

文章编号: 1001-0920(2003) 03-0367-04

# 一种基于并行遗传算法的非线性系统辨识方法

蒙祖强, 蔡自兴

(中南大学 信息科学与工程学院, 湖南 长沙 410083)

**摘要:** 结合并行算法的运算能力和遗传算法的搜索能力, 提出一种基于并行遗传算法的非线性系统辨识方法。其特点是通过并行遗传算法实现对 RBF 神经网络权值、宽度和中心位置等有关参数的估计, 其速度快、精度高, 从而通过 RBF 神经网络有效地完成了对非线性系统的辨识。最后给出了仿真结果, 证实了该方法的有效性。

**关键词:** 辨识; 非线性系统; 并行遗传算法; RBF 神经网络

中图分类号: TP18

文献标识码: A

## Identification method of nonlinear systems based on parallel genetic algorithm

MENG Zu-qiang, CAI Zi-xing

(College of Information Science & Engineering, Central South University, Changsha 410083, China)

**Abstract:** Combining calculation capability of parallel algorithm and search capability of genetic algorithm, an identification method of nonlinear systems based on parallel genetic algorithm is presented. This method is characterized by estimating parameters such as weight, width and central position of RBF NN using parallel genetic algorithm. High velocity and accuracy of the method enable nonlinear systems to be efficiently identified by using RBF NN. The simulation results show the effectiveness of this method.

**Key words:** Identification; Nonlinear system; Parallel genetic algorithm; RBF NN

### 1 引言

神经网络具有很好的函数逼近能力<sup>[1,2]</sup>, 在系统辨识方面具有重要的应用<sup>[3~6]</sup>。但是, 传统神经网络学习算法(如 BP 算法)基本上是基于梯度法原理, 学习能力不强, 速度不够理想, 带有很大的盲目性, 存在局部最优等问题。对于高度非线性系统, 传统学习算法更显得力不从心。而并行遗传算法对于非线性问题求解和大范围空间搜索等却有独特的优势。

本文提出以 RBF 神经网络为辨识器, 用并行遗传算法对其参数进行估计, 克服了传统学习算法的

缺点, 实现了对非线性系统的辨识。最后给出的仿真结果证实了该方法的有效性。

### 2 基于 RBF NN 的辨识问题建模

对于 RBF 神经网络, 设其输入向量  $X = (x_1, x_2, \dots, x_n)$ , 基函数为高斯函数

$$g_i(X) = \exp(-(X - C_i)^T \Delta_i (X - C_i))$$

其中

$$C_i = (C_{i1}, C_{i2}, \dots, C_{in})$$

收稿日期: 2001-12-31; 修回日期: 2002-04-01。

基金项目: 国家自然科学基金重点资助项目(60234030); 国家博士点基金资助项目(99053317); 湖南省自然科学基金资助项目(99JJY20062)。

作者简介: 蒙祖强(1974—), 男, 广西罗城人, 博士生, 从事并行处理技术、遗传算法、机器学习等研究; 蔡自兴(1938—), 男, 教授, 博士生导师, 从事人工智能、智能系统、智能控制等研究。

$$\Delta = \begin{bmatrix} 1/\sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & 1/\sigma_m^2 \end{bmatrix}, \quad i = 1, 2, \dots, m$$

$n > 0$  为输入向量  $X$  的分量个数,  $m > 0$  为隐含层神经元个数。则该网络的输出为

$$F(X) = \prod_{i=1}^m [g^i(X) w_i] / \prod_{i=1}^m g^i(X) \quad (1)$$

其中:  $w_i$  为权值,  $c_{ij}$  为中心位置,  $\sigma_j$  为宽度,  $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ 。

$F(X)$  是由参数  $w_i, c_{ij}, \sigma_{ij}$  决定的。令

$$\Omega = \{ (w_1, \dots, w_m, c_{11}, \dots, c_{1n}, \dots, c_{m1}, \dots, c_{mn}, \dots, \sigma_{11}, \dots, \sigma_{1n}, \dots, \sigma_{m1}, \dots, \sigma_{mn}) \mid w_i, c_{ij}, \sigma_{ij} \in \mathbf{R} \quad i = 1, 2, \dots, m, j = 1, 2, \dots, n \} \quad (2)$$

则可用  $F_{\theta}(X)$  来表示  $F(X)$  和各参数的关系, 其中  $\theta \in \Omega$ , 待估计参数个数为  $2mn + m$ 。

考虑待辨识的非线性系统  $\Gamma$ , 假设训练样本集

$\zeta = \{X_k \mid k = 1, 2, \dots, N\}$ ,  $N$  为样本集的大小。对  $X_k \in \zeta$  经过待辨识系统  $\Gamma$  的作用后其输出记为  $\Gamma(X_k)$ , 则该样本  $X_k$  对于  $\theta$  的平方误差为  $[\Gamma(X_k) - F_{\theta}(X_k)]^2/2$ , 样本集  $\zeta$  对于  $\theta$  的平方误差为

$$E(\theta, \zeta) = \frac{1}{2N} \sum_{k=1}^N [\Gamma(X_k) - F_{\theta}(X_k)]^2 \quad (3a)$$

由于只有一个样本集, 故把  $E(\theta, \zeta)$  简记为  $E(\theta)$ , 即

$$E(\theta) = \frac{1}{2N} \sum_{k=1}^N [\Gamma(X_k) - F_{\theta}(X_k)]^2 \quad (3b)$$

这样, 辨识问题就可建模为式(3), 辨识任务即为: 寻找  $\theta \in \Omega$ , 使得  $E(\theta)$  充分小。显然, 这是一个参数优化问题, 本质上是对 RBF 神经网络参数进行估计。

### 3 RBF NN 辨识器的参数估计

作为非线性系统的辨识器, RBF 神经网络隐含层的神经元个数较大, 待估计参数较多, 而且取值范围亦未知。传统的学习算法难以直接对这些众多参数进行有效的估计。一般先进行概率化处理, 减少待估计参数个数。这虽然达到简化问题的目的, 但也造成了极大的误差, 严重影响了模型的准确性。这种做法其实就是传统神经网络学习方法的无奈之举, 因为它无法胜任待估计参数较多的情况。因此, 要对较多的参数进行有效估计, 必须寻找一种具有很强的运算和处理能力的算法, 而并行遗传算法在处理非线性问题和巨量运算等方面却有独特的优势。以下设计一个能完成这些参数估计的并行遗传算法。

#### 3.1 用于参数估计的遗传算法设计

1) 编码。编码可用二进制或实数, 但考虑到辨

识的对象是非线性系统, 隐含层的神经元个数  $m$  较大, 使得待估计参数个数  $2mn + m$  就很大。如果采用二进制编码, 那么码字就很长, 这与精度要求相矛盾<sup>[7]</sup>。所以, 本文采用实数编码。实际上, 式(2)已给出了编码形式, 即一个码串具有形式

$$(w_1, \dots, w_m, c_{11}, \dots, c_{1n}, \dots, c_{m1}, \dots, c_{mn}, \dots, \sigma_{11}, \dots, \sigma_{1n}, \dots, \sigma_{m1}, \dots, \sigma_{mn}) \quad (4)$$

其中:  $w_i, c_{ij}, \sigma_{ij} \in \mathbf{R}, i = 1, 2, \dots, m, j = 1, 2, \dots, n$ 。对于一个具体的实数  $w_i, c_{ij}, \sigma_{ij}$ , 式(4)就表示一个个体, 记为  $\theta$ 。显然  $\theta \in \Omega$ 。

2) 群体设定。用  $P = \{\theta_k \mid \theta_k \in \Omega, k = 1, 2, \dots, P_{size}\}$  表示群体,  $P_{size}$  为群体规模。为保持个体多样性, 群体规模  $P_{size}$  应较大, 这样有利于克服早熟问题。但规模过大, 会导致计算量增加, 降低算法效率; 规模过小, 算法易陷入局部最优。故  $P_{size}$  取值应适中。但此概念仍较模糊, 实际应用中只能依据经验或实验确定, 尚无理论上的指导<sup>[8]</sup>。本文取  $P_{size} = 200$ 。

3) 适应度函数。适应度函数是算法对个体“好坏”的评价准则, 取值为正。这里直接取目标函数(3)作为适应度函数  $f(\theta)$ , 即

$$f(\theta) = E(\theta) = \frac{1}{2N} \sum_{k=1}^N [\Gamma(X_k) - F_{\theta}(X_k)]^2 \quad (5)$$

其中  $\theta \in P$ 。

#### 4) 遗传算子。

① 选择: 本文采用 Monte Carlo 选择法。

② 交叉: 令  $\theta = (\theta_1, \theta_2, \dots, \theta_n) \in \Omega, i = 1, 2, \dots, n$ , 则交叉操作定义为

$$\theta_s = \theta \otimes_{ram} \theta$$

$$\theta_{sj} = \begin{cases} \alpha\theta_j + (1 - \alpha)\theta_{2j}, & ram \leq 0.5 \\ \alpha\theta_{2j} + (1 - \alpha)\theta_j, & ram > 0.5 \end{cases}$$

其中:  $j = 1, 2, \dots, n; \alpha \in (0, 1)$  随机选取;  $ram$  是  $[0, 1]$  上的随机数;  $\otimes_{ram}$  表示交叉操作符<sup>[9]</sup>。由于该交叉只产生一个后代个体, 所以为保持群体规模  $P_{size}$  不变, 可用该子个体替换适应值较低的一个父本。

③ 变异: 它主要完成搜索空间的可达性, 避免算法陷入局部最优。方法是随机选取个体的某一个基因位, 然后对该位加上一个随机数。据系统的非线性特性, 其变异概率取 0.05。

#### 3.2 遗传算法的并行化方法

并行遗传算法主要有 4 种类型<sup>[10]</sup>, 考虑到我们使用的是由 PVM 提供的基于消息传递的并行虚拟环境, 故选择了粗粒度并行模型(CPGA), 也称孤岛模型。其主要思想就是把一个群体分成若干个子群

体, 并把这些子群体分别发送到各孤岛上, 然后彼此间独立进化, 当满足一定条件(如时间间隔或遗传代数)时, 孤岛间进行个体互换, 之后又进化, 循环直到算法收敛。孤岛间的个体交换方式是遗传算法并行化的关键技术之一, 文献[11]中已给出了一种随机的交换方式, 如图 1 所示(注: 作为一个例子, 图中  $P_i$  表示孤岛,  $i = 1, 2, \dots, 5$ ; 虚线表示可能数据交换方式)。本文采用此交换方式, 虽然在编程实现时比较麻烦, 但能更好地克服早熟问题<sup>[11]</sup>。

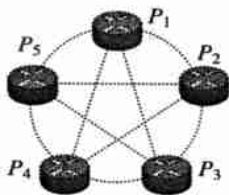


图 1 改进的 CPGA 结构

### 3.3 基于并行遗传算法的非线性系统辨识算法

由以上讨论, 系统辨识算法的总框架可用图 2 描述。

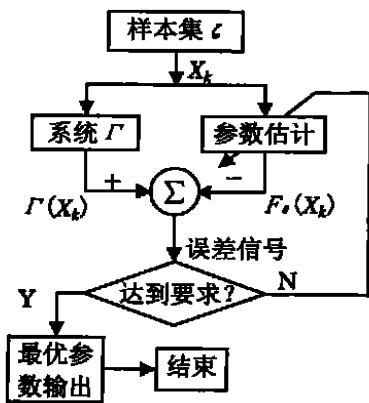


图 2 系统辨识的基本框架

采用 master/slave 的并行编程模式, 相应的算法可描述如下:

输入: 样本集  $\zeta$ ;

输出: 最优参数向量  $\theta \in \Omega$

begin

master(); /\* 启动 master 进程 \*/

init(P); /\* master 初始化群体  $P \in \Omega$  \*/

slave(); /\* master 启动若干个 slave 进程 \*/

divide(); /\* 把群体  $P$  分为若干子群体, 分别发送给各 slave 进程, 以完成相应的处理 \*/

finish = false; /\* 各进程置本地终止标记 finish 为 false \*/

```

par while(not finish); /* 各进程并行完成下列操作 */
{sum = 0; /* 当前子群体 theta in Omega 根据当前子群体和样本集 zeta 计算 sum */
  f(theta);
}
select(sum); /* 根据 sum 和每个个体的适应值进行蒙特卡罗选择 */
cross(); /* 交叉 */
mutate(); /* 变异 */
if (满足互换条件) then 进行孤岛间随机的个体交换;
if (自己是 master 进程) then
{if (达到精度要求) then
  {输出当前最优个体 theta;
   本地 finish = true;
   向各 slave 进程发出终止信号;
  }
}
}

```

```

else if(收到 master 进程发来的终止信号) then
  本地 finish = true;
} // end par while
end

```

该算法可以完成对 RBF NN 辨识器的参数估计, 从而可利用这一辨识器对非线性系统进行辨识。

### 4 仿真结果

考虑如下非线性系统<sup>[3]</sup>

$$y(t + 1) = 0.3y(t) + 0.6y(t - 1) + g[u(t)] \quad (6)$$

未知函数的形式为

$$g(u) = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u)$$

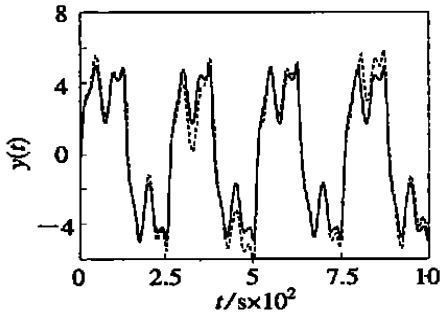
系统输入为

$$u(t) = \sin(2\pi t / 250)$$

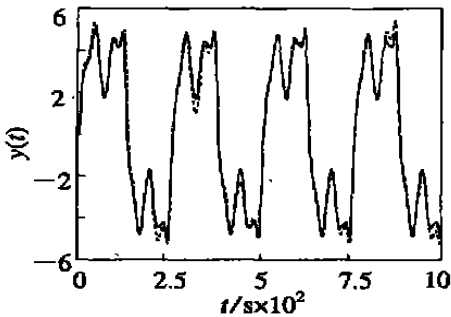
$$\text{令 } X_t = [x_1(t), x_2(t), x_3(t)]^T = [y(t), y(t - 1), u(t)]^T$$

由式(6)产生 800 个样本  $X_k$ , 即样本集  $\zeta = \{X_k | k = 1, 2, \dots, 800\}$ 。仿真是在由 PVM 提供的基于消息传递的并行虚拟环境(由 6 台 PC 机组成)下进行, 程序开始时先启动 master 进程, 然后由 master 进程启动 24 个 slave 进程(即采用 master/slave 模式), 每一

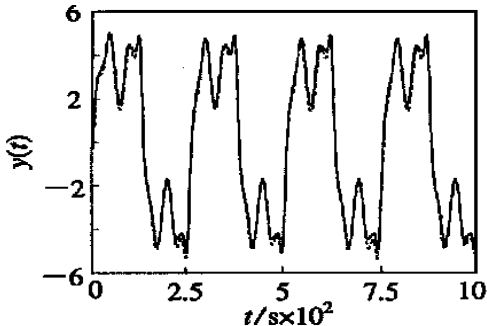
进程的作用相当于上文提到的孤岛,其子群体每经 25 代的进化就进行一次最优个体交换,每次交换 10 个当前最优个体。实验中各子群体的交叉概率取 0.7,变异概率取 0.05,结果如图 3 所示。



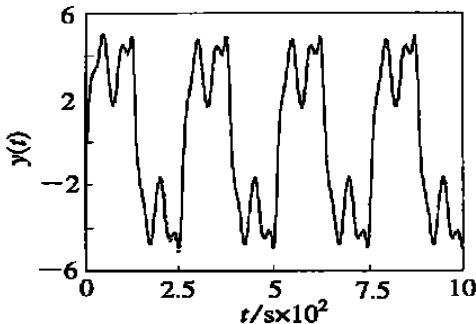
(a) Master 第 1 次提交参数后显示的结果



(b) Master 第 5 次提交参数后显示的结果



(c) Master 第 10 次提交参数后显示的结果



(d) Master 第 20 次提交参数后显示的结果

图 3 辨识器输出(虚线)和系统真实输出(实线)的对比

从图 3 中可看出,当 master 进程在第 20 次提交当前最优参数时,辨识误差已很小,RBF 网络很好地逼近了被测系统;与文献[3]得到当  $t = 500$  时发散的结果相反,该辨识算法不但不发散,而且算法收敛过程稳定,收敛速度快,结果非常满意。这说明此种全方位的参数估计方法有其独到之处,同时也证实了本文的非线性系统辨识方法的有效性。

## 5 结 论

介绍了用并行遗传算法对 RBF 神经网络的权值、宽度和中心位置等有关参数进行快速、高精度的估计,使得 RBF 网络对未知非线性系统有充分的函数逼近能力,从而实现了 RBF 神经网络辨识器对非线性系统的有效辨识。可将该方法进一步推广,如:考虑有多个隐含层的神经网络结构,对其所有参数采用同样的估计方法,其结果将会更理想,应用更广泛。这为解决参数较多、结构复杂的大型非线性系统辨识问题提供了一种可行的方法,也为非线性系统的自适应控制等有关方面的研究打下了基础,在国防工业和生产控制等方面有着重要的应用价值。

## 参考文献(References):

- [1] Cybenko G. Approximation by superposition of sigmoidal functions, mathematics of control[J]. *Signals and Systems*, 1989, 2: 303-314.
- [2] Chen T P, Chen H, Liu R W. Approximation capability in  $C^n$  by multilayer feedforward networks and related problems[J]. *IEEE Trans NN*, 1995, 6(1): 57-67.
- [3] Narendra K S, Parthasarathy K. Identification and control of dynamical systems using neural networks[J]. *IEEE Trans NN*, 1990, 1(1): 4-27.
- [4] Narendra K S, Parthasarathy K. Gradient methods for the optimization of dynamical systems containing neural networks[J]. *IEEE Trans NN*, 1991, 2(1): 252-262.
- [5] Hunt K J, Sbarbaro D. Neural networks for nonlinear internal model control[J]. *IEEE Proceeding-D*, 1991, 138(5): 431-438.
- [6] Fukuda T, Shibata T. Theory and application of neural networks for industrial control systems [J]. *IEEE Trans Industrial Electronics*, 1992, 39(6): 472-489.
- [7] 张晓绩,方浩,戴冠中. 遗传算法的编码机制研究[J]. *信息与控制*, 1997, 26(2): 134-139.  
(Zhang Xiaohui, Fang Hao, Dai Guangzhong. Studies on encoding mechanism of genetic algorithms [J]. *Information and Control*, 1997, 26(2): 134-139.)

(下转第 374 页)

SIEMENS 的 S7 系列 PLC 的语句表指令实现,而在工业计算机中则以 C 语言实现,这些 ACL 消息都要根据 PROFIBUS-FMS 协议转化为网络变量进行传输。

### 3.2 实验与分析

在 ICMMS 平台对水轮机组进行了多种工况下的模拟实验,人为设置各种在常规机组测频环节和导叶/桨叶电液伺服机构易发生的故障,结果表明现地监控 Agent 能够对这些故障作出及时的反应,监测与预测 Agent 和诊断与预诊断 Agent 也能及时获得各个现场设备的状态并据此对设备的性能作出较准确的判断和预报。同时实验表明技术管理 Agent 能够实现对调速系统运行的优化管理,并且完成了对维护活动的管理。各个 Agent 进行协作,共同完成了控制、维护和技术管理功能的集成。

## 4 结 语

ICMMS 是生产过程自动化系统的发展方向,多 Agent 系统是一种建立在分布自治基础上的智能、功能集成方法,利用多 Agent 技术来实现生产过程自动化系统的控制、维护和技术管理功能的集成具有实际应用价值。本文提出了一种适用于底层自动化系统的控制、维护、技术管理功能集成的多 Agent 模型,利用该方法首先实现生产过程中各个子系统的控制、维护和技术管理功能的集成,进而通过各个子系统间的协作实现整个生产过程的控制、维护和技术管理功能的集成。该多 Agent 模型在 ICMMS 平台中的应用已初步显示其有效性和可行性。

### 参考文献(References):

[1] Ye Luqing, Li Zhaohui, Liu Yongqian, et al. Intelli-

gent control-maintenance-management system and its application on hydropower[A]. *Proc 2000 IFAC Management and Control of Production and Logistic Conf* [C]. UK: Elsevier Science, 2001. 2: 609-614.

[2] 傅闯,王生铁,叶鲁卿. 智能控制-维护-技术管理集成系统中的智能化问题[J]. *水电能源科学*, 1999, 17(1): 9-12.

(Fu Chuang, Wang Shengtie, Ye Luqing. The intelligent of an integrated intelligent control-maintenance-management system [J]. *Int J of Hydroelectric Energy*, 1999, 17(1): 9-12.)

[3] Wooldridge M J, Jennings N R. Intelligent agents: Theory and practice [J]. *Knowledge Engineering Review*, 1995, 10(2): 115-152.

[4] Vidal J M, Buhler P A, Huhns M N. Inside an agent [J]. *IEEE Internet Computing*, 2001, 5(1): 82-86.

[5] Eleni E M angina, Stephen D J McArthur, J R McDonald, et al. A multi-agent system for monitoring industrial gas turbine start-up sequences[J]. *IEEE Trans Power Systems*, 2001, 16(3): 396-401.

[6] Wang H F. Multi-agent co-ordination for the secondary voltage control in power-system contingencies[J]. *IEEE Proc- Generation, Transmission and Distribution*, 2001, 148(1): 61-66.

[7] Tapio Heikkilä, Martin Kollingbaum, Paul Valckenaers, et al. An agent architecture for manufacturing control: Manage [J]. *Computers in Industry*, 2001, 46(3): 315-331.

[8] Fischer Klaus. Agent-based design of holonic manufacturing systems [J]. *Robotics and Autonomous System*, 1999, 27(1): 3-13.

[9] FIPA. specifications[EB/OL]. <http://www.fipa.org/>. 2002. 10. 12.

(上接第 370 页)

[8] Rudolph G. Convergence properties of canonical genetic algorithms[J]. *IEEE Trans NN*, 1994, 5(1): 96-101.

[9] 陈明. 基于进化遗传算法的优化计算[J]. *软件学报*, 1998, 9(11): 876-879.

(Chen Ming. Optimization computing based on evolution genetic algorithm[J]. *J of Software*, 1998, 9(11): 876-879.)

[10] 王洪燕, 杨敬安. 并行遗传算法研究进展[J]. *计算机科学*, 1999, 26(6): 48-53.

(Wang Hongyan, Yang Jing an. The advance on research of parallel genetic algorithm[J]. *Computer Science*, 1999, 26(6): 48-53.)

[11] 蒙祖强, 陈振州, 郑金华. 基于共享存储器通信方式的并行遗传算法[J]. *计算机工程与应用*, 2000, 36(5): 72-74.

(Meng Zuqiang, Chen Zhenzhou, Zheng Jinhua. The parallel genetic algorithms based on the way of communication by shared memory[J]. *Computer Engineering and Applications*, 2000, 36(5): 72-74.)