优控制效果.

4.2.2 跃变值系统输入(System input with step values)

第2种系统输入如式(17)所示:

$$\begin{cases} r_1 = 0.55 + 0.1(k - 150), \\ r_2 = 0.7 - 0.2(k - 50). \end{cases} \quad (18)$$

与恒定值输入相比, 第2种系统输入给定值发生了一些变化, 即 r_1 与 r_2 输入值分别在第150和50采样点时刻发生了跃变. 系统的仿真曲线如图4所示.

与图3比较可知, 前50个采样点时间内, 系统输入未发生改变, 粒子群及其改进算法下的跟踪曲线却发生了变化, 意味着CPSO、CRPSO与PSO算法的全局搜索方向是动态变化的; 而BP算法下的跟踪曲线未发生变化, 表明其搜索方向是单一

不变的, 与CPSO、CRPSO与PSO算法相比, BP算法的全局搜索能力较弱. 系统输入变化增加了系统控制难度, 与图3相比, 4种算法下的控制器控制性能均变坏, 表明了被控系统强耦合性与非线性本质. 由图4可见, 与其他3种控制器相比, MPIDNN_CPSO控制器使控制系统仍然具有最优的跟踪控制效果, 表明MPIDNN_CPSO控制器使系统具有最强的鲁棒性. MPIDNN_BP控制器的训练步数仍然保持为一步, 在调整了学习步长与初始化权值后, 依旧不能改变, 表明MPIDNN_BP控制系统适应能力较弱.

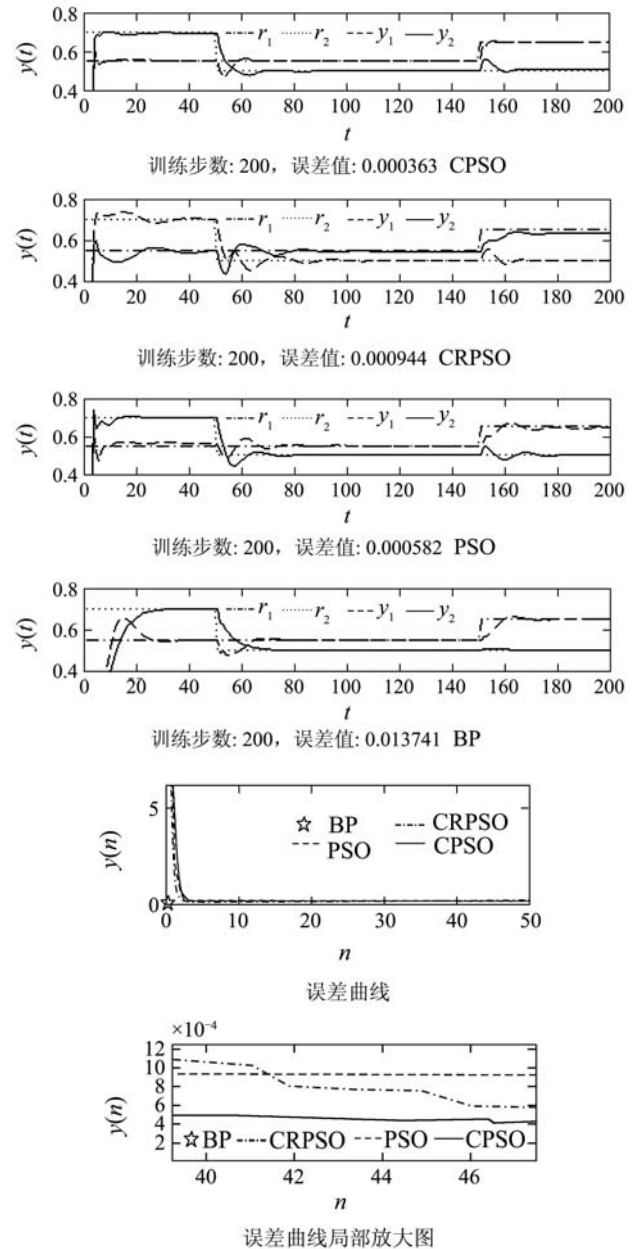


图4 跃变输入值下的MPIDNN控制系统仿真曲线图
Fig. 4 The simulation curves of MPIDNN control system with step system inputs

4.2.3 正余弦值系统输入(System input with sine and cosine values)

第3种系统输入为正余弦值输入:

$$\begin{cases} r_1 = 0.5 + 0.2 \cdot \sin(2 \cdot \pi \cdot t/100), \\ r_2 = 0.5 + 0.2 \cdot \cos(2 \cdot \pi \cdot t/100). \end{cases} \quad (19)$$

系统仿真结果如图5所示。

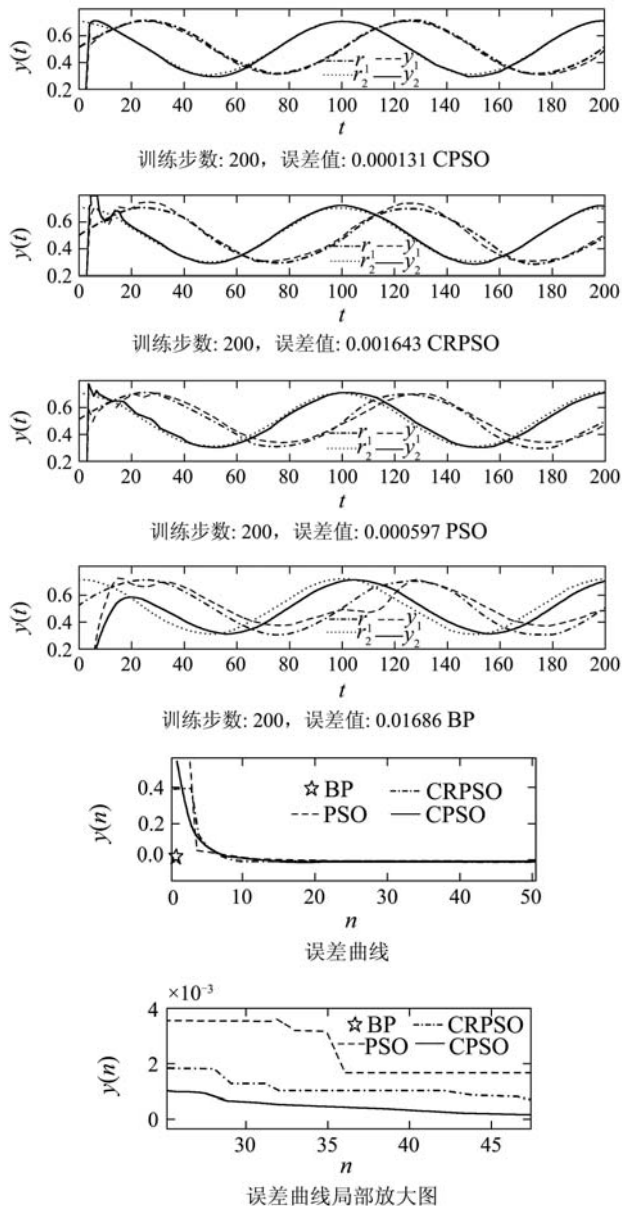


图5 正余弦输入下的MPIDNN控制系统仿真曲线图

Fig. 5 The simulation curves of MPIDNN control system with sine-cosine system inputs

由图5可知, 对于正余弦输入, MPIDNN_BP控制系统跟踪性能很不理想. 为改善其跟踪效果, 需重新选择初始化权值, 这表明初始化权值对控制结果有重大影响. 同时, 初始化值的重新选取, 不能保证控制结果. 由图3、4、5可知, 3种输入下,

MPIDNN_BP控制系统均只完成一步训练, 误差值均为最大, 跟踪效果随输入变化有较大起伏, 表明MPIDNN_BP控制系统稳定性较差, 若非强迫中止训练, 则误差发散, 最终导致系统不稳定, 从而控制器失去对系统的控制基础. 与此相反, 对应不同系统输入, 粒子群及其改进算法均使误差曲线具有衰减收敛特性, 从而保障了MPIDNN控制系统的控制稳定性, 其中, MPIDNN_CPSO控制系统具有最强的系统稳定特性.

5 结论(Conclusion)

非线性是系统的本质特性. 多变量非线性系统的控制难于实现. PIDNN神经网络定义了比例(P)神经元、积分(I)神经元和微分(D)神经元, 对控制系统而言, 提供了新的控制途径与工具. 但BP后向传播算法限制了PIDNN神经网络的应用, 不能有效控制多变量非线性系统. 为实现对此类非线性多变量系统的控制, 本文采用PIDNN神经网络设计了MPIDNN控制器, 并用本文作者提出的CPSO合作粒子群算法取代了传统BP后向传播算法, 通过比较MPIDNN_CPSO、MPIDNN_CRPSO、MPIDNN_PSO和MPIDNN_BP4种控制器的控制性能, 仿真结果表明, 基于CPSO算法的MPIDNN控制系统实现了对非线性多变量不对称系统的有效控制. 与传统的BP算法相比, CPSO算法提高了控制系统的稳定性、精确性与鲁棒性. MPIDNN_CPSO控制器为控制复杂的非线性系统提供了一个新的选择.

参考文献(References):

- [1] FARDADI M, SELK GHAFARI A, HANNANI S K. PID neural network control of SUT building energy management system[C] // *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. New York: IEEE Press, 2005: 682 – 686.
- [2] SHU H L, SHU H. Simulation study of PID neural network temperature control system in plastic injecting-moulding machine[C] // *Proceedings of the 6th International Conference on Machine Learning and Cybernetics*. New York: IEEE Press, 2007: 492 – 497.
- [3] KENNEDY J, EBERHART R C. Particle swarm optimization[C] // *Proceedings of IEEE International Conference on Neural Networks*. New York: IEEE Press, 1995: 1942 – 1948.
- [4] LI M, YANG C W. A modified PSO learning algorithm for PID neural network[C] // *Proceedings of the 25th Chinese Control Conference*. Beijing: Beijing University of Aeronautics and Astronautics Press, 2006: 1123 – 1125.
- [5] CHEN J Y, ZHENG Q. Particle swarm optimization with Local Search[C] // *Proceedings of International Conference on Neural Networks and Brain*. New York: IEEE Press, 2005: 481 – 484.

- [6] SHI Y, KENNEDY R C. A modified particle swarm optimizer[C] // *Proceedings of the IEEE Congress on Evolutionary Computation*. New York: IEEE Press, 1998: 69 – 73.
- [7] ZHAO L, YANG Y P. PSO-based single multiplicative neuron model for time series prediction[J]. *Expert Systems with Applications*, 2009, 36(2): 2805 – 2812.
- [8] ZOU J, FU X. A particle swarm optimization approach of PID parameters in hydraulic servo control system[C] // *Proceeding of the 6th World Congress on Intelligent Control and Automation*. New York: IEEE Press, 2006: 7725 – 7729.
- [9] 冯远静, 俞立, 冯祖仁. 采样粒子群优化模型及其动力学行为分析[J]. *控制理论与应用*, 2009, 26(1): 28 – 34.
(FENG Yuanjing, YU Li, FENG Zuren. Sample particle swarm op-

timization and its dynamic behavior[J]. *Control Theory & Applications*, 2009, 26(1): 28 – 34.)

作者简介:

朴海国 (1977—), 男, 博士研究生, 目前研究方向为兆瓦级风力发电机组偏航控制系统研究, E-mail: phg0805@sjtu.edu.cn;

王志新 (1964—), 男, 教授, 博士, 博士生导师, 目前研究方向为风力发电及其控制、智能机电控制系统、制造执行系统;

张华强 (1967—), 男, 教授, 博士, 目前研究方向为阵变换器的控制策略、基于矩阵变换器的交流励磁变速恒频风力发电技术研究.

下 期 要 目

- 量子分子动力学中的操纵技术及其系统控制理论 丛 爽
- 基于信息融合的汽车衡称重传感器故障诊断 林海军, 滕召胜, 迟 海, 吴阳平, 易 钊
- 基于Petri网的柔性制造系统无死锁遗传调度算法 任 磊, 王 峰, 邢科义
- 基于改进PCA-ELM方法的穿孔机导盘转速测量 肖 冬, 王继春, 潘孝礼, 毛志忠, 常玉清
- 改进自适应变空间差分进化算法 姚 峰, 杨卫东, 张 明, 李仲德
- 电基于LS-SVR的机器人空间4DOF无标定视觉定位 辛 菁, 刘 丁, 徐庆坤
- 基双重控制策略在流媒体连续播放中的应用 邹 勇, 赵 海, 魏立峰, 周 艳
- 包含SVC和非线性负荷的电力系统耗散实现与控制 刘艳红, 宋伟华, 王 杰
- 比例再保险问题的奇异型最优控制模型 袁继红
- 水雷运动变时滞区间系统保性能非脆弱 H_∞ 控制 肖 敏, 史忠科
- 两阶段过程模型下的产品线复用策略优化 吴志樵, 唐加福, 周进刚
- 群体动力学与协调控制研究中的若干问题 楚天广, 杨正东, 邓魁英, 王 龙, 谢广明
- 基于耦合度分析的多座不对称焦炉集气管压力模糊解耦控制 吴 敏, 周国雄, 雷 琪, 曹卫华
- 具有条件马尔科夫结构的离散随机系统最优控制 方洋旺, 王洪强, 伍友利
- 基于分数阶微积分 PD^λ 比例导引制导规律 王 飞, 雷虎民