

基于神经网络的工业大系统辨识及 稳态递阶优化方法¹⁾

李玉桥 万百五

(西安交通大学系统工程研究所 西安 710049)

梁天培

(香港理工学院工程学院 香港)

摘 要

为了对工业大系统进行稳态递阶优化,必须首先获得系统的稳态模型.从神经网络的分析入手,给出了工业大系统稳态模型的动态辨识方法及基于神经网络模型的推导方法.为了提高算法的收敛速度,引入 Lagrange 函数解决大系统优化问题中的各种约束,并用 Hopfield 网络实现了大系统稳态递阶优化的网络算法,最后给出了某一大系统辨识及优化的仿真结果.

关键词: 稳态递阶优化, Hopfield 神经网络,前向神经网络,工业大系统.

1 前言

在工业大系统稳态递阶优化中,子系统模型通常是未知的.用传统的辨识方法解决非线性问题存在许多困难,因而有必要研究神经网络的辨识方法.文献[1]给出了一般动态系统的神经网络辨识方法,本文将它推广到大系统问题中.对于稳态辨识问题,常用的方法是利用稳态信息直接辨识稳态模型.这种方法忽略了动态信息,为获得足够的样本不得不多次改变系统控制量的设定点,因此为实际系统运行所不取.本文提出利用一次设定点变动的动态信息完成动态系统辨识,进而利用动态模型直接求得系统稳态模型.

在优化过程中,由于传统的无导数优化算法收敛太慢,本文给出了神经网络模型的推导方法,为了进一步提高收敛速度,将 Hopfield 神经网络与 Lagrange 乘子和松弛变量相结合,获得了大系统非迭代网络优化算法.

2 大系统稳态模型的神经网络辨识及其推导方法

时变大系统稳态递阶优化任务^[2]可描述为

1) 浙江大学工业控制技术国家重点实验室基金和国家自然科学基金资助项目,本文曾在中国自动化学会第六届过程控制科学报告会上宣读.
本文于1993年11月4日收到

$$\min Q(c, u, y) = \sum_{j=1}^N \min Q_j(c_j, u_j, y_j), \quad (1)$$

$$\text{s.t. } (c_j, u_j, y_j) \in S_j, f_{ij}(y_j, c_j, u_j, t) = 0, j = 1, \dots, n_j,$$

式中

$$S_j = \{(c_j, u_j, y_j) | g_{ik}(c_j, u_j, y_j) \geq 0, k = 1, \dots, m_j;$$

$$h_{is}(c_j, u_j, y_j) = 0, s = 1, \dots, q_j; u_j = H_j y_j\},$$

$c_j \in R^{L_j}, u_j \in R^{p_j}, y_j \in R^{n_j}$, 分别是子系统 j 的控制、关联输入和输出分量, H_j 是关联矩

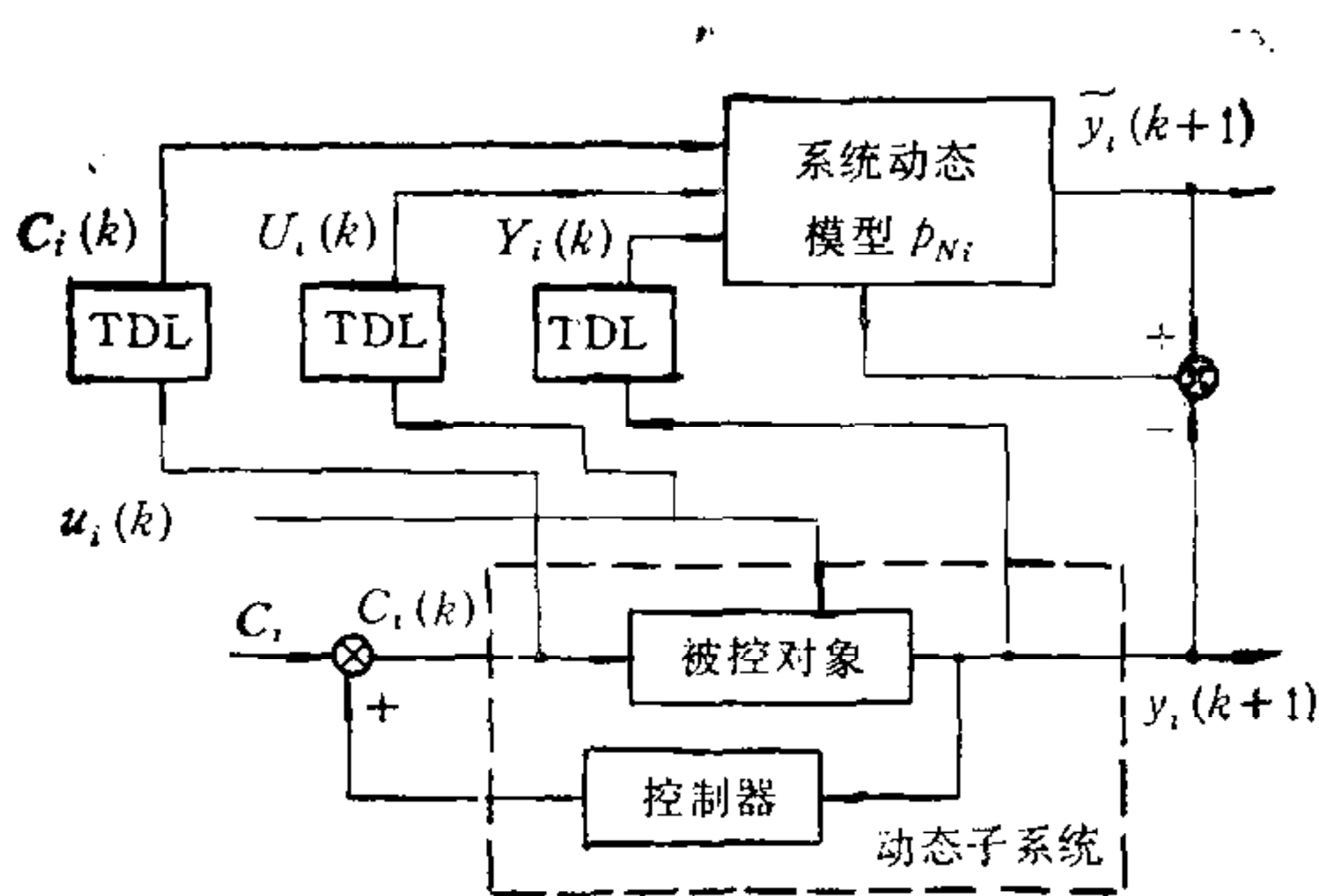


图 1 动态子系统 i 结构图

阵, $f_i(y_i, c_i, u_i, t) = 0$ 是有待辨识的慢时变子系统稳态模型, 其动态子系统如图 1 所示. 图中被控对象为 $y_i(k+1) = p_i \times (Y_i(k), C_i(k), U_i(k), k)$, 设为 γ 阶延迟系统, $(Y_i(k), C_i(k), U_i(k))$ 对应于向量 $(y_i(k), \dots, y_i(k-\gamma), c_i(k), \dots, c_i(k-\gamma), u_i(k), \dots, u_i(k-\gamma))$; 子系统控制器为 $G_i(Y_i(k))$, c_i 为控制量设定点, 且 $c_i(k) = c_i + G_i(Y_i(k))$.

1) 动态系统模型的辨识及推导. 系统动态模型的辨识方法是, 在 t^0 时刻(取 $k=0$) 将控制量设定点 c_i 加入, 而后保持不变; 对系统输入输出变量通过 γ 阶节拍延迟环节 TDL 进行采样, 取一次设定点阶跃变化的动态信息 $(Y_i(k), C_i(k), U_i(k), y_i(k+1))$, $k = \gamma, \dots, N_m$ 作为样本, 求 $p_i(Y_i(k), C_i(k), U_i(k), k)$ (见图 1). 由于系统为慢时变的, 故可认为在短时间内 p_i 与 k 无关, 而近似为 $p_i(Y_i(k), C_i(k), U_i(k))$. 采用如下前向神经网络:

$$\begin{aligned} net_{sp} &= \sum_{l=1}^{n_{p+1}} w_{slp} O_{lp-1}, & O_{n_{p+1}p} &= 1; \\ O_{sp} &= T(net_{sp}), & p &= 2, \dots, m, s = 1, \dots, n_p; \\ O_1 &= (Y_i(k), C_i(k), U_i(k)), & O_1 &\in R^{(n_i+p_i+L_i) \times \gamma}; \\ \tilde{y}_i(k+1) &= O_m, & O_m &\in R^{n_i}; \end{aligned} \quad (2)$$

式中 $net_{sp}, O_{sp}, w_{slp}$ 分别为第 p 层第 s 个神经元的输入、输出及第 $p-1$ 层第 1 个神经元对第 p 层第 s 个神经元的权系数. T 函数定义如下:

$$O_{sp} = T(net_{sp}, a_{sp}) = \sum_{i=1}^n a_{isp} (net_{sp})^i. \quad (3)$$

n 可根据对象的非线性特性选取, 训练的目的是 $\sum_{k=\gamma}^{N_m} (\tilde{y}_i(k+1) - y_i(k+1))^2$ 取极小. 因而 w 和 a 的训练算法如下:

$$\begin{aligned} a_{isp}^{(h+1)} &= a_{isp}^{(h)} - \delta_a \zeta_{sp} (net_{sp})^{i-1}, \\ w_{slp}^{(h+1)} &= w_{slp}^{(h)} - \delta_w \zeta_{sp} T'(net_{sp}) O_{lp-1}, \\ \zeta_{sp} &= \sum_{l=1}^{n_{p+1}} \zeta_{lp+1} w_{lsp+1} T'(net_{lp+1}), \\ \zeta_{sm} &= (\tilde{y}_{ic}(k+1) - y_{ic}(k+1)), \end{aligned}$$

$$p = 2, \dots, m; s = 1, \dots, n_p; l = 1, \dots, n_{p-1} + 1. \quad (4)$$

通过训练可获得 t^n 时刻子系统的动态模型如下:

$$\tilde{y}_i(k+1) = p_{Ni}(Y_i(k), C_i(k), U_i(k)), \quad (5)$$

式中 $p_{Ni} = (p_{Ni1}, \dots, p_{Nini})$.

设 $x = (Y_i(k), C_i(k), U_i(k))$, 导数 $\frac{\partial p_{Nit}}{\partial x_j}$ 可有以下式求得:

$$\begin{aligned} \frac{\partial p_{Nit}}{\partial x_j} &= T'(net_{im}) \frac{\partial net_{im}}{\partial x_j}, \quad i = 1, \dots, n_i; \quad \text{取} \quad \frac{\partial net_{i2}}{\partial x_j} = w_{siz}, \\ \frac{\partial net_{ip}}{\partial x_j} &= \sum_{l=1}^{p-1} w_{slp} T'(net_{lp-1}) \frac{\partial net_{lp-1}}{\partial x_j}, \quad p = 2, \dots, m. \end{aligned} \quad (6)$$

2) 系统稳态模型及其导数的获得. 设(5)式为一精确的子系统动态模型, 当系统处于稳态时, 必然有 $\tilde{y}_i(k+1) = \dots = \tilde{y}_i(k-\gamma) = y_i$, $u_i(k) = \dots = u_i(k-\gamma) = u_i$, $\tilde{c}_i(k) = \dots = \tilde{c}_i(k-\gamma) = c_i + G_i(Y_i)$. 将上述稳态值代入 p_{Ni} 可得一个用非线性方程表示的系统稳态模型

$$f_{Ni}(y_i, c_i, u_i) = y_i - p_{Ni}(Y_i, c_i + G(Y_i), \dots, c_i + G(Y_i), U_i) = 0, \quad (7.1)$$

式中 $(Y_i, U_i) = (y_i, \dots, y_i, u_i, \dots, u_i)$.

如果系统对于 c_i, u_i 及初值 $Y_i(\gamma)$ 存在稳态解, 则必满足(7.1)式, 但是满足(7.1)式的 y 并不一定是稳态解. 因为对于不同的 c_i, u_i 及初值 $Y_i(\gamma)$, 系统可能不是 BIBO 稳定的; 或者是 BIBO 稳定的, 但不一定有稳态解.

由上述分析可知, 要使(7.1)式能表示系统稳态模型, 必须满足两种约束: a) 稳态域约束, 它使得 c_i, u_i 及初值 $Y_i(\gamma)$ 既处在稳定域又存在稳态解; b) 可行域约束, 它给出稳态解 y_i 相对 c_i, u_i 及初值 $Y_i(\gamma)$ 的值域, t^n 时刻子系统稳态模型的约束

$$\begin{aligned} g_{Bik}(c_i, u_i, y_i, O) &\geq 0, \quad k = 1, \dots, m_{Bi}, \quad \text{稳态域}, \\ g_{Hik}(c_i, u_i, y_i, O, y_i) &\geq 0, \quad k = 1, \dots, m_{Hi}, \quad \text{可行域}. \end{aligned} \quad (7.2)$$

g_{Bik}, g_{Hik} 一般由控制器设计者给出. 由(6)式可给出稳态模型(7)式中 f_{Ni} 的导数

$$\begin{aligned} \frac{\partial f_{Nit}}{\partial y_{ij}} &= 1 - \sum_{l=1+\gamma(j-1)}^{\gamma j} \left[\frac{\partial p_{Nit}}{\partial x_l} + \sum_{k=1+\gamma n_i}^{\gamma(n_i+Li)} \frac{\partial p_{Nit}}{\partial x_k} \cdot \frac{\partial G}{\partial x_l} \right], \\ \frac{\partial f_{Ni}}{\partial v_j} &= - \sum_{k=1+\gamma(n_i+j-1)}^{\gamma(n_i+j)} \frac{\partial p_{Nit}}{\partial x_k}, \\ j &= 1, \dots, Li + pi, v = (c_i, u_i). \end{aligned} \quad (8)$$

3 基于 Hopfield 神经网络的大系统网络优化法 (NANOLS)

当通过神经网络训练获得如(7)式的子系统稳态模型后, 大系统稳态递阶优化任务(1)式就变为无导数非线性优化问题. 而通过(8)式获得子系统稳态模型的导数后, (1)式就变为有导数非线性优化问题. 但优化过程的收敛速度却较慢. 故在此引入 Hopfield 神经网络优化技术. 在用 Hopfield 神经网络求解非线性优化问题时, 常用罚函数法处理

约束。但这种方法必须取足够大的权。从而使得方程病态化，在数值求解时不得不取相当小的步长。这又使运算时间加长。为了获得较好的精度和收敛速度，文献[3]引用 Lagrange 乘子和松弛变量处理约束。本文将这一思想用于解决工业大系统稳态优化问题中，则(1)式可用如下 Lagrange 函数描述：

$$\max_{(\lambda, \eta, \zeta)} \min_{(c, u, y, z)} L(x(t)), \quad (9)$$

式中

$$\begin{aligned} L(x(t)) = & \sum_{i=1}^N L_i(c_i, u_i, y_i, \lambda, \eta_i, \zeta_i, z_i) \\ & + \sum_{k=1}^{m_{Bi}} \zeta_{Bik} \cdot (g_{Bik}(c_i, u_i, y_i) - z_{Bik}^2) \\ & + \sum_{k=1}^{m_{Hi}} \zeta_{Hik} \cdot (g_{Hik}(c_i, u_i, y_i, y_i) - z_{Hik}^2), \\ L_i = & \sum_{i=1}^N \left\{ Q_i(c_i, u_i, y_i) + \lambda_i^k u_i - \sum_{j=1}^N \lambda_j^k \cdot H_{ij} y_j \right. \\ & + \sum_{s=1}^{q_i} \eta_{is} h_{is}(c_i, u_i, y_i) + \sum_{k=1}^{m_i} \zeta_{lik} \cdot (g_{lik}(c_i, u_i, y_i) - z_{lik}^2) \\ & \left. + \sum_{j=1}^{n_i} \zeta_{2ij} f_{Nij}(y_j, c_i, u_i) \right\}. \end{aligned}$$

由(9)式，可给出大系统稳态递阶优化的非迭代网络优化法 (NANOLS) 如下：

λ ——协调网络 (λC)

$$\dot{\lambda}_i = C_\lambda^{-1} \cdot (u_i - H_i y), \quad i = 1, \dots, N;$$

子系统决策网络 (LD_i)

$$\dot{c}_i = -C_c^{-1} \cdot \left[\nabla_{c_i} L_i + \sum_{k=1}^{m_{Bi}} \zeta_{Bik} \cdot \frac{\partial g_{Bik}}{\partial c_i} + \sum_{k=1}^{m_{Hi}} \zeta_{Hik} \cdot \frac{\partial g_{Hik}}{\partial c_i} \right], \quad (10)$$

$$\dot{u}_i = -C_u^{-1} \cdot \left[\nabla_{u_i} L_i + \sum_{k=1}^{m_{Bi}} \zeta_{Bik} \cdot \frac{\partial g_{Bik}}{\partial u_i} + \sum_{k=1}^{m_{Hi}} \zeta_{Hik} \cdot \frac{\partial g_{Hik}}{\partial u_i} \right],$$

$$\dot{y}_i = -C_y^{-1} \cdot \left[\nabla_{y_i} L_i + \sum_{k=1}^{m_{Hi}} \zeta_{Hik} \cdot \frac{\partial g_{Hik}}{\partial y_i} \right],$$

$$\dot{z}_{tik} = C_z^{-1} \cdot \zeta_{tik} \cdot z_{tik}, \quad t = 1, B, H, k = 1, \dots, m_{i(B,H)},$$

$$\dot{\zeta}_{tik} = C_\zeta^{-1} \cdot (g_{tik}(c_i, u_i, y_i) - z_{tik}^2),$$

$$\dot{\zeta}_{2ij} = C_\zeta^{-1} \cdot f_{Nij}(y_j, c_i, u_i), \quad j = 1, \dots, n_i,$$

$$\dot{\eta}_{is} = C_\eta^{-1} \cdot h_{is}(c_i, u_i, y_i), \quad s = 1, \dots, q_i,$$

式中

$$\nabla_{(c)} L_i|_x = \frac{\partial Q_i}{\partial x} + \sum_{s=1}^{q_i} \eta_{is} \cdot \frac{\partial h_{is}}{\partial x}$$

$$+ \sum_{k=1}^{m_i} \zeta_{lik} \cdot \frac{\partial g_{ik}}{\partial x} + \sum_{j=1}^{n_i} \left[\zeta_{2ji} \cdot \frac{\partial f_{Nij}}{\partial x} \right].$$

C_x 为系数, 且 $C_x > 0$; f_{Nij} , $\frac{\partial f_{Nij}}{\partial x}$ 由神经网络辨识获得.

可以证明(10)式所述的动态方程组是渐近稳定的. 其稳态平衡点即(1)式在 t^* 时刻的最优解^[4].

4 仿真结果

一个含三个子系统的工业大系统如下:

$$\min Q(c, u) = \sum_{i=1}^3 \min Q_i(c_i, u_i),$$

关联项 $u = (y_{21}, y_1, y_3, y_{22})$;

$$Q_1(c_1, u_1) = (u_1 - 1)^2 + c_{11}^2 + (c_{12} - 2)^2, \quad c_{11} + u_1 - 1.006 \leq 0,$$

$$y_1(k+1) = a_{11}y_1(k) + a_{12}c_{11}(k) + a_{13}c_{12}(k) + a_{14}u_1(k) - 0.6872,$$

控制器 1 $c_{11}(k) = c_{11} - k_1y_1(k)$;

$$Q_2(c_2, u_2) = 2(c_{21} - 2)^2 + c_{22}^2 + 4u_{21}^2 + u_{22}^2 + 3c_{23}^2,$$

$$y_{21}(k+1) = a_{21}y_{21}(k) + c_{21}(k) - c_{22}(k) + a_{22}u_{21}(k) + a_{23}u_{22}(k),$$

$$y_{22}(k+1) = 1.02y_{22}(k) + a_{24}c_{22}(k) - c_{23}(k) + a_{25}u_{21}(k) + a_{26}u_{22}(k),$$

控制器 2 $c_{21}(k) = c_{21} - k_{21}y_{21}(k)$, $c_{23}(k) = c_{23} + k_{22}y_{22}(k)$;

$$Q_3(c_3, u_3) = (c_3 + 1)^2 + (u_3 - 1)^2, \quad -c_3 - u_3 - 0.5 \leq 0,$$

$$y_3(k+1) = 1.02y_3(k) + a_{31}c_3(k) + a_{32}u_3(k) - 0.125,$$

控制器 3 $c_3(k) = c_3 - k_3y_3(k)$.

对上述系统, 利用(6)式所示方法求解各子系统的导数结果见表 1, 由结果可以看出, 其精度是常用的差分法等所无法比拟的.

表 1

参 数	a_{11}	a_{12}	a_{23}	a_{31}
辨 识 值	1.0207	1.42429	0.3050	1.2495
实 际 值	1.02	1.4375	0.3	1.25

用实际系统的参数直接构成稳态子系统, 然后做大系统优化, 所得运算结果见表 2 中的实际值; 用上述辨识方法求得如(7)式的子系统稳态模型后, 用非导数罚函数法 SUMT、导数罚函数法 DSUMT 及非迭代网络优化法 NANOLS 做大系统优化, 所得运算结果见表 2. 由比较可以发现, NANOLS 运算速度比 DSUMT 和 SUMT 快得多.

基于神经网络的工业大系统稳态模型辨识及推导方法为动态过程的稳态辨识提供了新方法. 大系统优化网络法 (NANOLS) 使运算时间大大降低, 且不存在迭代, 是解决大系统优化问题的一条新途径.

表 2

	性能指标	CPU 时间 (s)	控制 c_{11}	迭代次数
SUMT	6.145642	416	0.459665	240
DSUMT	6.145663	147	0.459373	200
NANOLS	6.145629	4	0.459473	1
实际值	6.150711	—	0.459591	—

参 考 文 献

- [1] Narendra N S, Parthasarathy K. Identification and control of dynamical systems using neural network. *IEEE Trans on NN*. 1990, 1 (1): 4—27.
- [2] 李人厚, 邵福庆. 大系统的递阶与分散控制, 西安交通大学出版社, 1986.
- [3] Zhang S W, Constantindes A G. Lagrange program neural network, *IEEE Trans. on CAS*, 1992, 39 (7):441—452.
- [4] Li Y Q, Wan B W, Leung T P. Steady-state optimizing control for large-scale system with hop-field netork, The proceedings of First Asian Control Conference, Tokyo Japan, 1994, 3: 556—562.

IDENTIFICATION AND STEADY-STATE HIERARCHICAL OPTIMIZATION METHOD FOR LARGE-SCALE INDUSTRIAL SYSTEMS WITH NEURAL NETWORK

LI YUQIAO WAN BAIWU

(*Institute of Systems Engineering, Xi'an Jiaotong University Shanxi 710049*)

LIANG TIANPEI

(*Faculty of Engineering Hong Kong Polytechnic*)

ABSTRACT

In order to do steady-state hierarchical optimization for large-scale industrial systems, the steady-state model of the system must be obtained. By means of neural network, this paper presents a dynamic identification method for steady-state models of large-scale industrial systems with neural network, and proposes a way for modelling. For improving convergence, this paper firstly introduces Lagrange function to solve constraint problem in large-scale system optimization, secondly constructs the hierarchical optimization networks for large-scale industrial systems with Hopfield network.

Key words: Steady-state hierarchical optimization, Hopfield neural network, feedforward neural network, large-scale industrial system.



李玉桥 1963 年生于河北石家庄, 1984 年在第二炮兵工程学院自动控制系获学士学位, 1988 年获飞行器导航与控制专业硕士学位并任教。1991 年起, 为西安交通大学系统工程研究所博士生。现主要从事大系统递阶优化控制理论和大型飞行器仿真方面的研究。已发表文章 17 篇, 获全军科技进步一等奖 1 项。

万百五 照片、简介见本刊第 17 卷第 1 期。

梁天培 照片、简介见本刊第 17 卷第 5 期。