基于改进型粒子群算法的 PID 神经网络控制系统①

沈学利，徐 涛

(辽宁工程技术大学 电子与信息工程学院，葫芦岛 125105)

摘 要：针对传统的 PID 神经网络（PIDNN）应用范围受限及积分误差规则难以获取的问题。为实现对非线性多变量系统的有效控制，拓展神经网络控制系统的应用范围，提出了基于改进型粒子群算法在 PID 神经网络控制系统设计中的解决方案，取代了传统的 BP 反向传播算法。仿真结果表明，与传统的 PIDNN 相比，系统的稳定性、鲁棒性及精确性都有了明显的提高，该方法有效的提高了 PIDNN 控制的使用范围，为智能方法在 PID 控制中的应用提出了一个新的参考。

关键词：PID 神经网络；改进型粒子群算法；非线性控制系统；稳定性；精确性

PID Neural Network Control System Based on Improved Particle Swarm Optimization Algorithm

SHEN Xue-Li, XU Tao

(Electronic and Information Engineering College, Liaoning Technology University, Huludao 125105, China)

Abstract: The traditional PID neural network (PIDNN) limited the scope of application and integration problems are difficult to obtain the error rule. For the realization of nonlinear multivariable control systems, neural network control system to expand the application range of this paper, based on improved version particle swarm optimization algorithm for PID neural network control system design solution, replacing the traditional BP back the propagation algorithm, simulation results show that compared with traditional PIDNN, the steady-state system, robustness and accuracy have improved obviously, this method is effective to improve the use of PID control, intelligent method for the PID Control proposed a new reference.

Key words: PID neural network; improved version particle swarm optimization algorithm; nonlinear control systems; stability; accuracy

1 引言

传统 PID 控制在工业控制系统中具有广泛的应用，其控制结构简单，易于在线调整参数，主要适用于线性控制系统。然而现实工业环境中被控对象往往机理复杂，具有很强的非线性、时变不确定性和纯滞后等特点，常规的 PID 控制器存在参数整定不良、工矿适应性差等问题[1]。为实现有效控制，获得优良的控制性能，神经网络被引入控制领域，神经网络具有非线性特征，可实现自适应控制，适合解决非线性问题。但是直到现在，人工神经网络仍未在现实控制系统中得到广泛的应用，对此，文献[1]给出了具体的原因：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；BP 算法全局搜索能力弱，易陷入局部最优；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[2]。此外，为获得优良控制性能，神经网络层数也难于抉择。

针对上述问题，笔者提出一种新型的 PID 神经网络(PIDNN)模型，此模型基于改进型的粒子群算法(IPSO),与 BP、PSO 算法相比在精确性、搜索能力、稳

① 基金项目:2009 年度中国煤炭工业科技计划(MTKJ2009-240)
收稿时间:2011-03-09;收到修改稿时间:2011-04-06

定性、鲁棒性等方面都有了明显的提高。有助于PIDNN 在控制领域的广泛应用。

2　多变量PIDNN（MPIDNN）控制系统

为实现非线性不对称 MIMO 系统的解耦控制，建立了 MPIDNN 控制系统[1]。MPIDNN 由多个单变量 PID 神经网络结构（SPIDNN）子网交叉并联而成。如果被控对象为 M 个输入，N 个输出，该 MPIDNN 就有 2N 个输入单元，隐含层有 3N 个处理单元，构成 2N×3N×M 结构的网络。它的输入层到隐含层是按子网独立的，而其隐含层至输出层的连接权则是相互交叉连接的，使整个多输出 PIDNN 结合为一体。其结构如图 1 所示。

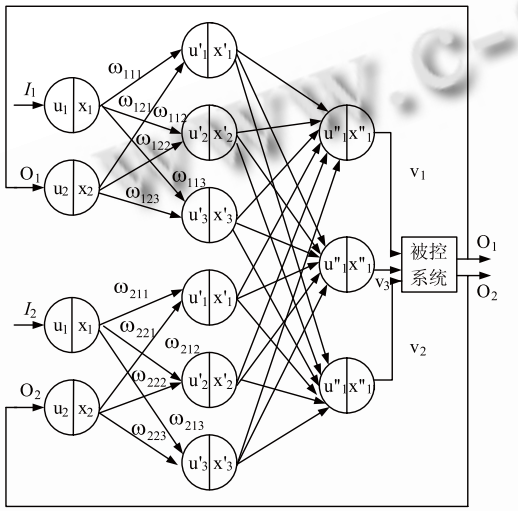

图 1　MPIDNN 控制系统拓扑结构图

由图 1 可知，被控系统是 3 输入 2 输出的多变量系统，层间的连接权值分别定义为 ω_{sij} 和 ω_{sjh}，其中 s 为 SPIDNN 的数目。MPIDNN 的价值函数为：

$$J=\sum_{s=1}^{2}E_s=\frac{1}{m}\sum_{s=1}^{2}\sum_{k=1}^{m}\left[I_s(k)-O_s(k)\right]^2 \tag{1}$$

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。式子中：m 为采样点的总数目；$I_s(k)$、$O_s(k)$ 分别为 SPIDNN 控制系统的输入与输出。

3　反向传播(BP)算法

1982 年，Rumelhard 和 Mcclelland 等人提出了著名的 BP 算法，其思想是使用梯度搜索理论，以使得网络实际输出（计算输出）与期望输出（目标输出）的均方差达到最小。网络的学习过程是，将输出层误差反向传播回去，并借以修正权值。图 1 的权值修改公式[2]为：

$$\Delta\omega_{ij}=\eta\frac{\partial J}{\partial\omega_{sij}}=$$

$$\begin{cases}-\frac{2}{m}\sum_{s=1}^{2}\sum_{h=1}^{3}\sum_{k=1}^{m}\left[I_s(k)-O_s(k)\right]\bullet\frac{\partial O_s}{\partial x''_h}\bullet\omega_{sih}(k)\bullet 1\bullet x_{si} & P_{neuron}\\ -\frac{2}{m}\sum_{s=1}^{2}\sum_{h=1}^{3}\sum_{k=1}^{m}\left[I_s(k)-O_s(k)\right]\bullet\frac{\partial O_s}{\partial x''_h}\bullet\omega_{sih}(k)\bullet u'_{sj}\bullet x_{si} & I_{neuron}\\ -\frac{2}{m}\sum_{s=1}^{2}\sum_{h=1}^{3}\sum_{k=1}^{m}\left[I_s(k)-O_s(k)\right]\bullet\frac{\partial O_s}{\partial x''_h}\bullet\omega_{sih}(k)\bullet\frac{\partial x'_{sj}}{\partial u'_{sj}}\bullet x_{si} & D_{neuron}\end{cases} \tag{2}$$

$$\Delta\omega_{jh}=\eta\frac{\partial J}{\partial\omega_{sjh}}=-\frac{2}{m}\sum_{s=1}^{2}\sum_{k=1}^{m}\left[I_s(k)-O_s(k)\right]\bullet\frac{\partial O_s}{\partial x''_h}\bullet x'_{sj}(k) \tag{3}$$

式(3)中：

$$\frac{\partial O_s}{\partial x''_h}=\mathrm{sgn}\left[\frac{O_s(k+1)-O_s(k)}{x''_h(k)-x''_h(k-1)}\right] \tag{4}$$

式(2)中：

$$\frac{\partial x'_{sj}}{\partial u'_{sj}}=\mathrm{sgn}\left[\frac{x'_{sj}(k)-x'_{sj}(k-1)}{u'_{sj}(k)-u'_{sj}(k-1)}\right] \tag{5}$$

由式 2 可以看出系统的非线性特性及其输入输出之间的强耦合特性。

4　粒子群算法(PSO)

4.1 PSO 算法基本原理

PSO 算法是 Kennedy J 和 Eberhart R 在 1995 年提出的，该算法是模拟鸟群的飞行觅食行为，通过个体之间的协作来搜寻最优解。设 N 维的搜索空间中，有 m 个粒子，其中第 i 个粒子的位置向量为 $X_i=(x_{i1},x_{i2},\cdots,x_{iN})$，i=1，2，…，m，粒子的速度向量为 $V_i=(v_{i1},v_{i2},\cdots,v_{iN})$。第 i 个粒子经过的最好位置（最优解）为 P_{best}，粒子群的最好位置记为 G_{best}。根据 Kenddy 等提出的 PSO 算法公式：

$$\begin{aligned}V_{id}(t+1)=v_{id}(t)+c_1\bullet Rand_1\bullet(P_{best}-x_{id}(t))\\+c_2\bullet Rand_2\bullet(G_{best}-x_{id}(t))\end{aligned} \tag{6}$$

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \tag{7}$$

其中，$Rand_1$ 和 $Rand_2$ 是[0，1]之间的随机数；c_1 和 c_2 为加速因子，在每一维，粒子都有一个最大限速度 V_{max}(V_{max}>0)，如果某一维速度超过设定的 V_{max}，则该维速度被限定为 V_{max}。

Shi 和 Eberhart 在 1998 年的论文中引入了惯性权重的概念，将速度更新方程修改为：

$$\begin{aligned}V_{id}(t+1)=\omega\bullet v_{id}(t)+c_1\bullet Rand_1\bullet(P_{best}-x_{id}(t))\\+c_2\bullet Rand_2\bullet(G_{best}-x_{id}(t))\end{aligned} \tag{8}$$

其中，ω 为惯性权重，可见基本 PSO 算法的惯性权重 $\omega=1$。

4.2 改进型 PSO 算法（IPSO）

虽然基本的PSO具有算法收敛速度快和简便易行等优点，但由于所有粒子采用统一的惯性权重，降低了粒子的多样性，易于早熟收敛，搜索精度不高，易陷入局部最优解。本文改进值 ω 和 c_1、c_2 的值，研究惯性权值 ω 对优化性能的影响，发现较大的 ω 值有利于全局搜索且收敛速度快，而较小的 ω 值有利于局部搜索但收敛速度慢。因此，提出了一种自适应改变 ω 值的非线性方法。算法在初期能长时间保持较大惯性权值，具有较强全局搜索能力，后期则能长时间保持较小惯性权值，提高局部搜索能力。

惯性权 ω 值的公式为：

$$\omega(iter)=\omega_{\max}-(\omega_{\max}-\omega_{\min})e^{\frac{iter}{it_{\max}-1}} \quad (9)$$

其中，$\omega_{\max}$ 取值为 1.2，$\omega_{\min}$ 取值为 0.4，$it_{\max}$ 为粒子群的最大迭代数，iter 为当前迭代数。随着迭代次数增加，ω 值非线性地减小，可以较快的速度搜索和寻找最优值。

采用动态调整 c_1 和 c_2 的方法，算法开始时设置较大的 c_1 和较小的 c_2，让粒子初期在自身邻域内的全局最优解；末期设置较小的 c_1 和较大的 c_2 ,可以使粒子更快速，更准确地收敛于全局最优解。因此，将学习因子 c_1 构造成随着进化推进而单调递减的函数，c_2 构造成随着进化推进而单调递增的函数，并通过对前人所设置的学习因子固定值进行仿真实验。加速因子 c_1 和 c_2 的改进公式为：

$$c_1=\frac{2}{1+iter^{0.25}} \quad (10)$$

$$c_2=\frac{iter}{it_{\max}} \quad (11)$$

为了说明改进算法的优越性，下面用经典的异或问题(XOR)对传统的 BP 算法，PSO 算法和 IPSO 算法分别进行实验测试，并比较实验结果。其中，本实验采用三层网络，输入层节点 2 个，隐含层节点数 50 个，输出层 1 个。共有 4 个样本，输入向量为 P=[0,0; 1,1; 0,1; 1,0]，目标向量为 T=[0,0,1,1]，最大迭代数为 400。实验结果如表 1 所示。

表 1 异或实验结果对比

算法	隐含层节点数	平均迭代次数	期望误差 (10^{-2})	平均 CPU 时间（ms）
BP	14	300	0.99	580
PSO	12	80	0.01	300
IPSO	9	40	0.001	160

由表 1 可知，通过三种算法的比较，IPSO 的平均迭代次数和平均 CPU 时间明显减少。因此，该实验结果说明本文所提出的 IPSO 算法优于传统 BP 和 PSO 算法。

4.3 IPSO 算法流程

Step 1: 初始化 MPIDNN 与 IPSO 的全部参数；定义速度、位置的维数与范围。初始化所有子群粒子的速度与位置。

Step 2: 计算所有粒子的适应值，即价值函数值，选出个体极值最好的作为全局极值。若所有粒子的个体极值中最好的优于当前全局极值，记为 Gbest ，记录该最好值的粒子序号 ID，那么该粒子对应的极值就是下次迭代中神经网络的最优权值。

Step 3: 计算出每个粒子的适应度值，若优于该粒子当前的个体极值，则将 ID 设置为该粒子的位置，且更新个体极值。若所有粒子的个体极值中最好的优于当前全局极值，则将 Gbest 设置为该粒子的位置，记录该粒子的序号，且更新全局极值。

Step 4: 根据式(9)更新惯性权重，根据式(10)和式(11)更新加速因子。

Step 5: 根据式(6)和(7)更新位置和速度。

Step 6: 检验迭代是否达到最大次数或最小误差要求时，若是停止迭代，全局极值对应的神经网络权值与连接结构，即为训练问题的最优解；否则，转到 Step 3

4.4 MPIDNN 控制系统的参数给定

MPIDNN-IPSO 控制系统的参数给定如下：

s=2，h=3，q=1，q′ =1,采样数=160，最大循环步数=40，子群数=3，每一个子群中粒子数=35；ω_{sij} 输入层与中间层的权值范围=[-2 2]，ω_{sjh} 中间层与输出层的权值范围=[-2.5 2.5]，权值解空间维数=35；设定 $Xmin_{sij}$= -2，$Xmax_{sij}$=2，$Xmin_{sjh}$= -2.5，$Xmax_{sjh}$=2.5；速度的最大值和最小值分别设定为：$Vmin_{sij}$=-0.15，$Vmax_{sij}$=0.15，$Vmin_{sjh}$= -0.15，$Vmax_{sjh}$=0.15；每一子群的位置与速度初始化公式[2]可表示为：

$$\begin{cases} pos = X\min+(X\max-X\min)\bullet rand \\ vel = V\min+2\bullet V\max\bullet rand \end{cases} \quad (12)$$

式中 rand 为 0 和 1 之间的随机数值。

MPIDNN-PSO 控制系统的参数给定，除粒子数等于 80 外，其余都与 MPIDNN-IPSO 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=160，最大循环步数 4000，学习步长 η =0.01；各层之间的权值初始化值

为：

ω_{sij} (1：s，1，1)=2.5，ω_{sij} (1：s，2，1)= -2.5，

ω_{sij} (1：s，1，2)=1.5，ω_{sij} (1：s，2，2)= -1.5，

ω_{sij} (1：s，1，3)=2，ω_{sij} (1：s，2，3)= -3，

ω_{sjh} (1：s，1：3，1：h)=0.1，ω_{sjh} (1，1：3，1：3)= 0.2，

5 仿真与分析

5.1 被控系统函数和控制系统参数

被控系统的方程为：

$$\begin{bmatrix} Y_1(t) \\ Y_2(t) \end{bmatrix} = \begin{bmatrix} \dfrac{0.6423t^{-1}}{1-0.4315t^{-1}} & \dfrac{0.5432t^{-1}}{1-0.7231t^{-1}} & \dfrac{0.2176t^{-1}}{1-0.8745t^{-1}} \\ \dfrac{0.1134t^{-1}}{1-0.7234t^{-1}} & \dfrac{0.6783t^{-1}}{1-0.6453t^{-1}} & \dfrac{0.4365t^{-1}}{1-0.7326t^{-1}} \end{bmatrix} \bullet \begin{bmatrix} V_1^2(t) \\ V_2(t) \\ V_3^2(t) \end{bmatrix} \tag{13}$$

通过两种典型的输入，分别对三种算法下的 MPIDNN 控制器的控制效果进行了 MATLAB 仿真实验

1) 恒定值系统输入：

系统的输入设定为：

$$\begin{cases} I_1 = 0.52 \\ I_2 = 0.65 \end{cases} \tag{14}$$

此时的仿真曲线为：

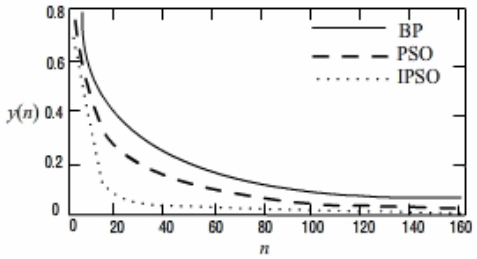

图 2　恒定值输入值下的 MPIDNN 控制系统误差仿真曲线

2) 跃变值系统输入

系统的输入设定为：

$$\begin{cases} I_1 = 0.6 + 0.15(k-120) \\ I_2 = 0.8 - 0.3(k-50) \end{cases} \tag{15}$$

与恒定值输入不同，第 2 种系统输入的设定值发生了一些变化，即 I_1 和 I_2 输入值分别在第 120 和第 50 采样点时刻发生了跃变。

此时的仿真曲线为：

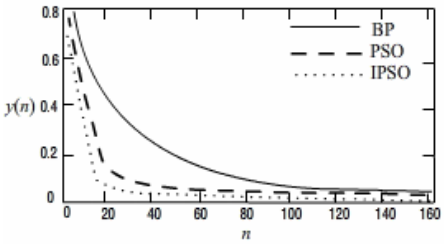

图 3　跃变值输入值下的 MPIDNN 控制系统误差仿真曲线

5.2 仿真分析

由图 2 和图 3 可知，BP、PSO、IPSO 算法误差曲线的衰减，表明搜索一直向着最优方向进行。三种算法中 BP 算法的误差值最大，全局搜索能力最弱，系统输入变化增加了系统控制难度。IPSO 具有最强的全局搜索能力，算法收敛于最小误差值。因此，IPSO 算法提高了 MPIDNN 控制系统的控制精度与跟踪速度，取得了最优的控制效果。

6 结语

反向传播（BP）算法限制了 PIDNN 神经网络的应用，不能有效地控制多变量非线性控制系统，为了实现此类系统的有效控制，本文提出以改进型粒子群算法 IPSO 替代 BP 算法，仿真结果表明，基于 IPSO 算法的 MPIDNN 控制系统实现了对多变量非线性系统的有效控制，提高了系统的稳定性、精确性及鲁棒性。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得问题，MPIDNN-IPSO 控制系统为复杂的多变量非线性控制环境提供了一个新的参考。

参考文献

1 赵俊,陈建军.混沌粒子群优化的模糊神经 PID 控制器设计.西安电子科技大学学报(自然科学版),2008,35(1):54–60.

2 朴海国,王志新,张华强.基于合作粒子群算法的 PID 神经网络非线性控制系统.控制理论与应用,2009,26(12):1317–1324.

3 Mudi RK, Pal Nr. A Self-tuning Fuzzy PI Controller. Fuzzy Sets and Systems, 2000,115(3):327–338.

4 高金源.计算机控制系统.北京:清华大学出版社,2010.50–65.

5 钱厚斌,张天平.控制增益符号未知的 MIMO 时滞系统自适应控制.控制与决策,2008,23(10):1153–1159.

6 李铁山,邹早建,罗伟林.基于 DSC 后推法的非线性系统的鲁棒自适应 NN 控制.自动化学报,2008,34(11):1424–1430.

7 王维杰,李东海,高琪瑞,王传峰.一种二自由度 PID 控制器参数整定方法.清华大学学报(自然科学版),2008,48(11):1962–1966.