

# 基于 BP 算法 PID 控制器的研究

朱喜娜, 陆 达, 范汉青

(厦门大学 计算机系统结构研究所, 福建 厦门 361005)

**摘要:**文中主要研究了基于 BP 算法的 PID 控制器在非线性系统中的控制效果以及对权值整定初始化的优化。在介绍 BP 网络基本原理的基础上以非线性控制系统 Simulink 仿真为例, 使用基于 BP 算法的 PID 控制器对该系统进行优化和整定, 并结合 Nguyen - Widrow 初始化算法为层产生初始权重和偏置值, 使得每层神经元的活动区域能大致平坦地分布在输入空间, 实现了对 PID 参数的实时调节, 并且使神经网络的学习和收敛速度加快, 大大改善系统的初始运行的稳定性。仿真结果表明, 基于 BP 算法的 PID 控制器在非线性控制系统中对其参数优化整定具有良好的效果。

**关键词:**BP; PID; 非线性系统; Nguyen - Widrow 算法

中图分类号: TP342

文献标识码: A

文章编号: 1673 - 629X(2010)05 - 0183 - 04

## Research of PID Controller Based on BP Algorithm

ZHU Xi-na, LU Da, FAN Han-qing

(Department of Computer Science, Xiamen University, Xiamen 361005, China)

**Abstract:** Studies the effects of the BP algorithm based on PID controller in the nonlinear system as well as setting the right values to initialize the optimization. In this paper take example for Simulink simulation of a non-linear control systems, use PID controller based on BP algorithm to optimize the system and setting, which combined with Nguyen - Widrow initialization algorithm, realize the real-time adjustment of the PID parameters, and accelerate the neural network learning convergence. Simulation results show that PID controller based on BP algorithm in the nonlinear control system for its tuning parameter has a good effect.

**Key words:** BP algorithm; PID; nonlinear systems; Nguyen - Widrow algorithm

## 0 引言

工业控制领域中, PID 控制是最早发展、最为常用的控制策略之一, 这是因为这种控制方式具有算法简单、容易实现、鲁棒性好、可靠性高等一系列优点。为了更好地应用 PID 控制, PID 控制器参数的整定与优化已成为一个重要的研究课题。随着现代工业系统非线性复杂程度的提高和被控对象不确定因素的增加, 线性 PID 控制器往往难以达到令人满意的控制效果, 而非线性 PID 控制却能真实地反映控制量与偏差信号之间的非线性关系, 在一定程度上克服了线性 PID 控制器的不足, 因此越来越受到控制界的关注<sup>[1]</sup>。

## 1 基于 BP 网络的 PID 控制算法

### 1.1 PID 控制算法

在模拟控制系统中, 控制器最常用的控制规律是

收稿日期: 2009 - 09 - 15; 修回日期: 2009 - 12 - 17

作者简介: 朱喜娜(1985 - ), 女, 青海西宁人, 硕士研究生, 研究方向为计算机自动控制; 陆 达, 教授, 硕士生导师, 研究方向为计算机控制。

PID 控制。模拟 PID 控制系统原理框图如图 1 所示。整个系统由 PID 控制器和被控对象组成。

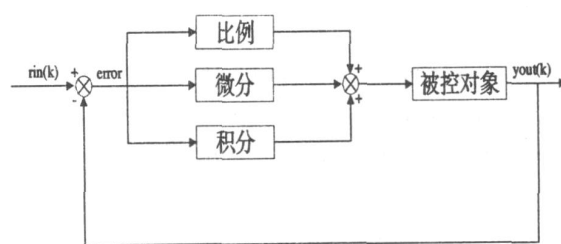


图 1 PID 控制原理图

PID 控制器是一种线性控制器, 它根据给定值  $r_{in}(t)$  和实际输出值  $y_{out}(t)$  构成控制偏差:

$$error(t) = r_{in}(t) - y_{out}(t)$$

PID 控制规律为:

$$u(t) = K_p(error(t)) + \frac{1}{T_I} \int_0^t error(t) dt + \frac{T_D d error(t)}{dt}$$

式中  $K_p$  为比例系数,  $T_I$  为积分时间常数,  $T_D$  为微分时间常数。

$$\text{又有 } K_I = \frac{K_p T_s}{T_I}, K_D = \frac{K_p T_D}{T_s}$$

式中  $K_I$  为积分放大系数,  $K_D$  为微分放大系数。

### 1.2 基于 BP 算法的 PID 控制器

根据使用简单 PID 控制器整定参数不够理想这一现状, 现结合 BP 神经网络算法对 PID 控制器进行整定以实现控制算法的优化。

PID 控制要取得较好的控制效果, 就必须通过调整好比例、积分和微分三种控制作用, 形成控制量中既相互配合又相互制约的关系。神经网络所具有的任意非线性表达能力, 可以通过对系统性能的学习来实现具有最佳组合效果的 PID 控制。采用 BP 网络, 可以建立参数  $K_p, K_I, K_D$  自学习的 PID 控制器<sup>[2]</sup>。

基于 BP 网络的 PID 控制器是由经典的 PID 控制器和神经网络两部分构成<sup>[3]</sup>。结构如图 2 所示。其中经典的 PID 控制器直接对被控对象进行闭环控制, 并且三个参数  $K_p, K_I, K_D$  为在线调整方式; 而神经网络部分根据系统的运行状态调节 PID 控制器的参数, 以达到某种指标的最优化, 得到输出层神经元的输出状态对应于某种最优控制下的 PID 控制器可调的参数。

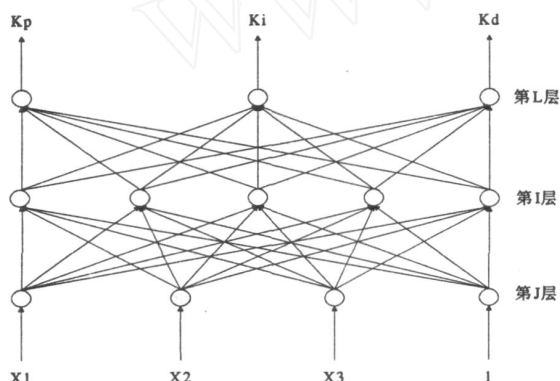


图 2 神经网络拓扑图

整个算法分二步进行, 即正向传播和反向传播。对于正向传播部分, 输入的样本从输入层经过隐单元一层一层进行处理, 通过所有的隐层之后, 最后传向输出层<sup>[4]</sup>。

网络输入层的输入设为:

$$O_j^{(1)} = x(j) \quad (j = 1, 2, \dots, M)$$

其中  $M$  为输入变量的个数, 取决于被控对象的复杂程度。

网络隐含层各神经元的输入、输出为:

$$\begin{aligned} \text{net}_i^{(2)}(k) &= \sum_{j=0}^M w_{ij}^{(2)} O_j^{(1)} \\ O_i^{(2)}(k) &= f(\text{net}_i^{(2)}(k)) \quad (i = 1, 2, \dots, Q) \end{aligned}$$

式中,  $w_{ij}^{(2)}$  为隐含层加权系数, 上角标 (1)、(2)、(3) 分别代表输入层、隐含层和输出层。其中隐层神经元活化函数取正负对称的 Sigmoid 函数:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

网络输出层的输入输出为:

$$\begin{aligned} \text{net}_l^{(3)}(k) &= \sum_{i=0}^Q w_{li}^{(3)} O_i^{(2)}(k) \\ O_l^{(3)}(k) &= g(\text{net}_l^{(3)}(k)) \quad (l = 1, 2, 3) \end{aligned}$$

其中:  $O_l^{(3)}(k)$ 、 $O_l^{(2)}(k)$  和  $O_l^{(1)}(k)$  即为输出层输出节点对应的三个可调参数  $k_p, k_i$  和  $k_d$ 。输出层神经元的活化函数取非负的 Sigmoid 函数:

$$g(x) = \frac{1}{2}(1 + \tanh(x)) = \frac{e^x}{e^x + e^{-x}}$$

在逐层处理的过程中, 每一层神经元的状态只对下一层神经元的状态产生影响。在输出层把现行输出和期望输出进行比较, 如果现行输出不等于期望输出, 则进入反向传播过程。反向传播时, 把误差信号按原来正向传播的通路反向传回, 并对每个隐层的各个神经元的权系数进行修改, 以使误差信号趋向最小。

取性能指标函数为:

$$E(k) = \frac{1}{2}(\text{rin}(k) - \text{yout}(k))^2$$

最后根据梯度下降法修正网络的权系数, 即按  $E(k)$  对加权系数的负梯度方向搜索调整, 并附加一个使搜索快速收敛的全局极小的惯性项:

$$w_{li}^{(3)}(k) = - \frac{\partial E(k)}{\partial w_{li}^{(3)}} + w_{li}^{(3)}(k-1)$$

经过推导计算可得网络的输出层学习算法为:

$$w_{li}^{(3)}(k) = \text{net}_l^{(3)}(k) O_i^{(2)}(k) + w_{li}^{(3)}(k-1)$$

$$\begin{aligned} \delta_l^{(3)} &= \text{error}(k) \text{sgn}\left(\frac{\partial y(k)}{\partial u(k)}\right) \\ &\cdot \frac{\partial u(k)}{\partial O_l^{(3)}(k)} g'(\text{net}_l^{(3)}(k)) \end{aligned}$$

同理可得隐含层加权系数的学习算法为<sup>[5]</sup>:

$$w_{ij}^{(2)}(k) = \delta_l^{(3)} O_j^{(1)}(k) + w_{ij}^{(2)}(k-1)$$

$$\delta_i^{(2)} = f'(\text{net}_i^{(2)}(k)) \sum_{l=1}^3 \delta_l^{(3)} w_{li}^{(3)}(k)$$

( $i = 1, 2, \dots, Q$ )

式中,  $g'(\cdot) = g(x)(1 - g(x))$ ,  $f'(\cdot) = (1 - f(x))/2$ 。

神经元网络连接权重初值的选取对整个网络的学习和收敛速度是很重要的, 因为连接权重初值决定了网络学习的起始点和收敛的初始方向。通常的初始权重与偏差随机化方法都是在区间  $(-1, 1)$  之间取均匀分布的随机数的函数, 计算表明, 该方法对复杂的高非线性控制系统初始运行的稳定性很难得到保证。

恰当地选择连接权重的初值, 可以使网络的学习和收敛速度加快且能避免陷入局部极小, 大大改善系统的初始运行的稳定性。

实验表明, Nguyen - Widrow 法对计算效率的提高有较明显的效果。

其算式为<sup>[6]</sup>:

$$W = 0.7s^{\perp} \text{normr}(2\text{rand}(s, r) - I(s, r))$$

$$B = 0.7s^{\perp} (2\text{rand}(s, 1) - I(s, r))$$

式中, W 为权重矩阵, B 为偏差矩阵, s 和 r 分别为网络隐含层数和输入样本的行数, rand(s, r) 为 s 行 r 列的均匀分布的随机数矩阵, I(s, r) 为 s 行 r 列的全 1 矩阵, normr(M) 为 M 矩阵的标准化归一矩阵。根据 Nguyen - Widrow 初始化算法为层产生初始权重和偏置值,使得每层神经元的活动区域能大致平坦地分布在输入空间。

## 2 S 函数

### 2.1 S 函数简介

由于 MATLAB/SIMULINK 仿真系统中没有可用的基于 BP 算法的 PID 控制器模块,所以利用 S 函数(S-Function)可以很容易地解决这个问题。Simulink 是一种图形化仿真工具包,能够进行动态系统建模、仿真和综合分析,可以处理线性和非线性控制系统,离散、连续和混合系统,以及单任务和多任务系统,具有非常高的开放性。

S-function 为用户提供了一种在 Simulink 模型中增加自制块的手段,使用户在复杂的系统中可以用 MATLAB 语言、Fortran 语言、C/C++ 语言等来描述具体的过程,构成 S 函数模块,然后在 Simulink 模型中通过 S 函数模块直接调用,从而利用 S 函数增强和扩展了 Simulink 的能力<sup>[7]</sup>。

S 函数有固定的编写格式,在 MATLAB 中自带了默认的模板,用户只要按需求修改相应的部分即可。

### 2.2 函数原理

Simulink 块包含一组输入、一组状态和一组输出。其中,输出是采样时间、输入和块状态的函数。可以用以下表达式表示:

$$y = f_0(t, x, u)$$

其中 t 为采样时间, x 为状态, u 为输入。

Simulink 模型的执行分几个阶段进行<sup>[8]</sup>: 首先进行的是初始化阶段,在此阶段,Simulink 将库块合并到模型中来,确定传送宽度、数据类型和采样时间,计算块参数,确定块的执行顺序,以及分配内存。然后,Simulink 进入到“仿真循环”,每次循环可认为是一个“仿真步”。在每个仿真步期间,Simulink 按照初始化阶段确定的块执行顺序依次执行模型中的每个块。对于每个块而言,Simulink 调用函数来计算块在当前采样时间下的状态、导数和输出。如此反复,一直持续到仿真结束。

## 3 系统仿真

该系统被控对象为一非线性环节,在其近似数学模型为:

$$yout(k) = \frac{a(k)yout(k-1)}{1 + yout^2(k-1)} + u(k-1)$$

式中,系数 a(k) 是慢时变的,  $a(k) = 1.2(1 - 0.8e^{-0.1k})$ 。

被控对象仿真模型如图 3 所示。

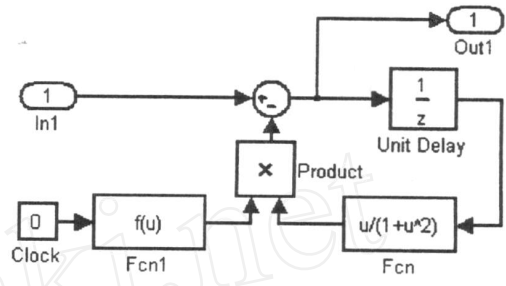


图 3 被控对象仿真模型

经计算,在 0s 时为系统输入阶跃信号,并为系统加入 1% 的随机扰动,通过对系统输出信号进行分析可知,此时该系统的最大超调量为 22.5%,稳态误差达到 0.09。

当该系统经过基于 BP 算法的 PID 控制器调节后的系统模型如图 4 所示。

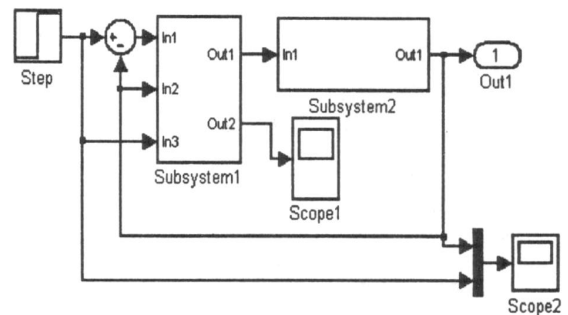


图 4 控制系统模型

其中基于 BP 算法的 PID 控制器封装在 BP-PID 子系统中, In1 为给定输入与系统输出的系统误差  $error(t)$ , In2 为系统输出  $yout(t)$ , In3 为系统给定输入  $rin(t)$ . Out1 为控制器输出  $u(k)$ , Out2 为  $K_p$ 、 $K_i$ 、 $K_d$  三个权值的变化量。非线性控制对象被封装在 subsystem2 子系统中,其子系统输入为控制器的输出,子系统输出为系统输出。

经过控制器控制后系统输出波形细节图如图 5 和图 6 所示。

通过仿真,得到经过基于 BP 算法的 PID 控制器调节后系统超调量 < 5%,稳态误差接近于 0。可以说,该控制器对于非线性系统的调节作用达到了很好的效果。

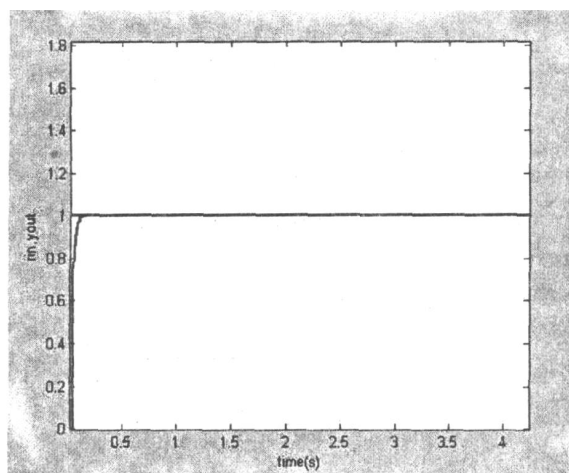


图 5 系统输出

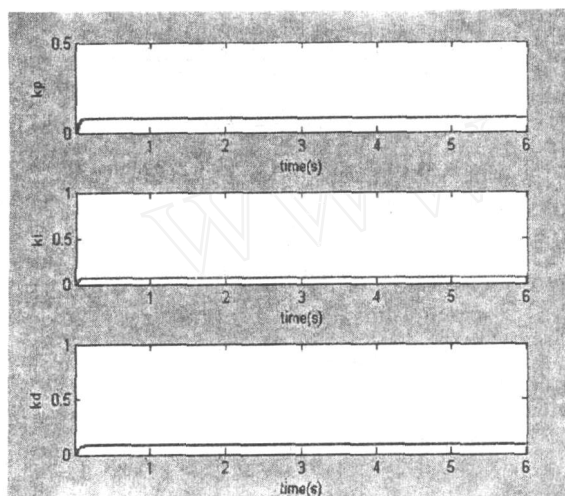


图 6  $K_p, K_i, K_d$  输出波形

### 4 结束语

仿真结果显示,用 S 函数实现的 BP 神经网络 PID 控制系统,并结合 Nguyen - Widrow 初始化算法,对非线性控制信号跟踪效果十分理想,实现了对 PID 参数的实时调节,使网络的学习和收敛速度加快,并且避免系统调节陷入局部极小,大大改善系统的初始运行的稳定性。

#### 参考文献:

- [1] 玉鑫,段宝岩.一种新型非线性 PID 控制器[J].控制与决策,2003,18(1):126-128.
- [2] 舒怀林.PID 神经网络及其控制系统[M].北京:国防工业出版社,2006.
- [3] 薛定宇.控制系统计算机辅助设计[M].北京:清华大学出版社,2006.
- [4] 刘金琨.先进 PID 控制 MATLAB 仿真[M].北京:电子工业出版社,2004.
- [5] 盛华,李润梅.计算机控制系统[M].北京:清华大学出版社,2007.
- [6] Abraham A. Neuro - fuzzy systems: state of the art modeling techniques[C]// Proc of the 6th International Conference on Artificial and Neural Networks. Berlin:Springer,2001:269-276.
- [7] 黄永安,马路,刘慧敏. MATLAB 7.0/Simulink 6.0 建模仿真开发与高级工程应用[M].北京:清华大学出版社,2005.
- [8] 庞清乐,孙同景,穆健.基于 PID 控制谐振接地系统的研制[J].继电器,2006,34(1):24-35.

(上接第 182 页)

### 3 结束语

高校学生资助工作意义重大。需要从全局角度审视整个工作机制,研究管理决策的流程,探讨反馈机制和评价体系,应充分发挥信息技术推动管理变革的重要作用。网络环境下基于 UML 的高校学生资助工作管理系统的研发,有助于科学规范高校学生资助工作的管理行为,提高管理水平和工作效率,进而改变传统的手工或半手工学生资助工作管理模式。为高校学生资助工作走向正规化、标准化和信息化提供了科学借鉴。同时,为应用 UML 建模技术进行面向对象软件开发提供了参考范例,实践表明 UML 建模技术能有效提高软件开发质量和开发效率。

#### 参考文献:

- [1] 国务院文件.关于建立健全普通本科高校、高等职业学校和中等职业学校家庭经济困难学生资助政策体系的意见.

国发[2007]13号[S].2007.

- [2] 段新娥.高校学生信息集成管理的必要性分析[J].山西高等学校社会科学学报,2008(3):120-122.
- [3] 徐宝文,周毓明,卢红敏.UML 与软件建模[M].北京:清华大学出版社,2006.
- [4] Boggs W,Boggs M.UML with Rational Rose 从入门到精通[M].邱仲潘,译.北京:电子工业出版社,2000.
- [5] 谢黎文.高校经济困难学生资助管理模式研究与信息系统实现[D].成都:西南交通大学,2006:50-54.
- [6] 张凌晓,刘克成.基于 UML 的全程办税系统的建模与实现[J].计算机技术与发展,2008,18(10):210-213.
- [7] 葛瑶,李晓风,王辉.基于 .NET 分布式系统模式研究[J].计算机工程与设计,2008,29(5):1049-1051.
- [8] 李律松,马传宝,李婷. Visual C# + SQL Server 数据库开发与实例[M].北京:清华大学出版社,2006.
- [9] 李志,赵韶平,罗海燕. PowerDesigner 系统分析与建模[M].北京:清华大学出版社,2004.