

把离散系统模型转换成连续系统模型的递推算法及其稳定性

连绥仁 谢两叁

(石油工业部) (休斯敦大学电机系)

摘 要

在 Newton-Raphson 法的基础上,本文提出收敛较快、数值稳定的递推算法把离散系统模型转换成等价的连续系统模型,分析了算法的稳定性并举例说明所提出的算法的效果。

关键词——控制系统,状态空间法,模型转换。

一、引 言

一部复杂的控制系统往往由若干个离散系统和连续系统组成。设计这种系统的一个行之有效的方法是把离散系统转换成等价的连续系统模型,再用分析与设计连续控制系统的种种技巧来设计整个控制系统^[1]。在识别一部连续系统时,常利用输入和输出的采样数据建立离散系统模型,然后将它转换成连续系统模型^[1-4]。关于把离散系统模型转换成连续系统模型的研究结果见文献[1-6]。

如果离散系统模型是

$$\mathbf{x}[(k+1)T] = G\mathbf{x}(kT) + H\mathbf{u}(kT). \quad (1a)$$

$$\mathbf{y}(kT) = C\mathbf{x}(kT). \quad (1b)$$

等价的连续系统模型是

$$\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}}(t) + B\hat{\mathbf{u}}(t). \quad (2a)$$

$$\hat{\mathbf{y}}(t) = C\hat{\mathbf{x}}(t). \quad (2b)$$

其中 $\mathbf{x}, \hat{\mathbf{x}} \in R^n$, $\mathbf{u}, \hat{\mathbf{u}} \in R^m$, $\mathbf{y}, \hat{\mathbf{y}} \in R^r$, G, H, C, A 和 B 是常数矩阵并且 $G, A \in R^{n \times n}$, $H, B \in R^{n \times m}$, $C \in R^{r \times n}$, T 是采样周期。若 $A = \frac{1}{T} \ln G$, $B = A(G - I_n)^{-1}H$, 并且 $\hat{\mathbf{u}}(t) = \mathbf{u}(kT)(kT \leq t < (k+1)T)$, 那么 $\hat{\mathbf{x}}(kT) = \mathbf{x}(kT)$ 。Sinha 等^[1,2,3]把 Newton-Raphson 法用于矩阵方程 $\exp(-AT) - G^{-1} = O_n$, 得到计算矩阵 A 的递推公式如下

$$A(k+1) = A(k) + \frac{1}{T} [I_n - \exp(A(k)T)G^{-1}], \quad A(0) = \frac{2}{T} [G - I_n][G + I_n]^{-1}, \quad (3a)$$

$$\lim_{k \rightarrow \infty} A(k) = A. \quad (3b)$$

算法(3)的精度取决于 $\exp(X)$ 的计算精度. 计算 $\exp(X)$ 的方法见文献[7], 最常用的是直接截断法

$$\exp(X) \cong \sum_{i=1}^p \frac{X^{i-1}}{(i-1)!}, \quad p = \text{正整数}. \quad (4)$$

文献[3]指出: 算法(3)在 $\sigma(AT) \leq 0.5$ 时收敛, 否则可能会不稳定; 对于给定的矩阵 G , 如果算法(3)不稳定则表明采样周期太大; 如果初值 $A(0)$ 离开答案很远, 算法(3)收敛很慢. 此外, 算法(3)需要求矩阵 G 的逆. 对于某些控制对象, 条件

$$\sigma(AT) \leq 0.5$$

并不一定能满足. 所以研究收敛较快并且数值稳定的递推算法以实现模型转换是必要的. 本文先讨论如何提高收敛速度, 在此基础上给出数值稳定的算法.

二、收敛速度的改进

为了得到收敛快的算法, 构造矩阵方程 $\ln[\exp(-AT)G] = O_n$, 用 Newton-Raphson 法可得

$$A(k+1) = A(k) + \frac{1}{T} \ln[\exp(-A(k)T)G]. \quad (5)$$

式(5)又可以写成

$$A(k+1) = A(k) - \frac{1}{T} \ln[\exp(A(k)T)G^{-1}]. \quad (6)$$

若取 $A(0) = O_n$, 则 $A(1) = \frac{1}{T} \ln G$ 就是所要的答案. 但算法(5)和(6)式并不能实际应用, 因为其中包含矩阵对数函数. 利用近似公式

$$\ln(X) \cong \sum_{i=1}^p (-1)^{i-1} \frac{(X - I_n)^i}{i}, \quad p = \text{正整数}, \quad (7)$$

那么算法(5)和(6)成为

$$A(k+1) = A(k) - \frac{1}{T} \sum_{i=1}^p \frac{(I_n - \exp(-A(k)T)G)^i}{i}, \quad A(0) = O_n, \quad (8)$$

$$A(k+1) = A(k) + \frac{1}{T} \sum_{i=1}^p \frac{(I_n - \exp(A(k)T)G^{-1})^i}{i}, \quad A(0) = O_n. \quad (9)$$

在(9)式中取 $p=1$ 便得到算法(3). 注意, 算法(8)式中没有矩阵求逆运算. 算法(8)和(9)式的收敛速度决定于 p . 由于在实际计算时存在舍入误差, 对于某些矩阵 G , 算法(3)、(8)和(9)式会不稳定. 算法(3)、(8)和(9)式数值稳定的充分条件是

$$|K_{ij}| \triangleq \begin{cases} 1 + \frac{1 - \frac{\lambda_i}{\lambda_j}}{\ln \frac{\lambda_i}{\lambda_j}} & i, j = 1, 2, \dots, n. \\ \frac{\lambda_i}{\lambda_j} & i \neq j \end{cases} \leq 1, \quad (10)$$

式中 λ_i 是 G 的特征值。如果条件 (10) 式不满足, 算法 (3)、(8) 和 (9) 式的最大发散速度(定义为相邻两次递推结果差值的范数之比)是

$$R = |K_{ij}|_{\max} \triangleq \text{Max}[|K_{ij}|, i, j = 1, 2, \dots, n]. \quad (11)$$

条件 (10) 及式 (11) 的推导如下。

考虑算法 (3)。假定存在矩阵 M 使得

$$M^{-1}GM = \text{diag}\{\lambda_i, i = 1, 2, \dots, n\} \triangleq \Lambda, \quad (12)$$

并假定第 k_0 次递推时存在舍入误差 $E(k_0)$, 经舍入后的结果是 $\tilde{A}(k_0)$, 第 k 次 ($k > k_0$) 递推结果是 $\tilde{A}(k)$ 、误差为 $E(k)$, 那么

$$\tilde{A}(k_0) = A(k_0) + E(k_0), \quad (13)$$

$$\tilde{A}(k) = A(k) + E(k), \quad k > k_0. \quad (14)$$

再假定由 $\tilde{A}(k)$ 计算 $\tilde{A}(k+1)$ 时没有舍入误差, 故

$$\tilde{A}(k+1) = \tilde{A}(k) + \frac{1}{T} [I_n - \exp(\tilde{A}(k)T)G^{-1}], \quad k > k_0. \quad (15)$$

将 (14) 式代入 (15) 式, 然后减去 (3a), 把矩阵指数函数展开成级数并略去 $E(k)$ 的高阶小量, 再用矩阵 M 进行变换, 最后得到下述 n^2 个数量差分方程式

$$\varepsilon_{ij}(k+1) = K_{ij}\varepsilon_{ij}(k), \quad \varepsilon_{ij}(k_0), \quad i, j = 1, 2, \dots, n, \quad (16a)$$

$$K_{ij} = \begin{cases} 0, & i = j, \\ 1 + \frac{1 - \frac{\lambda_i}{\lambda_j}}{\ln \frac{\lambda_i}{\lambda_j}}, & i \neq j. \end{cases} \quad (16b)$$

式中 $\varepsilon_{ij}(k)$ 是 $\varepsilon(k) \triangleq M^{-1}E(k)M$ 的元素, $\varepsilon_{ij}(k_0)$ 是 $\varepsilon(k_0) \triangleq M^{-1}E(k_0)M$ 的元素。如果式 (16) 中所有差分方程式稳定, 那么算法 (3) 数值稳定。由于

$$E(k) = M\varepsilon(k)M^{-1} = \left\{ \sum_{i=1}^n \sum_{j=1}^n \alpha_{ij}^{p,q} K_{ij}^{k-k_0} \varepsilon_{ij}(k_0), \quad p, q = 1, 2, \dots, n \right\}$$

把 $|K_{ij}|$ 的最大值记作 $|K_{ij}|_{\max}$, 次大值记作 $|K_{ij}|_{\text{submax}}$, 对应的系数矩阵记作

$$\{\alpha_{ij}^{p,q}, p, q = 1, 2, \dots, n\}_{\max}$$

和 $\{\alpha_{ij}^{p,q}, p, q = 1, 2, \dots, n\}_{\text{submax}}$ 。若 $|K_{ij}|_{\max} > 1$ 且 $\{\alpha_{ij}^{p,q}, p, q = 1, 2, \dots, n\}_{\max} \neq O_n$, 则当 $k \rightarrow \infty$ 时 $\|E(k)\| \cong \|\{\alpha_{ij}^{p,q}, p, q = 1, 2, \dots, n\}_{\max}\| \cdot |K_{ij}|_{\max}^{k-k_0}$ 。由发散速度定义

$$R = \|E(k+1)\| / \|E(k)\|$$

得到式 (11)。如果 $\{\alpha_{ij}^{p,q}, p, q = 1, 2, \dots, n\}_{\max} = O_n$, $|K_{ij}|_{\text{submax}} \leq 1$, 即使 $|K_{ij}|_{\max} > 1$, 算法 (3) 数值稳定。若 $\{\alpha_{ij}^{p,q}, p, q = 1, 2, \dots, n\}_{\max} = O_n$, $\{\alpha_{ij}^{p,q}, p, q = 1, 2, \dots, n\}_{\text{submax}} \neq O_n$, $|K_{ij}|_{\text{submax}} > 1$, 则发散速度 $R = |K_{ij}|_{\text{submax}}$ 。对于算法 (8) 和 (9) 式, 用同样的方法可以得到条件 (10) 和 (11) 式。

三、数值稳定的递推算法

为了得到数值稳定的递推算法, 令

$$N(k) \triangleq \exp(-A(k)T)G. \quad (17)$$

把它代入式(5)得到

$$-A(k+1)T = -A(k)T - \ln(N(k)), \quad A(0) = O_n. \quad (18)$$

对式(18)两边求指数函数,再利用式(17)和(7)可得

$$A(k+1) = A(k) - \frac{1}{T} \sum_{i=1}^p \frac{[I_n - N(k)]^i}{i}, \quad A(0) = O_n, \quad (19a)$$

$$N(k+1) = N(k) \exp \left[\sum_{i=1}^p \frac{[I_n - N(k)]^i}{i} \right], \quad N(0) = G. \quad (19b)$$

$$\lim_{k \rightarrow \infty} A(k) = A, \quad \lim_{k \rightarrow \infty} N(k) = I_n. \quad (19c)$$

算法(19)式在下述意义下是稳定的: 第 k 次递推时的舍入误差不会引起以后递推误差无限增加. 证明如下:

考虑算法(19)式中 $p = 1$ 的情形. 若第 k 次递推时 $N(k)$ 和 $A(k)T$ 的舍入误差分别是 $E(k)$ 和 $F(k)$, 经过舍入后的递推结果为 $\tilde{N}(k)$ 和 $\tilde{A}(k)$, 即

$$\tilde{N}(k) = N(k) + E(k). \quad (20a)$$

$$\tilde{A}(k)T = A(k)T + F(k). \quad (20b)$$

假定由 $\tilde{N}(k)$ 和 $\tilde{A}(k)$ 用下式计算 $\tilde{N}(k+1)$ 和 $\tilde{A}(k+1)$ 时没有舍入误差

$$\tilde{N}(k+1) = \tilde{N}(k) \exp[I_n - \tilde{N}(k)]. \quad (21a)$$

$$\tilde{A}(k+1)T = \tilde{A}(k)T - [I_n - \tilde{N}(k)]. \quad (21b)$$

把式(20)代入式(21)然后减去式(19)(取 $p = 1$), 将矩阵指数函数展开成级数并略去 $E(k)$ 和 $F(k)$ 的高阶小量, 考虑到 $k \rightarrow \infty$ 时 $\exp(-A(k)T) \rightarrow G^{-1}$, 故 $N(k) \rightarrow I_n$, $\exp(I_n - N(k)) \rightarrow I_n$, 最后得到

$$E(k+1) = O_n.$$

$$F(k+1) = F(k) + E(k). \quad (22b)$$

进一步假定在第 $(k+2)$ 次递推时没有舍入误差, 那么

$$F(k+2) = F(k+1) + E(k+1) = F(k+1).$$

故算法(19)在 $p = 1$ 时数值稳定. 用同样的方法可以证明算法(19)在 $p > 1$ 时数值稳定.

四、例 题

考虑一个两轴透平机的连续系统模型^[8]

$$A = \begin{bmatrix} -1.268 & -0.04582 & -1.498 & 951.5 \\ 1.002 & -1.957 & 8.52 & 1240 \\ 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & -100 \end{bmatrix}. \quad (23)$$

其特征值 $\sigma(A) = \{-1.3477, -1.8833, -10, -100\}$. 在 $T = 0.04$ 秒时对应的离散系统模型为

$$G = \begin{bmatrix} 0.95051060 & -0.00169810 & 0.04781321 & 8.9678048 \\ 0.03757718 & 0.92467160 & 0.27047831 & 11.73618076 \\ 0 & 0 & 0.67032005 & 0 \\ 0 & 0 & 0 & 0.01831564 \end{bmatrix}. \quad (24)$$

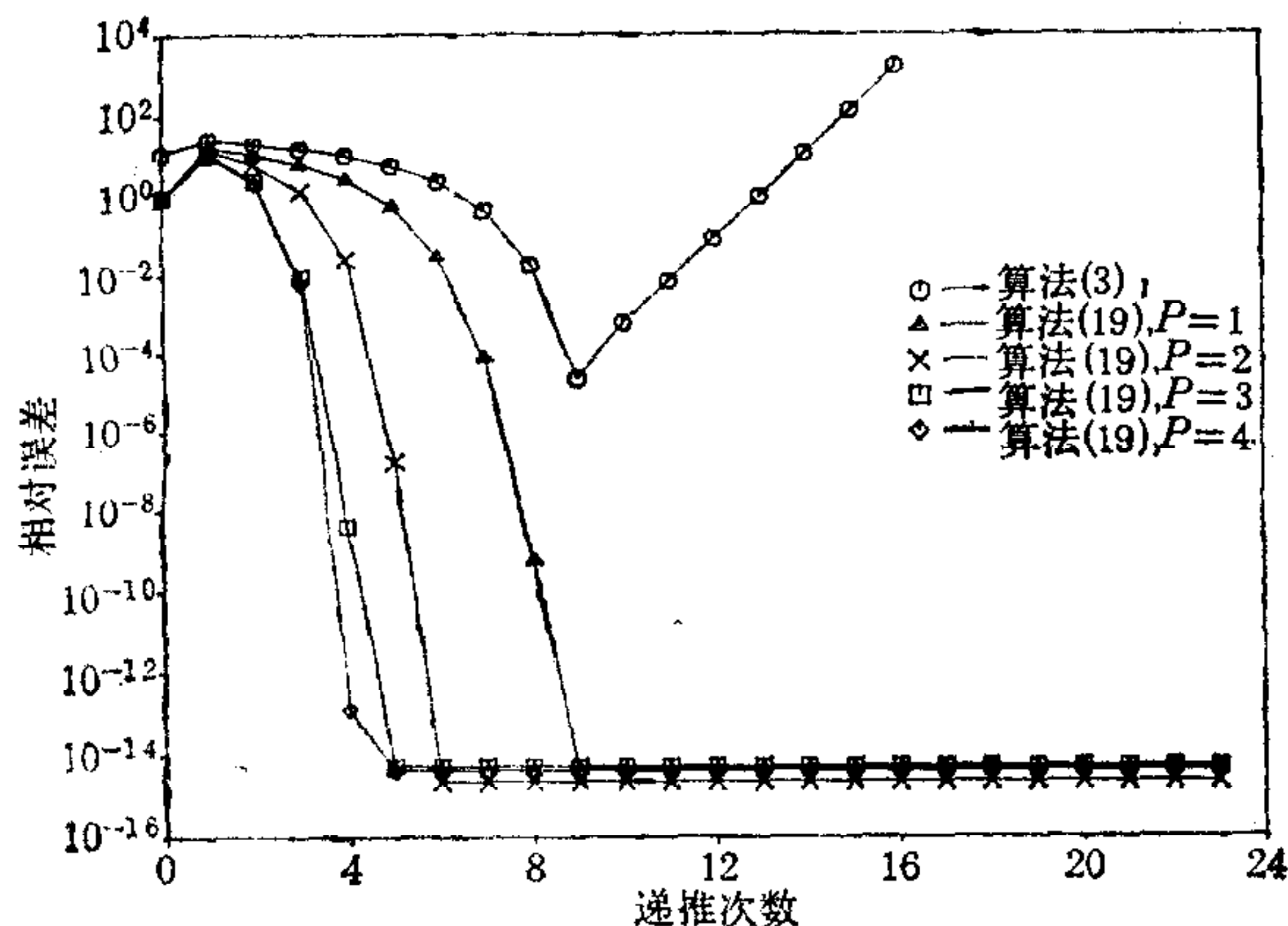


图1 由 G 计算 A 的计算机模拟的结果

然后以 $R = 11.897$ 发散。发散速度与按式 (11) 的计算结果 $R = 11.888$ 吻合。算法 (19) 式数值稳定, 在 $p = 1, 2, 3, 4$ 时它分别在 $k = 9, 6, 5, 4$ 时收敛到误差

$$\|(A(k) - A)A^{-1}\| = 4.4 \times 10^{-15}, 2.2 \times 10^{-15}, 5 \times 10^{-15}$$

和 1.3×10^{-13} 。

其特征值

$$\sigma(G) = \{0.94751927, 0.92743553, 0.607032005, 0.01831564\}.$$

用算法 (3) 和 (19) 由矩阵 G 计算矩阵 A 。在算法 (19) 式中取

$$p = 1, 2, 3, 4.$$

用直接截断法 (4) 计算矩阵指数函数, 项数取 31 项。在 VAX11/750 计算机上模拟的结果如图 1 所示。对于式 (24) 中的矩阵 G , 条件 (10) 不满足, 算法 (3) 数值不稳定, 在 $k = 9$ 时相对误差减少到

$$\|(A(9) - A)A^{-1}\| = 2.4 \times 10^{-5},$$

五、结 论

本文提出了收敛速度较快并且数值稳定的递推算法用于把离散系统模型转换成连续系统模型并举例说明所提出的算法的效果。本文还分析了已有的算法 (3) 产生数值不稳定的原因并导出了稳定条件。

参 考 文 献

- [1] Sinha, N. K. and Kuszta, B. Modeling and Identification of Dynamic Systems, New York, Van Nostrand Reinhold, 1983, 98—102.
- [2] Sinha N. K., and Lastman, G. J., Transformation Algorithm for Identification of Continuous-Time Multivariable Systems from Discrete Data, *Electronic Letter*, 17(1981), 779—780.
- [3] Sinha N. K., and Lastman, Identification of Continuous-Time Multivariable Systems from Sampled Data, *Int. J. Control*, 35(1982), 117—126.
- [4] Puthenpura and Sinha, N. K., Transformation of Continuous-Time Model of a Linear Multivariable System from Its Discrete-Time Model, *Electronic Letter*, 20(1984), 737—738.
- [5] Shieh, L. S., Tsai J. S. H., and Lian, S. R., Determining Continuous-Time State Equations from Discrete-Time State Equations via the Principal q th Root Method, *IEEE Trans. Automatic Control*, AC-31 (1986), 454—457.
- [6] Shieh, L. S., Wang H., and Yates, R. E., Discrete-Continuous Model Conversion, *Appl. Math. Modelling*,

£ (1980), 449—455.

- [7] C. Moler and C. Van Loan, Nineteen Dubious Ways to Compute the Exponential of a Matrix, *SIAM Review*, 20(1978), 801—836.
- [8] P. D. McMorran, Design of Gas-Turbine Controller Using Inverse Nyquist Method, *IEE Proceeding*, 117 (1970), 2050—2056.

THE RECURSIVE ALGORITHMS FOR MODELING CONTINUOUS-TIME SYSTEMS FROM DISCRETE-TIME SYSTEMS AND THEIR STABILITY

LIAN SUIREN

(Ministry of Petroleum Industry)

SHIEH S. LEANG

(University of Houston)

ABSTRACT

Based on the Newton-Raphson method, this paper presents the recursive algorithms that are rapidly convergent and numerically stable for modeling the equivalent continuous-time model from the available discrete-time model. A practical example is presented to demonstrate the effectiveness of the proposed procedures.

Key words —— Control systems; state-space approach; model conversions.