

动态系统模糊模型的辨识及自学习

徐承伟 吕勇哉
(昆明工学院) (浙江大学)

摘 要

本文提出 MISO 系统模糊模型的一种辨识算法及相应的自学习算法, 伴以计算示例. 辨识算法可产生具有一定精度的模型, 而自学习算法则可进一步改善模型精度.

一、引 言

为动态系统建立模糊模型的努力约始于十年前^[1], 后来的有关工作主要建基于模糊关系模型^[2-4]. 模糊模型对于模糊系统的分析、设计是不可或缺的, 一如非模糊系统的情形.

本文在文献 [3] 的基础上, 提出了一种 MISO (多输入单输出) 系统模糊模型的辨识算法; 此外还提出了一种与此辨识算法相配的自学习算法. 文末的示例有助于进一步了解算法的细节及说明算法的效果.

二、模糊模型及其辨识问题

1. 问题的提法

具有 m 个输入的 MISO 模糊系统可以表达为

$$y(t) = [y(t-1) \circ \cdots \circ y(t-n_y)] \circ [u_1(t-\tau_1) \circ u_1(t-\tau_1-1) \circ \cdots \circ u_1(t-\tau_1-n_1)] \circ \cdots \circ [u_m(t-\tau_m) \circ u_m(t-\tau_m-1) \circ \cdots \circ u_m(t-\tau_m-n_m)] \circ R, \quad (1)$$

其中 $y(\cdot)$ 为输出模糊变量; $u_1(\cdot), \cdots, u_m(\cdot)$ 为输入模糊变量; R 为基于参考集合^[3] 的模糊关系, “ \circ ”为基于参考集合的合成算子. 阶次 n_y, n_1, \cdots, n_m 和滞后 τ_1, \cdots, τ_m 是系统的结构参数. 辨识工作包括辨识结构和辨识 R 两部分. 结构辨识可以仿照线性系统的辨识方法, 在一定范围内寻找结构参数的某一组合, 使给定的指标 J 极小化. 由于模型的模糊性质, 可以预料, 模型对结构变化的灵敏度会有所下降(比非模糊模型). 实用时, 只要能满足精度要求, 不妨尽可能使用简单模型, 以减少对计算机功能的要求.

为便于表达, 记

$$\begin{aligned} x_1(t) &= y(t-1), \\ &\vdots \\ x_{n_y}(t) &= y(t-n_y), \\ x_{n_y+1}(t) &= u_1(t-\tau_1), \\ &\vdots \\ x_n(t) &= u_m(t-\tau_m-n_m). \end{aligned}$$

其中 $n = \sum_{i=1}^m (n_i + 1)$. 这样, (1) 式可写作

$$y(t) = x_1(t) \circ \cdots \circ x_n(t) \circ R. \quad (2)$$

模糊模型 (2) 的辨识问题提法如下: 给定性能指标 J , 从系统的 I/O 数据 $\{x_1(t), \cdots, x_n(t), y(t), t = 1, N\}$ 中按极小化 J 的原则确定: a) 各参考模糊集合; b) 模型的结构参数; c) 模糊关系 R .

本文只考虑确定参考集合与 R .

2. 参考模糊集合的确定

记 y, x_1, \cdots, x_n 分别为离散论域 Y, X_1, \cdots, X_n 的一般元素. 假定各论域有相同个数 ($=r$) 的参考模糊集合. 记这些参考集合为

$$\begin{cases} A_{i1}, \cdots, A_{ir} \in F(X_i), i = 1, n \\ B_1, \cdots, B_r \in F(Y). \end{cases} \quad (3)$$

其中 $F(Y)$ 表示集合 $\{Y$ 上的全体模糊集合}, 余类推. 参考集合可以解释作论域上的语义. 确定参考集合即确定其隶属函数,

$$\begin{cases} A_{i1}(x_i), \cdots, A_{ir}(x_i): X_i \rightarrow [0, 1], i = 1, n. \\ B_1(y), \cdots, B_r(y): Y \rightarrow [0, 1]. \end{cases} \quad (4)$$

为了保证模糊模型的性能并为进一步的研究提供一个统一的基础, 应保证所有参考集合都是正规的凸模糊集合并满足以下的完备性条件:

$$\forall x_i \in X_i, \exists j \in \bar{r}, A_{ij}(x_i) > 0, i = 1, n. \quad (5)$$

$$\forall y \in Y, \exists j \in \bar{r}, B_j(y) > 0. \quad (6)$$

这里 $\bar{r} \triangleq \{1, \cdots, r\}$, 即上至 r 的正整数集合.

本文中的参考模糊集合将主观地确定.

3. 辨识算法 A1——确定 R

辨识的主要内容是由系统的 I/O 数据计算模糊关系 R . 建立在参考模糊集合基础之上的 R 与直接联系论域元素的 R 在形式及意义上都有所不同. R 的隶属函数 $R(s_1, \cdots, s_n, s)$ 表达了参考集合 $A_{1s_1}, \cdots, A_{ns_n}, B_s$ 间的联系 ($s_1, \cdots, s_n, s = 1, r$).

以下给出由 I/O 数据计算 R 的算法 A1, 它是从文献 [3] 的基础上发展而来的.

算法 A1.

a) 计算第 k 组 ($k = 1, N$) 数据 $\{x_1(k), \cdots, x_n(k), y(k)\}$ 在各相应参考集合上的可能性分布^[3],

$$\begin{cases} p_{ij}(k) \triangleq \text{poss}(A_{ij} | x_i(k)) = \sup_{x_i} \min[A_{ij}(x_i), x_i(k)(x_i)], i = 1, n. \\ p_j(k) \triangleq \text{poss}(B_j | y(k)) = \sup_y \min[B_j(y), y(k)(y)]. \quad j = 1, r. \end{cases} \quad (7)$$

其中 $x_i(k)(x_i)$ 和 $y(k)(y)$ 是 $x_i(k)$ 和 $y(k)$ 的隶属函数.

b) 对 $k = 1, N$, 构造向量

$$\begin{cases} p_{x_i}(k) = [p_{i1}(k), \dots, p_{ir}(k)], & i = 1, n. \\ p_y(k) = [p_1(k), \dots, p_r(k)]. \end{cases} \quad (8)$$

于是, 由第 k 组数据构造出了模糊关系 R_k ,

$$R_k = p_{x_1}(k) \times \dots \times p_{x_n}(k) \times p_y(k), \quad (9)$$

式中“ \times ”表示笛卡尔积运算, 即

$$R_k(s_1, \dots, s_n, s) = \min[p_{1s_1}(k), \dots, p_{ns_n}(k), p_s(k)], \\ s_1, \dots, s_n, s = 1, r. \quad (10)$$

这是 max-min 合成的情形. 若采用 max-product 合成, (10) 式应改为

$$R_k(s_1, \dots, s_n, s) = p_{1s_1}(k) p_{2s_2}(k) \dots p_{ns_n}(k) p_s(k), \\ s_1, \dots, s_n, s = 1, r. \quad (11)$$

c) 计算 R

$$R = \bigcup_{k=1}^N R_k, \quad (12)$$

即

$$R(s_1, \dots, s_n, s) = \bigvee_{k=1}^N R_k(s_1, \dots, s_n, s), \quad s_1, \dots, s_n, s = 1, r. \quad (13)$$

4. 使用模糊模型

R 已知时, 利用模型 (1) 从 $x_1(k), \dots, x_n(k)$ 计算 $y(k)$ 的过程如下: 先寻找与 $x_1(k), \dots, x_n(k)$ 最“接近”的参考模糊集合, 分别记之为 $A_{1\lambda_1}, \dots, A_{n\lambda_n}$, 其中

$$\lambda_i = \{j | p_{ij}(k) > q, i \in \bar{r}\}, \quad 0 < q < 1, \quad i = 1, n. \quad (14)$$

若 $\lambda_1, \dots, \lambda_n$ 皆唯一, 可计算 $y(k)$ 如下:

$$y(k)(y) = \max_s \min [R(\lambda_1, \dots, \lambda_n, s), B_s(y)]. \quad (15)$$

若 $\lambda_1, \dots, \lambda_n$ 不唯一, 分别记其为 $\lambda_1^{(1)}, \dots, \lambda_1^{(k_1)}, \dots, \lambda_n^{(1)}, \dots, \lambda_n^{(k_n)}$, 则 (15) 式变作

$$y(k)(y) = \max_{i_1 \in \bar{k}_1} \dots \max_{i_n \in \bar{k}_n} \max_s \min [R(\lambda_1^{(i_1)}, \dots, \lambda_n^{(i_n)}, s), B_s(y)]. \quad (16)$$

注意式 (15) 或 (16) 仍对应 max-min 合成的情形. 当采用 max-product 合成时, (15) 或 (16) 式中的 min 运算应改为乘积运算.

运算中还会遇到模糊变量与非模糊变量的相互转化问题, 可参见文献 [3, 5] 等.

三、模糊模型的自学习

出于两方面的原因, 希望在模型中引入自学习功能. 其一, 用算法 A1 得到的模糊模型往往有改进的余地; 其二, 模型应具备一定的跟踪系统时变特性的能力.

自学习是在一定的性能指标意义之下进行的. 这里, 模型的预报误差被选为性能指标. 记系统的实际输出为 $y(k)$, 模糊模型的输出为 $\hat{y}(k)$, 则预报误差 $e(k) \triangleq y(k) - \hat{y}(k)$. $e(k)$ 是自学习机构的推动力, 见图 1.

一个模糊模型包含了多条规则。不同论域上的各参考集合,通过 R 的联结形成规则。

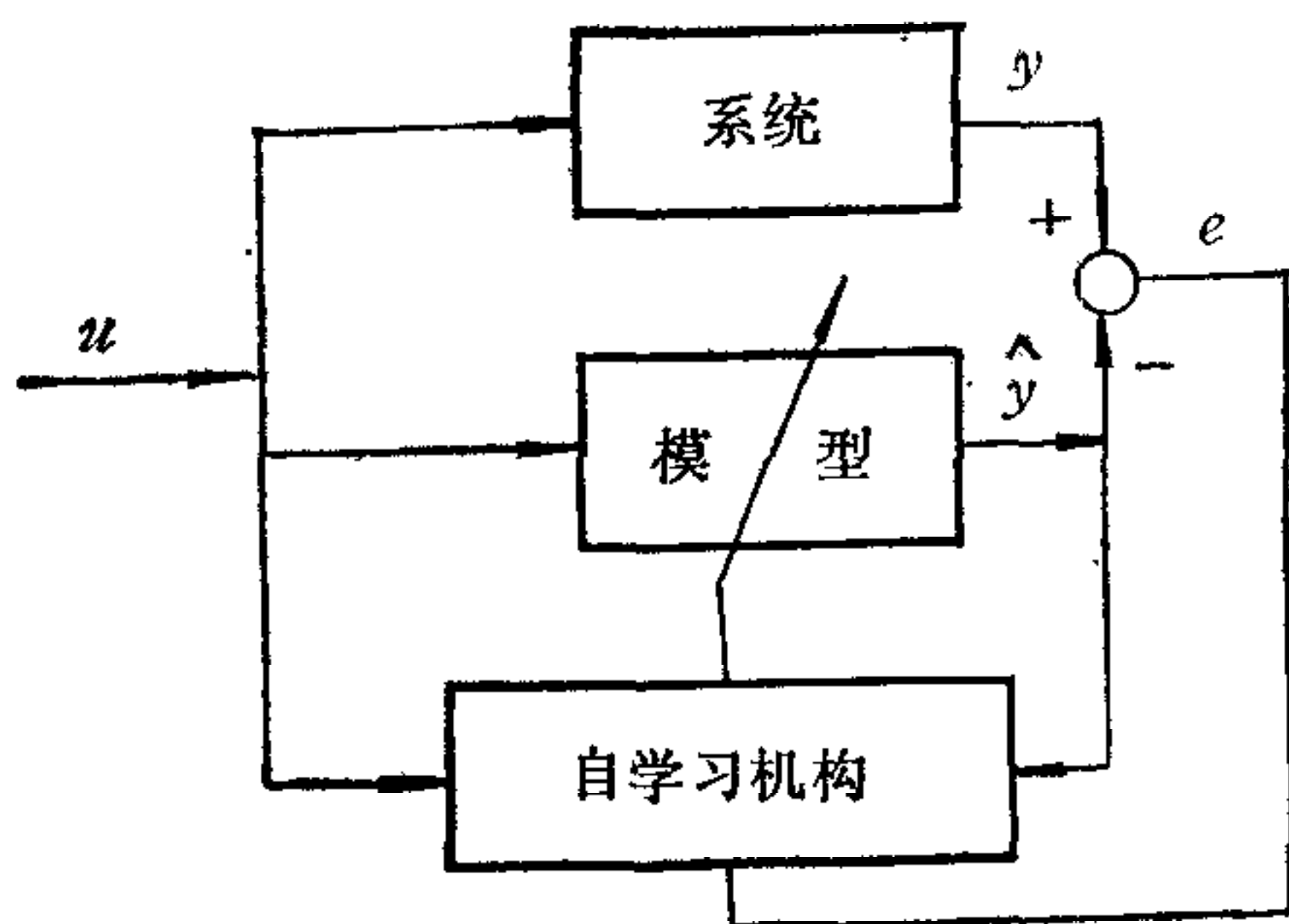


图1 模糊模型和自学习机构

R 的每一个元素对应于一条规则(元素的大小表明了该规则可信度的大小)。每次模型根据 $x_1(k), \dots, x_n(k)$ 产生 $y(k)$ 时,只启用了模型中的部分规则(例如,当 $\lambda_1, \dots, \lambda_n$ 皆为唯一时,只启用了 r 条规则)。因此,某一次模型的预报出现误差时,只需要修改参与该次预报的那些规则,不必涉及其余规则。

我们把参考模糊集合固定下来,认为修改规则即修改模糊关系 R 。

下面给出用于离线修正 R 的自学习算法 A2,它可以方便地转化为在线算法的形式。

算法 A2.

假定已有了初始模糊关系 $R^{(0)}$,现根据观测数据 $\{x_1(k), \dots, x_n(k), y(k), k = 1, N\}$ 修正 $R^{(0)}$ 。

a) $k = 0$;

b) $k + 1 \Rightarrow k$;

c) $x_1(k), \dots, x_n(k)$ 送入模糊模型 ($R^{(k-1)}$),产生预报 $\hat{y}(k)$ 。令 $e(k) = y(k) - \hat{y}(k)$ 。若 $|e(k)| < \varepsilon$ ($\varepsilon > 0$, 预先给定),则 $R^{(k-1)} \Rightarrow R^{(k)}$, 返回 b); 否则做 d);

d) 按 (7) 式算出 $p_{1j}(k), \dots, p_{nj}(k), p_j(k), j = 1, r$; 继而按 (14) 式算出 $\lambda_1, \dots, \lambda_n$ 。假定 $\lambda_1, \dots, \lambda_n$ 唯一。显然,预报 $\hat{y}(k)$ 是由与 $A_{1\lambda_1}, \dots, A_{n\lambda_n}$ 有关的 r 条规则

$$\text{if}(A_{1\lambda_1}, \dots, A_{n\lambda_n}) \text{ then } \text{poss}(B_i | y(k)) = R^{(k-1)}(\lambda_1, \dots, \lambda_n, i), i = 1, r \quad (17)$$

产生的。因此,需要修改的就是这 r 条规则,或者说,就是 R 的与这 r 条规则相对应的 r 个元素, $R^{(k-1)}(\lambda_1, \dots, \lambda_n, s), s = 1, r$ 。

e) 定义修正量 $\Delta_s^{(k)}$,

$$\Delta_s^{(k)} \triangleq p_{1\lambda_1} * \dots * p_{n\lambda_n}(k) * p_s(k), s = 1, r. \quad (18)$$

式中,对于 max-min 合成, $*$ = \wedge ; 对于 max-product 合成, $*$ = \times 。值得提醒的是,在 (18) 式中,

$$p_s(k) = \text{poss}(B_s | y(k)), s = 1, r. \quad (19)$$

f) 修正 $R^{(k-1)}$, 得到 $R^{(k)}$,

$$R^{(k)}(s_1, \dots, s_n, s) = \begin{cases} \alpha_s \Delta_s^{(k)} + (1 - \alpha_s) R^{(k-1)}(s_1, \dots, s_n, s), & \text{当 } s_1 = \lambda_1, \dots, s_n = \lambda_n. \\ R^{(k-1)}(s_1, \dots, s_n, s), & \text{其它.} \end{cases} \quad (20)$$

$$\alpha_s \in [0, 1], s = 1, r.$$

g) 若 $k = N$ 则结束,否则返 b)。

考察一下 $\Delta_s^{(k)}(s = 1, r)$ 可见,如果把产生了 $\hat{y}(k)$ 的规则 (17) 改为

$$\text{if}(A_{1\lambda_1}, \dots, A_{n\lambda_n}) \text{ then } \text{poss}(B_i | y(k)) = \Delta_i^{(k)}, i = 1, r. \quad (21)$$

则由于 $\Delta_1^{(k)}, \dots, \Delta_r^{(k)}$ 是根据实测值 $y(k)$ 算得的, (21) 式将力图产生如下的推断:

$$\text{if}(x_1(k), \dots, x_n(k)) \text{ then } y(k). \quad (22)$$

修正作用正是借此实现的. 在算法 A2 中, 若对所有 s 有 $\alpha_s = 0$, 则修正作用消失; 若对所有 s 有 $\alpha_s = 1$, 则意味着在 $R^{(k)}$ 中将以规则 (21) 完全代替 $R^{(k-1)}$ 中的规则 (17). 考虑到 $y(k)$ 中总是包含噪声, 若取 $\alpha_s = 1$ 则噪声的影响将立即侵入模型, 因此对所有 s , 总是使 $\alpha_s < 1$. 此时, 自学习的修正作用取决于多步的累积效果, 相当于每一次修正都在产生了 $y(k)$ 的旧规则 (17) 和力图产生 $y(k)$ 的新规则 (21) 之间采取折中. 可见, α_s 决定了规则更新的“步幅”. α_s 大时自学习见效快, 但噪声的影响也容易侵入模型; α_s 小虽能减弱噪声的影响, 但自学习的速度变慢.

$\alpha_s (s = 1, r)$ 由两个因素决定. 第一个因素是 $|e(k)|$ 的大小. $|e(k)|$ 大则 α_s 亦应大. $e(k) = 0$ 时应有 $\alpha_s = 0$. 第二个因素是产生了 $y(k)$ 的 r 条规则对 $y(k)$ 的贡献的相对大小. 显然, 贡献大的规则受到的修改亦应大 (α_s 大). 为了考察 (17) 式中各条规则对 $y(k)$ 的贡献, 我们写出

$$y(k)(y) = \max_{s} \min [R^{(k-1)}(\lambda_1, \dots, \lambda_n, s), B_s(y)]. \quad (23)$$

若记

$$a_s = R^{(k-1)}(\lambda_1, \dots, \lambda_n, s), \quad s = 1, r \quad (24)$$

并定义 $C_1, \dots, C_r \in F(Y)$,

$$C_s(y) = a_s, \quad \forall y, \quad s = 1, r. \quad (25)$$

再定义

$$\hat{Y}_s(k) = B_s \cap C_s, \quad s = 1, r. \quad (26)$$

于是, (15) 式可以写成

$$y(k) = \bigcup_{s=1}^r \hat{Y}_s(k). \quad (27)$$

可以看出, 上式中的 $\hat{Y}_s(k)$ 正是 (17) 式中的一条旧规则

$$\text{if}(A_{1\lambda_1}, \dots, A_{n\lambda_n}) \text{ then } \text{poss}(B_s | y(k)) = R^{(k-1)}(\lambda_1, \dots, \lambda_n, s)$$

对 $y(k)$ 的贡献 ($s = 1, r$). 我们定义描述 $\hat{Y}_1(k), \dots, \hat{Y}_r(k)$ 对 $y(k)$ 贡献相对大小的量 β_1, \dots, β_r ,

$$\beta_s \triangleq \sum_y (\hat{Y}_s(k)(y))^2 \geq 0, \quad s = 1, r. \quad (28)$$

显然, β_s 愈大, $\hat{Y}_s(k)$ 对 $y(k)$ 的贡献也愈大.

于是, α_s 被定义为

$$\alpha_s \triangleq h \cdot \beta_s \cdot |e(k)|, \quad s = 1, r. \quad (29)$$

其中 h 为常数, 用于调整 α_s 的变化范围.

需要指出, 即使 $\lambda_1, \dots, \lambda_n$ 不唯一, 自学习算法 A2 以及关于 α_s 的讨论也都是适用的. 本文的计算示例中就允许 $\lambda_1, \dots, \lambda_n$ 不唯一.

四、计算示例

例 1. 利用 Box 的燃烧炉 I/O 数据. 这些数据在多处文献中引用过^[1,3,6,7]. 假定

模糊模型具有结构“if($y(k - \tau_1)$ and $u(k - \tau_2)$) then $y(k)$ ”, 性能指标 J 被定义为模型输出与实测值间的均方误差. $\tau_1 = 1$, $\tau_2 = 4$ 使 J 达到极小: $J = 0.4555$. 而经过自学习之后 $J = 0.328$. 与文献 [1,3,6,7] 等比较可以看到本文的 J 较小.

例 2. 略¹⁾.

五、结 语

用模糊模型描述复杂系统的动态行为,既有必要,也有可能. 本文给出的辨识算法可以获得具有一定精度的模糊模型;自学习算法则可以进一步改善这一精度,并能为模型的在线应用提供跟踪系统时变特性的能力. 不过,所做的初步努力无疑尚有进一步深化的必要.

参 考 文 献

- [1] Tong, R. M., Synthesis of Fuzzy Models for Industrial Processes, *Int. J. Gen. Systems* 4(1978), 143—62.
- [2] Czogala, E. and Pedrycz, W., On Identification in Fuzzy Systems and Its Applications in Control Problems. *Fuzzy Sets and Systems* 10(1981), 73—83.
- [3] Pedrycz, W., An Identification Algorithm in Fuzzy Relational Systems. *Fuzzy Sets and Systems* 13(1984), 153—67.
- [4] Higashi, M. and Klir, G. J., Identification of Fuzzy Relation Systems. *IEEE Trans. Syst. Man & Cybern.*, SMC-14(1984), 349—55.
- [5] Dubois, D. and Prade, H., *Fuzzy Sets And Systems: Theory And Applications*. New York, Academic, 1980.
- [6] 李宝绶、刘志俊, 用模糊理论测辨系统模型, 信息与控制, 9(1980), 第三期.
- [7] Box, G. E. P. and Jenkins, G. M., *Time Series Analysis, Forecasting and Control*, San Francisco, Holden Day, 1970.

IDENTIFICATION AND SELF-LEARNING OF FUZZY MODELS FOR DYNAMIC SYSTEMS

XU CHENGWEI

(Kunming Institute of Technology)

LU YONGZAI

(Zhejiang University)

ABSTRACT

This paper presents an identification algorithm and the associated self-learning algorithm for fuzzy models of MISO dynamic systems, together with numerical examples. The identification algorithm can produce fuzzy models with fairly high accuracy, and the self-learning algorithm may further improve the model's accuracy.

1) 徐承伟, 模糊系统的辨识, 控制及优化, 浙江大学硕士论文, 1986 年.