

# 系统辨识(7): 递阶辨识原理与方法

丁锋<sup>1,2,3</sup>

## 摘要

递阶辨识是系统辨识的一个重要分支. 递阶辨识原理是在大系统递阶控制的“分解-协调原理”基础上发展起来的, 它不仅能够解决参数数目多、维数高、大规模系统辨识算法计算量大的问题, 而且能够解决结构复杂的双线性参数系统、多线性参数系统以及非线性系统的辨识问题. 首先介绍递阶辨识原理和线性方程组  $Ax = b$  的著名雅可比迭代和高斯-赛德尔迭代, 给出了线性方程组的迭代方法族; 其次将雅可比迭代思想和递阶辨识原理用于研究一般矩阵方程和耦合矩阵方程的递阶梯度迭代求解方法和递阶最小二乘迭代求解方法; 再次介绍了方程误差模型的两阶段最小二乘辨识方法(一个简单的递阶辨识方法)和线性回归模型的递阶最小二乘辨识方法; 最后研究了类多变量 CARMA 系统的递阶辨识方法.

## 关键词

迭代辨识; 递推辨识; 参数估计; FIR 模型; 方程误差模型; CAR 模型; CARMA 模型; CARAR 模型; CARARMA 模型; 输出误差模型; OEMA 模型; OEAR 模型; 辅助模型辨识; 多新息辨识; 递阶辨识; 耦合辨识

中图分类号 TP273

文献标志码 A

收稿日期 2012-04-16

资助项目 国家自然科学基金(60973043)

作者简介

丁锋, 男, 博士, 教授, 博士生导师, 主要从事系统辨识、过程建模、自适应控制方面的研究. [fding@jiangnan.edu.cn](mailto:fding@jiangnan.edu.cn)

## 0 引言

随着自动化电子科学技术的进步以及计算设备能力的提高, 人们处理大规模问题的能力也越来越强. 随着问题规模的扩大, 随之而来的是方法的计算量也越大. 同样, 系统辨识面临的辨识对象规模也是越来越庞大, 变量越来越多, 维数越来越高, 导致辨识算法的计算量也越来越大. 在这种背景下, 提出新思想、新理论、新原理、新概念, 研究出新辨识方法, 减小辨识算法的计算量已成为必然.

递阶辨识是基于辨识模型分解而发展起来的一种新型辨识方法, 其基础是递阶辨识原理, 是为解决结构复杂、维数高、大规模系统辨识问题而提出的. 其基本思想是通过对辨识模型的分解, 使得子辨识问题的规模变小, 相对于原问题要简单, 从而减小辨识方法的计算量. 新思想、新理论、新原理、新概念的诞生都是科学史上的重要里程碑. 就研究建立系统数学模型的理论和方法的系统辨识而言, 辅助模型辨识思想、多新息辨识理论、递阶辨识原理、耦合辨识概念的诞生, 有助于推动系统辨识学科的研究进程<sup>[1-7]</sup>.

本文介绍一类新型的辨识方法: 递阶辨识方法. 它是基于本文作者提出的递阶辨识原理而建立的一些辨识方法. 递阶辨识的重要研究成果都发表在《Automatica》和《IEEE Transactions on Automatic Control》等国际著名期刊.

递阶辨识(hierarchical identification), 即分解辨识的概念是本文作者博士毕业不久, 1996年给清华大学硕士、博士研究生讲授《大系统理论及应用》时, 受大系统递阶控制的分解-协调原理(decomposition-coordination principle)的启发提出的. 其第1篇递阶辨识论文“大系统的递阶辨识”正式发表在《自动化学报》1999年第1期上<sup>[8]</sup>. 递阶辨识不仅可用于解决维数大、变量数多的大系统辨识的计算量大的问题, 而且对所谓的小系统以及具有复杂结构耦合多变量系统的辨识也非常有用. 如在多变量系统传递函数阵主模型(main model)的递阶辨识方面, 显示出其独到的特点. 在递阶辨识领域, 本文作者等首次提出了递阶辨识原理的概念, 提出了传递函数主模型的递阶梯度迭代辨识方法和递阶随机梯度辨识方法<sup>[9]</sup>、递阶最小二乘迭代辨识方法和递阶最小二乘辨识方法<sup>[10-11]</sup>, 提出了状态空间模型的递阶辨识方法<sup>[12]</sup>、双率系统提升状态空间模型的递阶辨识方法<sup>[13]</sup>等. 其在递阶辨识领域的主要学术贡献如下.

1 江南大学 物联网工程学院, 无锡 214122

2 江南大学 控制科学与工程研究中心, 无锡, 214122

3 江南大学 教育部轻工过程先进控制重点实验室, 无锡 214122

1) 发展了“基于子系统分解的大系统递阶控制(hierarchical control)的分解-协调原理”,进而提出了“基于辨识模型分解的递阶辨识原理及其交互估计理论”,并将其应用于变量多、结构复杂、强耦合大规模多变量系统的辨识建模,首次提出了这类复杂耦合大规模多变量系统的递阶辨识方法,有效地解决一类复杂多变量系统辨识方法计算量大的问题,为复杂系统的辨识建模和参数估计提供了新方法<sup>[8-13]</sup>.提出的递阶辨识原理,是根据系统参数化后辨识模型的结构特征,对辨识模型表达式进行分解,不同于传统多变量系统的子系统分解辨识方法,也不同于再建立一个时变参数模型的传统递阶分层辨识.进一步结合辅助模型辨识思想,采用递阶辨识原理及其交互估计理论,提出了有色噪声干扰 Hammerstein 非线性 ARMAX 系统的递阶辨识方法,解决了一类有色噪声干扰非线性系统的辨识难题<sup>[14-19]</sup>.

2) 首次将递阶辨识原理用于研究大型李雅普诺夫(Lyapunov)矩阵方程、西尔维斯特(Sylvester)矩阵方程、一般矩阵方程、耦合 Sylvester 矩阵方程、一般耦合矩阵方程的递阶迭代求解问题.把这类耦合矩阵方程中的未知待求解矩阵看作一个待辨识系统参数矩阵,提出了计算量小、收敛速度快的递阶最小二乘迭代方法和递阶梯度迭代方法,为系统与控制中大型耦合矩阵方程的递阶迭代解提出了新方法<sup>[20-27]</sup>.

3) 利用李雅普诺夫稳定性理论、随机过程理论和鞅收敛定理,研究了一些矩阵方程求解迭代算法的收敛性和多变量系统递阶辨识方法的收敛性能,丰富和发展了辨识建模理论.

基于递阶辨识原理的递阶辨识方法已经成为一个崭新的辨识领域,可以用于研究各种结构复杂的多变量系统和非线性系统的辨识问题.例如:

- 1) 大系统的递阶辨识<sup>[8]</sup>;
- 2) 多变量离散时间系统传递函数阵参数的递阶梯度迭代辨识方法与递阶随机梯度辨识方法<sup>[9]</sup>;
- 3) 多变量系统传递函数模型的递阶最小二乘迭代辨识方法与递阶最小二乘辨识方法<sup>[10]</sup>;
- 4) 多输入多输出 ARX-like 系统递阶最小二乘辨识算法的一致性<sup>[11]</sup>;
- 5) 基于递阶辨识原理的一类矩阵方程的递阶梯度迭代算法<sup>[21]</sup>;
- 6) 基于递阶辨识原理的耦合西尔维斯特矩阵方程与一般耦合矩阵方程的最小二乘迭代解<sup>[23]</sup>;

7) 基于递阶辨识原理的一般双率系统提升状态空间模型的递阶辨识<sup>[13]</sup>;

8) 双率系统递阶最小二乘参数估计及其收敛性<sup>[28]</sup>;

9) 使用递阶辨识原理的广义西尔维斯特矩阵方程的迭代解<sup>[22]</sup>;

10) 基于递阶辨识原理的一般耦合矩阵方程的递阶梯度迭代解<sup>[20]</sup>;

11) 多变量系统状态空间模型的递阶辨识<sup>[12]</sup>;

12) 一般双率系统状态空间模型及其递阶辨识<sup>[29]</sup>;

13) 非均匀周期采样多率系统的一种递阶辨识方法<sup>[30]</sup>;

14) 衰减激励条件下递阶最小二乘辨识的均方收敛性<sup>[31]</sup>;

15) 鞅超收敛定理与传递函数阵递阶随机梯度辨识方法的收敛性分析<sup>[32]</sup>;

16) 递阶辨识方法在转台伺服系统调试中的应用研究<sup>[33]</sup>.

本文应用递阶辨识原理,讨论一些递阶辨识方法:如矩阵方程的递阶梯度迭代算法、耦合矩阵方程的递阶最小二乘迭代算法、线性回归模型的递阶辨识方法以及多变量系统的递阶迭代辨识算法等.本文较长,为便于阅读,特将本文框架结构列示如下.

## 0 引言

### 1 递阶辨识原理

### 2 迭代方法族

#### 2.1 雅可比和高斯-赛德尔迭代

#### 2.2 矩阵方程 $Ax = b$ 迭代方法族

#### 2.3 矩阵方程 $AXB = F$ 的迭代解

### 3 一般矩阵方程

#### 3.1 西尔维斯特矩阵方程 $AX + XB = F$

#### 3.2 矩阵方程 $AXB + X = F$

#### 3.3 矩阵方程 $AXB + CXD = F$

#### 3.4 矩阵方程 $A_1XB_1 + \dots + A_pXB_p = F$

### 4 耦合矩阵方程

#### 4.1 耦合西尔维斯特矩阵方程

#### 4.2 一般耦合矩阵方程

#### 4.3 其他矩阵方程

### 5 方程误差模型的两阶段递推最小二乘算法

#### 5.1 系统描述与问题构成

#### 5.2 两阶段递推辨识算法

#### 5.3 仿真例子

- 6 线性回归模型的递阶最小二乘辨识方法
  - 6.1 辨识模型与问题构成
  - 6.2 递阶最小二乘算法的推导
  - 6.3 递阶最小二乘算法的收敛性
- 7 类多变量 CARMA 系统的递阶辨识方法
  - 7.1 递阶梯度迭代辨识算法
  - 7.2 递阶最小二乘迭代算法
  - 7.3 交互噪声干扰的情形
- 8 结语

## 1 递阶辨识原理

递阶辨识原理(hierarchical identification principle)分为3步. 1) 辨识模型分解. 将一个辨识模型分解为多个维数较小、变量较少的子辨识模型或称子系统(这些子系统可能是虚拟的). 有时这种分解是必须的,如具有公分母传递函数矩阵的辨识问题: 辨识模型既包含一个参数向量,又包含一个参数矩阵. 有时为减小计算量,也需要把一个线性辨识模型进行分解. 2) 子系统辨识或子模型辨识,即辨识每一个子系统或子模型. 忽略各子辨识模型间的交叉关联项,把耦合的关联项看作是已知的,根据最小二乘原理等分别辨识每个子系统的参数向量或参数矩阵. 3) 协调—处理子系统间关联项. 辨识模型分解为  $N$  个子辨识模型,各个子辨识模型间存在耦合关联项,即一个子模型包含其他一些子模型的未知变量. 辨识的困难在于如何处理各子系统间的关联项. 因为第  $i$  个子系统包含其他一些子系统的未知参数向量或参数矩阵,使得子系统辨识算法难以实现. 为了解决这一问题,在计算  $t$  时刻第  $i$  个子系统参数的估计时,包含在第  $i$  个子系统中的其他子系统的未知参数向量或参数矩阵,用它们在前一时刻的估计值代替,使得每个子系统的辨识方法能够实现.

递阶辨识实际上是一种交互估计理论. 递阶辨识算法也可称为松弛辨识算法(bootstrap identification algorithm).

我们已经讨论的一些辨识方法,如最小二乘法、最小均方算法等,都是采用描述输入输出关系的线性或伪线性回归辨识模型. 线性回归模型如 CAR 模型(ARX 模型),伪线性回归模型如 CARMA 模型(ARMAX 模型)、OE 模型、CARAR/ARARX 模型、CARARMA/ARARMAX 模型等,输出关于参数空间是线性的,且具有如下最小二乘格式:

$$y(t) = \theta^T \varphi(t) + v(t), \quad (1)$$

其中  $y(t)$  是系统输出变量(向量),  $v(t)$  是零均值随机白噪声(向量),  $\theta$  为参数向量(矩阵),  $\varphi(t)$  是由  $t$  时刻以前系统输入  $u(t)$  和输出  $y(t)$  等变量(向量)构成的回归信息向量. 在这类具有最小二乘格式的辨识模型中,输出  $y(t)$  关于参数空间  $\theta$  是线性的,所以其辨识问题相对比较简单.

然而,有些模型是难以写成如式(1)的最小二乘格式的. 例如,对于状态空间模型

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) + w(t), \\ y(t) = Cx(t) + Du(t) + v(t), \end{cases}$$

其输出与系统参数矩阵是非线性关系:

$$y(t) = C(zI - A)^{-1} [Bu(t) + w(t)] + Du(t) + v(t),$$

其中  $I$  是一个适当维数的单位阵,  $x(t) \in \mathbf{R}^n$  为未知状态向量,  $u(t) \in \mathbf{R}^r$  为输入向量,  $y(t) \in \mathbf{R}^m$  为输出向量,  $w(t) \in \mathbf{R}^n$  为零均值过程噪声向量,  $v(t) \in \mathbf{R}^m$  为零均值观测噪声向量,  $A \in \mathbf{R}^{n \times n}$ ,  $B \in \mathbf{R}^{n \times r}$ ,  $C \in \mathbf{R}^{m \times n}$ ,  $D \in \mathbf{R}^{m \times r}$  为未知系统参数矩阵. 状态空间辨识模型要复杂得多,它既包含系统未知参数矩阵,又包含未知系统状态,且是它们的乘积关系的非线性函数. 这使得研究状态空间模型的辨识方法更为困难. 状态空间模型的辨识可采用近 20 年发展起来的子空间系统辨识方法(subspace state space identification method),简称为 4SID. 4SID 已经形成一个辨识分支. 另一些状态空间模型辨识方法是递阶辨识,可参见文献[11-13, 34-35].

本文作者于 1999 年提出的递阶辨识原理及其交互估计理论,是专门用来解决这类复杂关系、非常规辨识模型的辨识问题的. 递阶辨识模型通常要复杂得多,它可能包含系统的未知参数向量和未知参数矩阵,甚至包含未知系统状态,还可能是它们乘积关系的非线性函数,一般不具备最小二乘格式,如双线性参数系统、多线性参数系统、非线性系统等. 这使得研究这类模型的辨识方法更为困难.

递阶辨识模型一般可表示为

$$y(t) = H(\psi_1(t), \psi_2(t), \dots, \psi_m(t), \theta_1, \theta_2, \dots, \theta_r) + v(t), \quad (2)$$

其中  $\psi_i(t)$  ( $i = 1, 2, \dots, m$ ) 是由系统输入输出数据构成的回归向量(矩阵),  $\theta_i$  ( $i = 1, 2, \dots, r$ ) 是系统待辨识的参数向量或参数矩阵.  $\psi_i(t)$  可以是向量,也可以是矩阵,各  $\psi_i(t)$  的维数可以不相等; 同样,  $\theta_i$  可以是向量,也可以是矩阵,各  $\theta_i$  的维数也可以不相等.

下面列举一些递阶辨识模型:

- 1)  $y(t) = \psi_1(t) \theta_1 + \theta_2 \psi_2(t) + v(t)$ ,
- 2)  $y(t) = \psi_1(t) \theta_1 + \theta_2 \psi_2(t) \theta_3 + v(t)$ ,
- 3)  $y(t) = \theta_1 \psi_1(t) + \theta_2 \psi_2(t) \theta_3 + v(t)$ .

线性辨识模型(1)也可以采用基于分解的方法,提出递阶最小二乘辨识方法,来减小计算量,可参见本文第6节和文献[8-28].

有些递阶辨识模型通过重新参数化,可以化为最小二乘格式,但是这种参数化会导致参数向量维数大大增加,增加算法的计算量.例如,一个  $m \times n$  维参数矩阵,化为参数向量为  $mn$  维,则最小二乘参数估计的协方差矩阵由  $m \times m$  维或  $n \times n$  维增加到  $mn \times mn$  维,计算量大大增加.例如,Hammerstein 非线性系统经过参数化得到下列递阶辨识模型<sup>[18]</sup>:

$$y(t) = \varphi^T(t) a + b^T F(t) c + v(t),$$

其中  $y(t)$  是系统输出,  $\varphi(t)$  是信息向量,  $F(t)$  是系统输入的非线性函数构成的回归矩阵,  $v(t)$  是干扰噪声,  $a \in \mathbf{R}^{n_a}$ ,  $b \in \mathbf{R}^{n_b}$  和  $c \in \mathbf{R}^{n_c}$  是系统参数向量. 这个非线性辨识模型包含了系统非线性部分参数与线性部分参数的乘积,可以用递阶辨识原理导出其辨识方法,可参见文献[18]. 如果对该模型进行参数化,导出一个最小二乘辨识模型,其参数向量的维数大大增加(因为参数向量包含了线性部分参数与非

线性部分参数的乘积项,参数向量  $\begin{bmatrix} a \\ b \\ c \end{bmatrix} \in \mathbf{R}^{n_a + n_b + n_c}$  增

大到  $\begin{bmatrix} a \\ b \otimes c \end{bmatrix} \in \mathbf{R}^{n_a + n_b n_c}$ ,会增加算法的计算量,这种方法称为过参数化方法(over-parameterization method),可参见文献[14-16].

一个多变量系统传递函数矩阵模型,经过参数化得到下列最典型的递阶辨识模型<sup>[9-10]</sup>:

$$y(t) + \psi(t) \alpha = \theta^T \varphi(t) + w(t),$$

其中  $y(t) \in \mathbf{R}^m$  为系统输出,  $\psi(t) \in \mathbf{R}^{m \times n}$  为输出信息矩阵,  $\varphi(t) \in \mathbf{R}^{n_0}$  为输入信息向量,  $w(t) \in \mathbf{R}^m$  为噪声向量. 这个模型辨识的困难在于:既包含一个参数向量  $\alpha \in \mathbf{R}^n$  (系统特征多项式的系数),又包含了一个参数向量  $\theta^T \in \mathbf{R}^{m \times n_0}$ . 这类复杂模型的辨识可以用递阶辨识原理解决,参见本文第7节.

## 2 迭代方法族

本节先介绍线性方程组  $Ax = b$  著名的雅可比迭代方法(Jacobi iterative method)和高斯-赛德尔迭代方法(Gauss-Seidel iterative method),接着讨论本文

作者提出的一大类迭代方法族,进一步把迭代方法族推广到矩阵方程  $AXB = F$ ,提出相应的梯度迭代方法和最小二乘迭代方法,本节主要内容选自文献[20-22-24].

### 2.1 雅可比和高斯-赛德尔迭代

考虑3个未知数  $(x_1, x_2, x_3)$  的线性方程组:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3. \end{cases}$$

这里  $a_{ij}$  为系数,  $b_i$  为常数. 设  $a_{ii} \neq 0$ , 从这3个方程可以分别求出:

$$\begin{cases} x_1 = \frac{1}{a_{11}} [b_1 - a_{12}x_2 - a_{13}x_3], \\ x_2 = \frac{1}{a_{22}} [b_2 - a_{21}x_1 - a_{23}x_3], \\ x_3 = \frac{1}{a_{33}} [b_3 - a_{31}x_1 - a_{32}x_2]. \end{cases}$$

但是,上述方程右边存在未知数,所以无法求解. 解决的方法之一是采用迭代原理: 设  $k = 1, 2, 3, \dots$  为迭代变量,  $x_i(k)$  是  $x_i$  的迭代解, 给定  $x_i(0)$ , 那么可通过下列方程获得迭代解(iterative solution):

$$\begin{cases} x_1(k) = \frac{1}{a_{11}} [b_1 - a_{12}x_2(k-1) - a_{13}x_3(k-1)], \\ x_2(k) = \frac{1}{a_{22}} [b_2 - a_{21}x_1(k-1) - a_{23}x_3(k-1)], \\ x_3(k) = \frac{1}{a_{33}} [b_3 - a_{31}x_1(k-1) - a_{32}x_2(k-1)]. \end{cases}$$

这就是雅可比迭代(Jacobi iteration).

这里是依次计算  $x_1(k), x_2(k), x_3(k)$  的. 由于在计算  $x_2(k)$  时  $x_1(k)$  已经得到,故上述第2个方程中的  $x_1(k-1)$  可用  $x_1(k)$  代替; 同理,在计算  $x_3(k)$  时,  $x_1(k)$  和  $x_2(k)$  已经得到,就用  $x_1(k)$  和  $x_2(k)$  代替上述第3个方程中的  $x_1(k-1)$  和  $x_2(k-1)$ , 就得到高斯-赛德尔迭代(Gauss-Seidel iteration):

$$\begin{cases} x_1(k) = \frac{1}{a_{11}} [b_1 - a_{12}x_2(k-1) - a_{13}x_3(k-1)], \\ x_2(k) = \frac{1}{a_{22}} [b_2 - a_{21}x_1(k) - a_{23}x_3(k-1)], \\ x_3(k) = \frac{1}{a_{33}} [b_3 - a_{31}x_1(k) - a_{32}x_2(k)]. \end{cases}$$

如果按照逆序计算  $x_3(k), x_2(k), x_1(k)$  (当然,也可以先计算  $x_2(k)$ ), 则得到另一种形式的高斯-赛德尔迭代算法:

$$\begin{cases} x_1(k) = \frac{1}{a_{11}} [b_1 - a_{12}x_2(k) - a_{13}x_3(k)], \\ x_2(k) = \frac{1}{a_{22}} [b_2 - a_{21}x_1(k-1) - a_{23}x_3(k)], \\ x_3(k) = \frac{1}{a_{33}} [b_3 - a_{31}x_1(k-1) - a_{32}x_2(k-1)]. \end{cases}$$

推广到一般情形, 考虑  $n$  个变量  $x = [x_1, x_2, \dots, x_n]^T \in \mathbf{R}^n$  的矩阵方程

$$Ax = b, \quad (3)$$

其中

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \in \mathbf{R}^{n \times n}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \in \mathbf{R}^n.$$

把矩阵  $A$  分解为严格下三角阵 (strictly lower triangle matrix)  $L$ , 对角阵 (diagonal matrix)  $D$  和严格上三角阵 (strictly upper triangle matrix)  $U$ , 它们满足

$$A = L + D + U,$$

其中

$$D = \text{diag} [a_{11}, a_{22}, \dots, a_{nn}] \in \mathbf{R}^{n \times n},$$

$$L = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ a_{21} & 0 & \ddots & & \vdots \\ a_{31} & a_{32} & 0 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ a_{n1} & a_{n2} & \cdots & a_{n, n-1} & 0 \end{bmatrix} \in \mathbf{R}^{n \times n},$$

$$U = \begin{bmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & & a_{2n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & a_{n-1, n} \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix} \in \mathbf{R}^{n \times n}.$$

那么雅可比和高斯-赛德尔迭代具有下列一般形式:

$$Mx(k) = Nx(k-1) + b, \quad k = 1, 2, 3, \dots$$

当  $M = D, N = -(L + U)$  时, 得到雅可比迭代算法:

$$Dx(k) = -(L + U)x(k-1) + b.$$

当  $M = L + D, N = -U$  时, 得到高斯-赛德尔迭代算法:

$$(L + D)x(k) = -Ux(k-1) + b.$$

当  $A$  为对称正定阵, 或为对角优势矩阵, 雅可比迭代和高斯-赛德尔迭代可以保证迭代解  $x(k)$  收敛于精确解  $x = A^{-1}b$ , 即  $x(k) \rightarrow x = A^{-1}b$ .

一般情况下, 雅可比和高斯-赛德尔迭代不能保证迭代解收敛于精确解, 也不适合求解非方系统, 即  $A \in \mathbf{R}^{m \times n}$  为非方阵. 这促使我们研究雅可比迭代和

高斯-赛德尔迭代算法的收敛条件, 以及新的迭代方法族.

## 2.2 矩阵方程 $Ax = b$ 迭代方法族

对于方程  $Ax = b$ , 设  $G \in \mathbf{R}^{n \times n}$  为一个满秩待定矩阵,  $\mu > 0$  为迭代步长或收敛因子. 我们提出的一大类迭代方法族 (a large family of iterative methods) 如下<sup>[20, 22-23]</sup>:

$$x(k) = x(k-1) + \mu G_k [b - Ax(k-1)], \quad k = 1, 2, 3, \dots \quad (4)$$

这个迭代方法包括雅可比迭代方法和高斯-赛德尔迭代方法作为特例. 当  $G_k = D^{-1}$  和  $\mu = 1$  时, 得到雅可比迭代方法; 当  $G_k = (L + D)^{-1}$  和  $\mu = 1$  时, 得到高斯-赛德尔迭代方法.

定理 1<sup>[20]</sup> 假设线性系统 (3) 有唯一解, 则对于任意有限初值  $x(0)$ , 算法 (4) 给出的迭代解  $x(k)$  收敛于精确解  $x$  (即  $\lim_{k \rightarrow \infty} nx(k) = x$ ) 的充分条件是下列不等式成立:

$$\mu (G_k A)^T (G_k A) + \varepsilon I \leq (G_k A)^T + (G_k A) \quad \text{for all } k, \quad (5)$$

其中  $\varepsilon$  是一个不依赖  $k$  的小正常数. 事实上, 如果  $(G_k A)^T + (G_k A)$  是正定阵, 那么收敛因子的一个保守选择是

$$0 < \mu \leq \frac{\lambda_{\min} [(G_k A)^T + (G_k A)]}{\lambda_{\max} [(G_k A)^T (G_k A)]} \quad \text{for all } k,$$

其中  $\lambda_{\max} [X]$  ( $\lambda_{\min} [X]$ ) 代表方阵  $X$  的最大 (最小) 特征值.

值得注意的是, 对于时不变系统:  $x(k) = Hx(k-1)$ ,  $H \in \mathbf{R}^{n \times n}$ ,  $H$  的所有特征值都在单位圆内, 可以保证  $x(k)$  收敛于零. 但是, 对于时变系统  $x(k) = H_k x(k-1)$ ,  $H_k \in \mathbf{R}^{n \times n}$ ,  $H_k$  的特征值在单位圆内或在单位圆外, 既不是时变系统稳定的必要条件 (necessary condition), 也不是充分条件 (sufficient condition)<sup>[36]</sup>.

$(I - D^{-1}A)$  的特征值在单位圆内保证雅可比迭代解收敛于精确解,  $[I - (L + D)^{-1}A]$  的特征值在单位圆内, 就保证高斯-赛德尔迭代解收敛于精确解. 当引入收敛因子  $\mu$  后, 它们的收敛条件得以减弱, 变为  $(I - \mu D^{-1}A)$  和  $[I - \mu(L + D)^{-1}A]$  的特征值在单位圆内. 从定理 1 可以得到下列两个重要推论<sup>[20]</sup>.

推论 1 取  $G_k = A^T$  ( $A$  可以是一个列满秩  $m \times n$  非方矩阵), 则梯度迭代算法 (Gradient Iterative algorithm, GI 算法):

$$\begin{cases} \mathbf{x}(k) = \mathbf{x}(k-1) + \mu \mathbf{A}^T [\mathbf{b} - \mathbf{A}\mathbf{x}(k-1)], \\ \mu \leq 2\{\lambda_{\max}[\mathbf{A}^T \mathbf{A}]\}^{-1} \text{ 或 } \mu \leq \frac{2}{\|\mathbf{A}\|^2}, \end{cases} \quad (6)$$

保证  $\mathbf{x}(k) \rightarrow \mathbf{x}$ . 矩阵范数定义为  $\|\mathbf{X}\|^2 = \text{tr}[\mathbf{X}\mathbf{X}^T]$ . 因为算法(6)可以通过极小化  $J(\mathbf{x}) := \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ , 使用负梯度搜索得到, 故称为梯度迭代算法.

推论 2 取  $\mathbf{G}_k = \mathbf{A}^{-1}$ , 则下列最小二乘迭代算法产生的迭代解收敛于  $\mathbf{x}$ :

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{x}(k-1) + \mu \mathbf{A}^{-1} [\mathbf{b} - \mathbf{A}\mathbf{x}(k-1)], \\ 0 < \mu &\leq 2. \end{aligned} \quad (7)$$

如果  $\mathbf{A}$  是一个列满秩  $m \times n$  非方矩阵, 那么下列最小二乘迭代算法(Least Squares based Iterative algorithm, LSI 算法)保证  $\mathbf{x}(k) \rightarrow \mathbf{x}$ :

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{x}(k-1) + \mu (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T [\mathbf{b} - \mathbf{A}\mathbf{x}(k-1)], \\ 0 < \mu &\leq 2. \end{aligned} \quad (8)$$

当  $\mu = 1$  时, 用这个算法只需计算一步, 就得到  $\mathbf{A}\mathbf{x} = \mathbf{b}$  的最小二乘解(least squares solution):  $\mathbf{x}(1) = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ , 故该算法具有最快的收敛速度. 因此, 迭代算法(8)也称为最小二乘迭代算法, 或迭代最小二乘算法. 尽管这个迭代算法需要计算矩阵的逆, 但是对求解耦合矩阵(如  $\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{B} = \mathbf{F}$ )的最小二乘迭代解非常有用, 因为  $\mathbf{A}$  是常数矩阵, 故只需计算一次矩阵逆.

从推论 1—2 可知, 梯度迭代算法(6)在确定收敛因子时, 都需要计算矩阵特征值, 支出附加的计算量. 梯度迭代算法(6)和最小二乘迭代算法(8)还适合非方系统, 故可以用来计算非方系统(non-square system of linear equations)的迭代解, 且最小二乘迭代算法的收敛因子并不依赖矩阵  $\mathbf{A}$ , 收敛因子容易选择, 它只需要计算一次矩阵逆.

### 2.3 矩阵方程 $\mathbf{A}\mathbf{X}\mathbf{B} = \mathbf{F}$ 的迭代解

考虑矩阵方程

$$\mathbf{A}\mathbf{X}\mathbf{B} = \mathbf{F}, \quad (9)$$

其中  $\mathbf{A} \in \mathbf{R}^{p \times m}$ ,  $\mathbf{B} \in \mathbf{R}^{n \times q}$  和  $\mathbf{F} \in \mathbf{R}^{p \times q}$  是给定常数矩阵,  $\mathbf{X} \in \mathbf{R}^{m \times n}$  是未知矩阵.

引理 1<sup>[22]</sup> 如果  $\mathbf{A}$  (列满秩),  $\mathbf{B}$  (行满秩) (隐含  $p \geq m$ ,  $n \leq q$ ), 则在最小二乘意义下, 方程(9)有唯一解

$$\mathbf{X} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{F} \mathbf{B}^T (\mathbf{B} \mathbf{B}^T)^{-1},$$

而齐次矩阵方程(homogeneous matrix equation)  $\mathbf{A}\mathbf{X}\mathbf{B} = \mathbf{0}$  有零解  $\mathbf{X} = \mathbf{0}$ .

根据推论 1 和推论 2, 可以得出求解方程(9)的梯度迭代算法和最小二乘迭代算法, 总结为以下

定理.

定理 2<sup>[22]</sup> 在引理 1 条件下, 梯度迭代算法(Gradient based Iterative algorithm, GI 算法):

$$\mathbf{X}(k) = \mathbf{X}(k-1) + \mu \mathbf{A}^T [\mathbf{F} - \mathbf{A}\mathbf{X}(k-1)\mathbf{B}]\mathbf{B}^T, \quad (10)$$

$$0 < \mu \leq \frac{2}{\lambda_{\max}[\mathbf{A}^T \mathbf{A}]\lambda_{\max}[\mathbf{B}\mathbf{B}^T]} \text{ 或 } \mu \leq \frac{2}{\|\mathbf{A}\|^2 \|\mathbf{B}\|^2}, \quad (11)$$

和最小二乘迭代算法(Least Squares based Iterative algorithm, LSI 算法):

$$\begin{aligned} \mathbf{X}(k) &= \mathbf{X}(k-1) + \mu (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T [\mathbf{F} - \mathbf{A}\mathbf{X}(k-1)\mathbf{B}]\mathbf{B}^T (\mathbf{B}\mathbf{B}^T)^{-1}, \\ 0 < \mu &\leq 2. \end{aligned} \quad (12)$$

保证  $\mathbf{X}(k) \rightarrow \mathbf{X}$ .

## 3 一般矩阵方程

本节运用递阶辨识原理推导西尔维斯特矩阵方程的递阶梯度迭代求解算法, 主要内容选自文献[21-22].

### 3.1 西尔维斯特矩阵方程 $\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{B} = \mathbf{F}$

对于矩阵  $\mathbf{A} = (a_{ij}) \in \mathbf{R}^{m \times n}$  和  $\mathbf{B} = (b_{ij}) \in \mathbf{R}^{p \times r}$ ,  $\mathbf{A} \otimes \mathbf{B}$  表示它们的克罗内克积(Kronecker product)或直积, 定义为  $\mathbf{A} \otimes \mathbf{B} = (a_{ij}\mathbf{B}) \in \mathbf{R}^{mp \times nr}$ .

对于  $m \times n$  矩阵  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbf{R}^{m \times n}$ ,  $\mathbf{x}_i \in \mathbf{R}^m$ ,  $i = 1, 2, \dots, n$ ,  $\text{col}[\mathbf{X}]$  表示由矩阵  $\mathbf{X}$  的列构成的向量, 即

$$\text{col}[\mathbf{X}] = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} \in \mathbf{R}^{mn}.$$

考虑下列西尔维斯特矩阵方程(Sylvester matrix equation)的求解问题:

$$\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{B} = \mathbf{F}, \quad (13)$$

其中  $\mathbf{A} \in \mathbf{R}^{m \times m}$ ,  $\mathbf{B} \in \mathbf{R}^{n \times n}$  和  $\mathbf{F} \in \mathbf{R}^{m \times n}$  均是已知常数矩阵,  $\mathbf{X} \in \mathbf{R}^{m \times n}$  是待求的未知矩阵.

引理 2 设  $\lambda_i[\mathbf{X}]$  表示矩阵  $\mathbf{X}$  的第  $i$  个特征值. 矩阵方程(13)有唯一解的充分必要条件是: 对任意的  $i$  和  $j$ , 有  $\lambda_i[\mathbf{A}] + \lambda_j[\mathbf{B}] \neq 0$ . 这个唯一解可表示为

$$\text{col}[\mathbf{X}] = [(\mathbf{I}_n \otimes \mathbf{A}) + (\mathbf{B}^T \otimes \mathbf{I}_m)]^{-1} \text{col}[\mathbf{F}], \quad (14)$$

对应的齐次方程  $\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{B} = \mathbf{0}$  有唯一解  $\mathbf{X} = \mathbf{0}$ .

特别地, 当  $\mathbf{B} = \mathbf{A}^T$  时, 式(13)退化为连续时间李雅普诺夫矩阵方程. 存在唯一解的充分必要条件是对于任意的  $i$  和  $j$ ,  $\lambda_i[\mathbf{A}] + \lambda_j[\mathbf{A}] \neq 0$ .

虽然式(14)可以用来求解矩阵方程(13), 但是随着矩阵  $\mathbf{X}$  维数的增加, 所需的存储量迅速增加, 因

为用式(14)计算  $X$ , 需要求  $(mn) \times (mn)$  维矩阵的逆, 使得存储量大大增加. 因此, 有必要提出一个实用算法迭代计算矩阵方程(13)的解  $X$ .

根据递阶辨识原理, 把系统(13)分解为两个子系统, 且认为  $X$  为系统的待辨识参数矩阵, 然后利用最小二乘优化原理(least squares optimization principle)和推论 1, 分别导出每个子系统参数矩阵的迭代解, 具体步骤如下.

1) 系统分解

定义两个矩阵:

$$b_1 := F - XB, \quad (15)$$

$$b_2 := F - AX. \quad (16)$$

那么从式(13)可以得到两个虚拟子系统:

$$S_1: AX = b_1,$$

$$S_2: XB = b_2.$$

2) 子系统迭代解

对于子系统  $S_1$  和  $S_2$ , 分别构造两个准则函数:

$$J_1(X) := \|AX - b_1\|^2,$$

$$J_2(X) := \|XB - b_2\|^2,$$

分别计算其梯度

$$\text{grad}[J_1(X)] = 2A^T[AX - b_1],$$

$$\text{grad}[J_2(X)] = 2[XB - b_2]B^T,$$

令  $X_1(t)$  和  $X_2(t)$  分别对应于  $S_1$  和  $S_2$  参数矩阵  $X$  的估计或迭代解, 用负梯度搜索极小化  $J_1(X)$  和  $J_2(X)$  或者应用推论 1 于系统  $S_1$  和  $S_2$ , 直接导出下列迭代方程:

$$X_1(t) = X_1(t-1) + \mu A^T [b_1 - AX_1(t-1)], \quad (17)$$

$$X_2(t) = X_2(t-1) + \mu [b_2 - X_2(t-1)B]B^T. \quad (18)$$

这里  $\mu$  称为迭代步长或收敛因子, 可以选择为

$$\mu = \frac{1}{\|A\|^2 + \|B\|^2}.$$

将式(15)代入式(17), 式(16)代入式(18)得到  $X_1(t) = X_1(t-1) + \mu A^T [F - AX_1(t-1) - XB]$ , (19)

$$X_2(t) = X_2(t-1) + \mu [F - AX - X_2(t-1)B]B^T. \quad (20)$$

因为式(19)和(20)右边包含了未知矩阵  $X$ , 所以不可能实现算法(19)和(20).

3) 关联项协调

为了处理两个算法(19) — (20) 右边的关联项  $X$ , 利用递阶辨识原理: 式(19)和(20)中未知  $X$  分别用其在  $(t-1)$  时刻的估计值  $X_1(t-1)$  和  $X_2(t-1)$  代替, 得到

$$X_1(t) = X_1(t-1) + \mu A^T [F - AX_1(t-1) - X_1(t-1)B], \quad (21)$$

$$X_2(t) = X_2(t-1) + \mu [F - AX_2(t-1) - X_2(t-1)B]B^T. \quad (22)$$

事实上, 只需要一个迭代解  $X(t)$ , 而不是两个解  $X_1(t)$  和  $X_2(t)$ . 一种方法是取它们的平均值作为  $X(t)$ , 即得到计算矩阵方程(13)迭代解的递阶梯度迭代算法(Hierarchical Gradient based Iterative algorithm, HGI 算法)<sup>[21-22]</sup>:

$$X(t) = \frac{X_1(t) + X_2(t)}{2}, \quad (23)$$

$$X_1(t) = X(t-1) + \frac{A^T [F - AX(t-1) - X(t-1)B]}{\|A\|^2 + \|B\|^2}, \quad (24)$$

$$X_2(t) = X(t-1) + \frac{[F - AX(t-1) - X(t-1)B]B^T}{\|A\|^2 + \|B\|^2}. \quad (25)$$

该算法名称来历是使用了递阶辨识原理和梯度搜索的迭代算法, 其初值可取为  $X(0) = \mathbf{1}_{m \times n} / p_0$ ,  $p_0 = 10^6$ ,  $\mathbf{1}_{m \times n}$  是其元均为 1 的  $m \times n$  矩阵, 即

$$\mathbf{1}_{m \times n} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \in \mathbf{R}^{m \times n}.$$

定理 3<sup>[21-22]</sup> 如果方程(13)在最小二乘意义下有唯一解  $X$ , 那么对于任意有限初值  $X(0)$ , 算法(23) — (25) 给出的迭代解  $X(t)$  收敛于  $X$ , 即  $\lim_{t \rightarrow \infty} X(t) = X$  或估计误差  $X(t) - X$  收敛于零.

证明 定义估计误差矩阵:

$$\tilde{X}_1(t) := X_1(t) - X,$$

$$\tilde{X}_2(t) := X_2(t) - X.$$

那么有

$$\tilde{X}(t) = X(t) - X = \frac{\tilde{X}_1(t) + \tilde{X}_2(t)}{2}. \quad (26)$$

令

$$\xi(t) := A \tilde{X}(t-1), \quad (27)$$

$$\eta(t) := \tilde{X}(t-1)B. \quad (28)$$

利用式(13)和(23) — (25), 可得

$$\tilde{X}_1(t) = \tilde{X}(t-1) + \frac{A^T [-A \tilde{X}(t-1) - \tilde{X}(t-1)B]}{\|A\|^2 + \|B\|^2} =$$

$$\tilde{X}(t-1) + \frac{A^T [-\xi(t) - \eta(t)]}{\|A\|^2 + \|B\|^2},$$

$$\tilde{X}_2(t) = \tilde{X}(t-1) + \frac{[-A \tilde{X}(t-1) - \tilde{X}(t-1)B]B^T}{\|A\|^2 + \|B\|^2} =$$

$$\tilde{X}(t-1) + \frac{[-\xi(t) - \eta(t)]B^T}{\|A\|^2 + \|B\|^2}.$$

使用式(27)和(28)可得

$$\begin{aligned} & \| \tilde{X}_1(t) \|^2 = \text{tr} [ \tilde{X}_1^T(t) \tilde{X}_1(t) ] = \\ & \| \tilde{X}(t-1) \|^2 + \frac{2\text{tr}\{ \tilde{X}^T(t-1) A^T [ -\xi(t) - \eta(t) ] \}}{\| A \|^2 + \| B \|^2} + \\ & \frac{\| A^T [ -\xi(t) - \eta(t) ] \|^2}{(\| A \|^2 + \| B \|^2)^2} \leq \\ & \| \tilde{X}(t-1) \|^2 + \frac{2\text{tr}\{ \xi^T(t) [ -\xi(t) - \eta(t) ] \}}{\| A \|^2 + \| B \|^2} + \\ & \frac{\| A \|^2 \| \xi(t) + \eta(t) \|^2}{(\| A \|^2 + \| B \|^2)^2}. \end{aligned} \quad (29)$$

类似地, 有

$$\begin{aligned} & \| \tilde{X}_2(t) \|^2 = \text{tr} [ \tilde{X}_2^T(t) \tilde{X}_2(t) ] \leq \\ & \| \tilde{X}(t-1) \|^2 + \frac{2\text{tr}\{ [ -\xi(t) - \eta(t) ] \eta^T(t) \}}{\| A \|^2 + \| B \|^2} + \\ & \frac{\| B \|^2 \| \xi(t) + \eta(t) \|^2}{(\| A \|^2 + \| B \|^2)^2}. \end{aligned} \quad (30)$$

利用式(29)和(30), 从式(26)可得

$$\begin{aligned} & \| \tilde{X}(t) \|^2 = \frac{\| \tilde{X}_1(t) + \tilde{X}_2(t) \|^2}{4} \leq \\ & \frac{\| \tilde{X}_1(t) \|^2 + \| \tilde{X}_2(t) \|^2}{2} \leq \\ & \| \tilde{X}(t-1) \|^2 - \frac{\| \xi(t) + \eta(t) \|^2}{2(\| A \|^2 + \| B \|^2)} \leq \\ & \| \tilde{X}(0) \|^2 - \sum_{i=1}^t \frac{\| \xi(i) + \eta(i) \|^2}{2(\| A \|^2 + \| B \|^2)}. \end{aligned}$$

于是有

$$\sum_{i=1}^{\infty} \frac{\| \xi(i) + \eta(i) \|^2}{2(\| A \|^2 + \| B \|^2)} < \infty,$$

和

$$\xi(t) + \eta(t) \rightarrow 0, \text{ 当 } t \rightarrow \infty,$$

或

$$A \tilde{X}(t-1) + \tilde{X}(t-1) B \rightarrow 0, \text{ 当 } t \rightarrow \infty.$$

根据引理2可知:  $\tilde{X}(t-1) \rightarrow 0$ . 定理3证毕.

算法(21)或(22)称为单边迭代算法(single-side iterative algorithm), 它不能保证  $X_i(t)$  ( $i=1, 2$ ) 收敛于  $X$ , 而(23)~(25)称为平衡迭代算法(balanced iterative algorithm), 它可以简单写为

$$\begin{aligned} X(t) = & X(t-1) + \mu A^T [ F - AX(t-1) - \\ & X(t-1) B ] / 2 + \mu [ F - AX(t-1) - \\ & X(t-1) B ] B^T / 2. \end{aligned} \quad (31)$$

为了加快算法的收敛性能, 从定理3证明过程可以看出: 收敛因子也可选择为

$$0 < \mu \leq \frac{2}{\lambda_{\max} [ A^T A ] + \lambda_{\max} [ B B^T ]}. \quad (32)$$

这个收敛因子并不是最佳的(保守的). 事实上, 存在一个最佳的  $\mu$  以致于  $X(t)$  收敛于  $X$  有更快的收敛速度.

### 3.2 矩阵方程 $AXB + X = F$

下面研究矩阵方程

$$AXB + X = F \quad (33)$$

的迭代解, 其中  $A \in \mathbf{R}^{m \times m}$ ,  $B \in \mathbf{R}^{n \times n}$  和  $F \in \mathbf{R}^{m \times n}$  为给定常数矩阵,  $X \in \mathbf{R}^{m \times n}$  是未知矩阵. 特别地, 当  $A = B^T$  时, 式(33)是离散时间李雅普诺夫矩阵方程. 此外, 如果式中  $B$  是可逆的, 用  $B^{-1}$  右乘上述方程, 则得到式(13)的形式.

但是, 一个简单的迭代算法

$$X(t) = F - AX(t-1)B$$

不能保证  $X(t)$  收敛于  $X$ .

引理3 矩阵方程(33)有唯一解的充分必要条件是: 对于任意  $i$  和  $j$ ,  $\lambda_i [A] \lambda_j [B] \neq -1$ . 这个唯一解为

$$\text{col} [X] = [ (B^T \otimes A) + I_{mn} ]^{-1} \text{col} [F],$$

对应齐次方程  $AXB + X = 0$  有唯一零解  $X = 0$ .

类似于上面的推导, 应用定理2, 能够得到计算矩阵方程(33)迭代解的递阶梯度迭代算法<sup>[21-22]</sup>:

$$X(t) = \frac{X_1(t) + X_2(t)}{2}, \quad (34)$$

$$\begin{aligned} X_1(t) = & X(t-1) + \\ & \mu A^T [ F - AX(t-1)B - X(t-1) ] B^T, \end{aligned} \quad (35)$$

$$\begin{aligned} X_2(t) = & X(t-1) + \\ & \mu [ F - AX(t-1)B - X(t-1) ], \end{aligned} \quad (36)$$

$$\mu = \frac{2}{\lambda_{\max} [ A^T A ] \lambda_{\max} [ B B^T ] + 1} =: 2\mu_0, \text{ 或}$$

$$0 < \mu \leq \frac{2}{\| A \|^2 \| B \|^2 + 1}. \quad (37)$$

定理4 如果矩阵方程有唯一解  $X$ , 那么算法(34)~(37)给出的迭代解  $X(t)$  收敛于  $X$ , 即  $\lim_{t \rightarrow \infty} X(t) = X$ .

### 3.3 矩阵方程 $AXB + CXD = F$

本小节主要内容选自文献[21-22].

应用递阶辨识原理求解下列广义西尔维斯特矩阵方程(generalized Sylvester matrix equation)的迭代解:

$$AXB + CXD = F, \quad (38)$$

其中  $A, C \in \mathbf{R}^{m \times m}$ ,  $B, D \in \mathbf{R}^{n \times n}$  和  $F \in \mathbf{R}^{m \times n}$  是给定常数矩阵,  $X \in \mathbf{R}^{m \times n}$  是未知矩阵.

引理4 方程(38)有唯一解, 当且仅当对任意  $i$

和  $j \lambda_i [A] + \lambda_j [B] \neq 0$ , 且这个唯一解为

$$\text{col}[X] = [B^T \otimes A + D^T \otimes C]^{-1} \text{col}[F], \quad (39)$$

对应的齐次方程  $AXB + CXD = 0$  有零解, 即  $X = 0$ .

特别地, 当  $D = A^T, B = C = I$ , 方程 (38) 退化为连续时间李雅普诺夫矩阵方程, 其唯一解存在的充分必要条件是对于任意  $i$  和  $j, \lambda_i [A] + \lambda_j [A] \neq 0$ .

根据递阶辨识原理, 把系统 (38) 分解为两个子系统, 然后分别计算每个子系统的迭代解, 方法如下. 定义两个矩阵:

$$b_1 := F - CXD, \quad (40)$$

$$b_2 := F - AXB. \quad (41)$$

那么, 从 (38) 可以得到两个虚拟子系统:

$$S_3: \quad AXB = b_1, \quad (42)$$

$$S_4: \quad CXD = b_2. \quad (43)$$

根据定理 2 的式 (10), 容易写出子系统  $S_3$  和  $S_4$  的梯度迭代解  $X_1(t)$  和  $X_2(t)$ :

$$X_1(t) = X_1(t-1) + \mu A^T [b_1 - AX_1(t-1)B]B^T,$$

$$X_2(t) = X_2(t-1) + \mu C^T [b_2 - CX_2(t-1)D]D^T.$$

收敛因子  $\mu$  在后面给出. 将式 (40) 和 (41) 分别代入上两式, 得到

$$X_1(t) = X_1(t-1) + \mu A^T [F - AX_1(t-1)B - CXD]B^T,$$

$$X_2(t) = X_2(t-1) + \mu C^T [F - AXB - CX_2(t-1)D]D^T.$$

同样, 因为上述两个表达式的右边包含了未知的矩阵  $X$ , 故不可能实现这个算法. 解决的方法还是基于递阶辨识原理: 两式中的  $X$  分别用它们在前一时刻的估计值  $X_1(t-1)$  和  $X_2(t-1)$  代替, 得到

$$X_1(t) = X_1(t-1) + \mu A^T [F - AX_1(t-1)B - CX_1(t-1)D]B^T, \quad (42)$$

$$X_2(t) = X_2(t-1) + \mu C^T [F - AX_2(t-1)B - CX_2(t-1)D]D^T. \quad (43)$$

取  $X_1(t)$  和  $X_2(t)$  的平均值作为  $X$  的迭代解  $X(t)$ , 即

$$X(t) = \frac{X_1(t) + X_2(t)}{2}$$

保证迭代方程的收敛性, 收敛因子可以选择为

$$0 < \mu \leq \frac{2}{\lambda_{\max}[AA^T] \lambda_{\max}[BB^T] + \lambda_{\max}[CC^T] \lambda_{\max}[DD^T]} =: 2\mu_0.$$

由此可得广义西尔维斯特矩阵方程的递阶梯度迭代算法<sup>[21-22]</sup>:

$$X(t) = \frac{X_1(t) + X_2(t)}{2}, \quad (44)$$

$$X_1(t) = X(t-1) + \mu A^T [F - AX(t-1)B - CX(t-1)D]B^T, \quad (45)$$

$$X_2(t) = X(t-1) + \mu C^T [F - AX(t-1)B - CX(t-1)D]D^T, \quad (46)$$

$$0 < \mu \leq \frac{2}{\lambda_{\max}[AA^T] \lambda_{\max}[BB^T] + \lambda_{\max}[CC^T] \lambda_{\max}[DD^T]} =: 2\mu_0. \quad (47)$$

其初值仍选择为  $X(0) = 1_{m \times n} / p_0, p_0 = 10^6$ .

定理 5<sup>[22]</sup> 如果方程 (38) 有唯一解 (或最小二乘解)  $X$ , 那么对任意有限的初值  $X(0)$ , 迭代算法 (44) — (47) 产生的迭代解  $X(t)$  收敛于  $X$ , 即  $X(t) \rightarrow X$ .

证明 定义误差矩阵:

$$\tilde{X}_1(t) := X_1(t) - X,$$

$$\tilde{X}_2(t) := X_2(t) - X.$$

则有

$$\begin{aligned} \tilde{X}(t) &= X(t) - X = \frac{\tilde{X}_1(t) + \tilde{X}_2(t)}{2}, \\ \|\tilde{X}(t)\|^2 &= \frac{\|\tilde{X}_1(t) + \tilde{X}_2(t)\|^2}{4} \leq \\ &= \frac{\|\tilde{X}_1(t)\|^2 + \|\tilde{X}_2(t)\|^2}{2}. \end{aligned} \quad (48)$$

令

$$\xi(t) = A \tilde{X}(t-1) B, \quad (49)$$

$$\eta(t) = C \tilde{X}(t-1) D. \quad (50)$$

使用式 (38), (45) 和 (46), 可得

$$\begin{aligned} \tilde{X}_1(t) &= \tilde{X}(t-1) + \\ &\mu A^T [-A \tilde{X}(t-1)B - C \tilde{X}(t-1)D]B^T = \\ &\tilde{X}(t-1) + \mu A^T [-\xi(t) - \eta(t)]B^T, \end{aligned}$$

$$\begin{aligned} \tilde{X}_2(t) &= \tilde{X}(t-1) + \\ &\mu C^T [-A \tilde{X}(t-1)B - C \tilde{X}(t-1)D]D^T = \\ &\tilde{X}(t-1) + \mu C^T [-\xi(t) - \eta(t)]D^T. \end{aligned}$$

利用公式  $\text{tr}[AB] = \text{tr}[BA], \text{tr}[A^T] = \text{tr}[A]$ , 以及 (49) 和 (50), 有

$$\begin{aligned} \|\tilde{X}_1(t)\|^2 &= \text{tr}[\tilde{X}_1^T(t) \tilde{X}_1(t)] = \\ &\|\tilde{X}(t-1)\|^2 + 2\mu \text{tr}\{\tilde{X}^T(t-1)A^T[-\xi(t) - \eta(t)]B^T\} + \mu^2 \|A^T[-\xi(t) - \eta(t)]B^T\|^2 \leq \\ &\|\tilde{X}(t-1)\|^2 + 2\mu \text{tr}\{\xi^T(t)[- \xi(t) - \eta(t)]\} + \\ &\mu^2 \lambda_{\max}[AA^T] \lambda_{\max}[BB^T] \|\xi(t) + \eta(t)\|^2. \end{aligned}$$

同理

$$\|\tilde{X}_2(t)\|^2 = \text{tr}[\tilde{X}_2^T(t) \tilde{X}_2(t)] \leq$$

$$\| \tilde{X}(t-1) \|^2 + 2\mu t \{ [-\xi(t) - \eta(t)] \eta^T(t) \} + \mu^2 \lambda_{\max} [CC^T] \lambda_{\max} [DD^T] \| \xi(t) + \eta(t) \|^2.$$

将上述两式代入(48)可得

$$\begin{aligned} \| \tilde{X}(t) \|^2 &\leq \| \tilde{X}(t-1) \|^2 - \\ &\mu \| \xi(t) + \eta(t) \|^2 + \frac{\mu^2}{2} \{ \lambda_{\max} [AA^T] \lambda_{\max} [BB^T] + \\ &\lambda_{\max} [CC^T] \lambda_{\max} [DD^T] \} \| \xi(t) + \eta(t) \|^2 = \\ &\| \tilde{X}(t-1) \|^2 - \frac{\mu}{2} \left( 2 - \frac{\mu}{\mu_0} \right) \| \xi(t) + \\ &\eta(t) \|^2 \| \tilde{X}(0) \|^2 - \\ &\frac{\mu}{2} \left( 2 - \frac{\mu}{\mu_0} \right) \sum_{i=1}^t \| \xi(i) + \eta(i) \|^2. \end{aligned}$$

如果选择

$$0 < \mu \leq 2\mu_0 = \frac{2}{\lambda_{\max} [A^T A] \lambda_{\max} [BB^T] + \lambda_{\max} [C^T C] \lambda_{\max} [DD^T]},$$

则有

$$\sum_{i=1}^{\infty} \| \xi(i) + \eta(i) \|^2 < \infty.$$

根据级数收敛的必要条件,可得

$$\xi(t) + \eta(t) \rightarrow 0, \text{ 当 } t \rightarrow \infty,$$

或

$$A \tilde{X}(t-1) B + C \tilde{X}(t-1) D \rightarrow 0, \text{ 当 } t \rightarrow \infty.$$

根据引理4,当  $t \rightarrow \infty$  时,有  $\tilde{X}(t) \rightarrow 0$ . 定理5证毕.

同样,算法(42)或(43)称为单边迭代算法,它不能保证  $X_i(t)$  收敛到  $X$ ; 算法(44) — (46) 称为平衡迭代算法<sup>[21-22]</sup>,它可以简单写为

$$\begin{aligned} X(t) &= X(t-1) + \mu A^T [F - AX(t-1)B - \\ &CX(t-1)D] B^T / 2 + \mu C^T [F - \\ &AX(t-1)B - CX(t-1)D] D^T / 2. \end{aligned}$$

类似地,如果根据定理2的式(12),能够推出矩阵方程(38)的递阶最小二乘迭代算法如下<sup>[21-22]</sup>:

$$\begin{aligned} X(t) &= \frac{X_1(t) + X_2(t)}{2}, \\ X_1(t) &= X(t-1) + \mu (A^T A)^{-1} A^T [F - \\ &AX(t-1)B - CX(t-1)D] B^T (BB^T)^{-1}, \\ X_2(t) &= X(t-1) + \mu (C^T C)^{-1} C^T [F - \\ &AX(t-1)B - CX(t-1)D] D^T (DD^T)^{-1}, \\ &\mu \leq 4. \end{aligned}$$

(笔者也难以证明这个递阶最小二乘迭代算法的收敛性). 方程(38)可以推广为

$$AX_1 B + CX_2 D = F.$$

### 3.4 矩阵方程 $A_1 X B_1 + \dots + A_p X B_p = F$

本小节主要内容选自文献[21-22].

考虑一般矩阵方程(general matrix equation):

$$A_1 X B_1 + A_2 X B_2 + \dots + A_p X B_p = F, \quad (51)$$

其中  $A_j \in \mathbf{R}^{p \times m}$ ,  $B_j \in \mathbf{R}^{n \times q}$  和  $F \in \mathbf{R}^{m \times n}$  为给定常数矩阵,  $X \in \mathbf{R}^{m \times n}$  是未知矩阵.

引理5 如果  $\sum_{j=1}^p (B_j^T \otimes A_j)$  非奇异,则在最小二乘意义下,矩阵方程(51)有唯一解:

$$\text{col}[X] = \left[ \sum_{j=1}^p (B_j^T \otimes A_j) \right]^{-1} \text{col}[F],$$

而齐次方程  $A_1 X B_1 + A_2 X B_2 + \dots + A_p X B_p = 0$  有零解  $X = 0$ .

参照3.3的推导方法,能够写出计算方程(51)迭代解的递阶梯度迭代算法<sup>[21-22]</sup>:

$$X(t) = [X_1(t) + X_2(t) + \dots + X_p(t)] / p, \quad (52)$$

$$\begin{aligned} X_i(t) &= X(t-1) + \mu A_i^T \left[ F - \sum_{j=1}^p A_j X(t-1) B_j \right] B_i^T, \\ &i = 1, 2, \dots, p, \end{aligned} \quad (53)$$

$$0 < \mu \leq \frac{2}{\sum_{j=1}^p \lambda_{\max} [A_j^T A_j] \lambda_{\max} [B_j B_j^T]}, \text{ 或}$$

$$0 < \mu \leq \frac{2}{\sum_{j=1}^p \| A_j \|^2 \| B_j \|^2}. \quad (54)$$

定理6<sup>[21-22]</sup> 如果方程(38)有唯一解(或最小二乘解)  $X$ , 那么对任意有限的初值  $X(0)$ , 迭代算法(52) — (54)产生的迭代解  $X(t)$  收敛于  $X$ , 即  $X(t) \rightarrow X$ .

类似地,能够得到计算方程(51)迭代解的递阶最小二乘迭代算法<sup>[21-22]</sup>:

$$X(t) = [X_1(t) + X_2(t) + \dots + X_p(t)] / p, \quad (55)$$

$$\begin{aligned} X_i(t) &= X(t-1) + \mu (A_i^T A_i)^{-1} A_i^T \left[ F - \right. \\ &\left. \sum_{j=1}^p A_j X(t-1) B_j \right] B_i^T (B_i B_i^T)^{-1}, \end{aligned} \quad (56)$$

$$0 < \mu \leq 2p, \quad i = 1, 2, \dots, p. \quad (57)$$

一般矩阵方程,包括西尔维斯特矩阵方程的递阶最小二乘迭代算法的收敛性证明是极其困难的,笔者还没有找到办法,还有待读者深入研究、进一步证明.

一般矩阵方程(51)可以推广为

$$A_1 X_1 B_1 + A_2 X_2 B_2 + \dots + A_p X_p B_p = F,$$

求解其的递阶梯度迭代算法和递阶最小二乘迭代算法的收敛性都是需要进一步研究的.

## 4 耦合矩阵方程

应用递阶辨识原理先研究耦合西尔维斯特矩阵

方程的递阶梯度迭代算法, 然后通过引入块矩阵内积 (block matrix inner product), 简称 star 积或\* 积, 把梯度迭代算法推广用于一般耦合矩阵方程的求解. 本节主要内容选自文献 [20].

#### 4.1 耦合西尔维斯特矩阵方程

考虑下列耦合西尔维斯特矩阵方程 (coupled Sylvester matrix equation):

$$\begin{cases} AX + YB = C, \\ DX + YE = F. \end{cases} \quad (58)$$

这里  $A, D \in \mathbf{R}^{m \times m}$ ,  $B, E \in \mathbf{R}^{n \times n}$ ,  $C, F \in \mathbf{R}^{m \times n}$  是给定常数矩阵,  $X, Y \in \mathbf{R}^{m \times n}$  是要求解的未知矩阵.

下列引理是明显的.

引理 6 矩阵方程 (58) 有唯一解的充分必要条件是矩阵

$$\Phi = \begin{bmatrix} I_n \otimes A & B^T \otimes I_m \\ I_n \otimes D & E^T \otimes I_m \end{bmatrix} \in \mathbf{R}^{(2mn) \times (2mn)}$$

非奇异, 且这个唯一解为

$$\text{col}[X, Y] = \Phi^{-1} \text{col}[C, F], \quad (59)$$

对应的齐次方程组 ( $AX + YB = 0$ ,  $DX + YE = 0$ ) 有唯一解  $X = Y = 0$ .

讨论耦合矩阵方程 (coupled matrix equation) 的迭代解, 方法仍是根据递阶辨识原理, 唯一的差别是如何处理两个方程的分解问题, 即如何把系统 (58) 分解为两个子系统. 这里把  $X$  和  $Y$  当作系统待辨识的参数矩阵, 然后利用推论 1, 分别导出每个子系统参数矩阵的迭代解, 具体步骤如下.

定义两个矩阵:

$$b_1 := \begin{bmatrix} C - YB \\ F - YE \end{bmatrix}, \quad (60)$$

$$b_2 := [C - AX, F - DX]. \quad (61)$$

从式 (58), 可以得到两个虚拟子系统:

$$S_5: \begin{bmatrix} A \\ D \end{bmatrix} X = b_1,$$

$$S_6: Y[B, E] = b_2.$$

令  $X(t)$  和  $Y(t)$  分别为系统  $S_5$  和  $S_6$  中  $X$  和  $Y$  的估计或迭代解. 应用推论 1 于系统  $S_5$  和  $S_6$ , 能够导出下列迭代方程:

$$X(t) = X(t-1) + \mu \begin{bmatrix} A \\ D \end{bmatrix}^T \left\{ b_1 - \begin{bmatrix} A \\ D \end{bmatrix} X(t-1) \right\}, \quad (62)$$

$$Y(t) = Y(t-1) + \mu \{ b_2 - Y(t-1)[B, E] \} [B, E]^T. \quad (63)$$

这里  $\mu$  为迭代步长或收敛因子, 可以取为

$$0 < \mu \leq \frac{2}{\|A\|^2 + \|B\|^2 + \|D\|^2 + \|E\|^2}.$$

将式 (60) 代入式 (62), 式 (61) 代入式 (63) 得到

$$X(t) = X(t-1) + \mu \begin{bmatrix} A \\ D \end{bmatrix}^T \left\{ \begin{bmatrix} C - YB \\ F - YE \end{bmatrix} - \begin{bmatrix} A \\ D \end{bmatrix} X(t-1) \right\} =$$

$$X(t-1) + \mu \begin{bmatrix} A \\ D \end{bmatrix}^T \begin{bmatrix} C - YB - AX(t-1) \\ F - YE - DX(t-1) \end{bmatrix}, \quad (64)$$

$$Y(t) = Y(t-1) + \mu \{ [C - AX, F - DX] -$$

$$Y(t-1)[B, E] \} [B, E]^T =$$

$$Y(t-1) + \mu [C - AX - Y(t-1)B,$$

$$F - DX - Y(t-1)E] [B, E]^T. \quad (65)$$

借助于递阶辨识原理, 式 (64) 和 (65) 中的未知量  $Y$  和  $X$  分别用它们在  $t-1$  时刻的估计  $Y(t-1)$  和  $X(t-1)$  代替, 就得到计算耦合矩阵方程 (58) 迭代解的递阶梯度迭代算法:

$$X(t) = X(t-1) +$$

$$\mu \begin{bmatrix} A \\ D \end{bmatrix}^T \begin{bmatrix} C - AX(t-1) - Y(t-1)B \\ F - DX(t-1) - Y(t-1)E \end{bmatrix}, \quad (66)$$

$$Y(t) = Y(t-1) + \mu [C - AX(t-1) - Y(t-1)B,$$

$$F - DX(t-1) - Y(t-1)E] [B, E]^T, \quad (67)$$

$$0 < \mu \leq \frac{2}{\|A\|^2 + \|B\|^2 + \|D\|^2 + \|E\|^2}. \quad (68)$$

这个迭代算法的初值可以选择为  $X(0) = Y(0) = 10^{-6} \times \mathbf{1}_{m \times n}$ .

定理 7<sup>[20]</sup> 如果耦合矩阵方程 (58) 在最小二乘意义下有唯一解  $X$  和  $Y$ , 那么对任意有限初值, 算法 (66) — (67) 给出的迭代解  $X(t)$  和  $Y(t)$  分别收敛于  $X$  和  $Y$ , 即

$$\lim_{t \rightarrow \infty} X(t) = X, \quad \lim_{t \rightarrow \infty} Y(t) = Y.$$

收敛因子也可以按下式选择

$$0 < \mu \leq \frac{2}{\lambda_{\max}[A^T A] + \lambda_{\max}[D^T D] + \lambda_{\max}[B B^T] + \lambda_{\max}[E E^T]}$$

#### 4.2 一般耦合矩阵方程

丁锋等<sup>[20, 23]</sup> 首次通过定义块矩阵内积 (star 积或\* 积), 广义化耦合西尔维斯特矩阵方程, 进一步导出下列一般耦合矩阵方程 (general coupled matrix equations) 的迭代解:

$$\begin{cases} A_{11} X_1 B_{11} + A_{12} X_2 B_{12} + \cdots + A_{1p} X_p B_{1p} = F_1, \\ A_{21} X_1 B_{21} + A_{22} X_2 B_{22} + \cdots + A_{2p} X_p B_{2p} = F_2, \\ \cdots \\ A_{p1} X_1 B_{p1} + A_{p2} X_2 B_{p2} + \cdots + A_{pp} X_p B_{pp} = F_p. \end{cases} \quad (69)$$

这里  $A_{ij} \in \mathbf{R}^{m \times m}$ ,  $B_{ij} \in \mathbf{R}^{n \times n}$  和  $F_i \in \mathbf{R}^{m \times n}$  是给定常数矩阵,  $X_i \in \mathbf{R}^{m \times n}$  是未知矩阵.

为了更简洁表示要讨论的迭代算法, 定义\* 积, 即块矩阵内积, 用符号\* 表示<sup>[22-23]</sup>. 它不同于 Had-

amard 积(内积),也不同于一般矩阵乘法.下列矩阵  $X, Y, S_A$  等的各元都是由矩阵构成的.当这些矩阵元为标量时,star 积退化为简单的内积(Hadamard 积).定义

$$X := \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} \in \mathbf{R}^{(mp) \times n}, Y := \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_p \end{bmatrix} \in \mathbf{R}^{(mp) \times m}, X_i, Y_i^T \in \mathbf{R}^{m \times n},$$

$$S_A := \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{bmatrix}, S_B := \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ B_{p1} & B_{p2} & \cdots & B_{pp} \end{bmatrix},$$

$$S_{B^T} := \begin{bmatrix} B_{11}^T & B_{12}^T & \cdots & B_{1p}^T \\ B_{21}^T & B_{22}^T & \cdots & B_{2p}^T \\ \vdots & \vdots & \ddots & \vdots \\ B_{p1}^T & B_{p2}^T & \cdots & B_{pp}^T \end{bmatrix},$$

$$S_p := \begin{bmatrix} B_{11}^T \otimes A_{11} & B_{12}^T \otimes A_{12} & \cdots & B_{1p}^T \otimes A_{1p} \\ B_{21}^T \otimes A_{21} & B_{22}^T \otimes A_{22} & \cdots & B_{2p}^T \otimes A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ B_{p1}^T \otimes A_{p1} & B_{p2}^T \otimes A_{p2} & \cdots & B_{pp}^T \otimes A_{pp} \end{bmatrix}.$$

这些矩阵的\*积定义为

$$X^* Y = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} * \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_p \end{bmatrix} = \begin{bmatrix} X_1 Y_1 \\ X_2 Y_2 \\ \vdots \\ X_p Y_p \end{bmatrix},$$

$$S_A^* X = \begin{bmatrix} A_{11} X_1 & A_{12} X_2 & \cdots & A_{1p} X_p \\ A_{21} X_1 & A_{22} X_2 & \cdots & A_{2p} X_p \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} X_1 & A_{p2} X_2 & \cdots & A_{pp} X_p \end{bmatrix},$$

$$X^* S_B = \begin{bmatrix} X_1 B_{11} & X_1 B_{12} & \cdots & X_1 B_{1p} \\ X_2 B_{21} & X_2 B_{22} & \cdots & X_2 B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_p B_{p1} & X_p B_{p2} & \cdots & X_p B_{pp} \end{bmatrix},$$

$$S_A^* S_B = \begin{bmatrix} A_{11} B_{11} & A_{12} B_{12} & \cdots & A_{1p} B_{1p} \\ A_{21} B_{21} & A_{22} B_{22} & \cdots & A_{2p} B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} B_{p1} & A_{p2} B_{p2} & \cdots & A_{pp} B_{pp} \end{bmatrix}.$$

在上述定义中,假设矩阵和被乘矩阵维数是兼容的,即维数一致可相乘.我们定义块矩阵 star Kronecker 积(用符号  $\otimes$  表示)如下:

$$S_{B^T} \otimes S_A = S_p.$$

对于 Hadamard 积,用符号  $\circ$  表示,有  $X \circ Y = Y \circ X$ ,但是  $X \circ S_A$  没有意义.star 积优于矩阵乘法的计算,即  $AB^* C = A(B^* C) \neq (AB)^* C$ ,矩阵和被乘矩阵不必维数相同.一般地  $A^* B \neq B^* A, A^* B^* C = (A^* B) C \neq A^* (B^* C)$ .

令  $I_{mp \times m} = [I_m, I_m, \dots, I_m]^T \in \mathbf{R}^{(mp) \times m}$ .那么 star 积有下列性质:

$$I_{mp \times m}^T X^* Y = [X_1, X_2, \dots, X_p] Y = \sum_{i=1}^p X_i Y_i,$$

$$\text{tr} \left\{ X_i^T \begin{bmatrix} A_{1i} \\ A_{2i} \\ \vdots \\ A_{pi} \end{bmatrix} \begin{bmatrix} \tilde{F}_1 \\ \tilde{F}_2 \\ \vdots \\ \tilde{F}_p \end{bmatrix} * \begin{bmatrix} B_{1i}^T \\ B_{2i}^T \\ \vdots \\ B_{pi}^T \end{bmatrix} \right\} = \text{tr} \left\{ \begin{bmatrix} A_{1i} X_i B_{1i} \\ A_{2i} X_i B_{2i} \\ \vdots \\ A_{pi} X_i B_{pi} \end{bmatrix} \begin{bmatrix} \tilde{F}_1 \\ \tilde{F}_2 \\ \vdots \\ \tilde{F}_p \end{bmatrix} \right\},$$

$$\left\| \begin{bmatrix} A_{1i} \\ A_{2i} \\ \vdots \\ A_{pi} \end{bmatrix} \begin{bmatrix} \tilde{F}_1 \\ \tilde{F}_2 \\ \vdots \\ \tilde{F}_p \end{bmatrix} * \begin{bmatrix} B_{1i}^T \\ B_{2i}^T \\ \vdots \\ B_{pi}^T \end{bmatrix} \right\|^2 \leq \sum_{j=1}^p \|A_{ji}\|^2 \|B_{ji}\|^2 \left\| \begin{bmatrix} \tilde{F}_1 \\ \tilde{F}_2 \\ \vdots \\ \tilde{F}_p \end{bmatrix} \right\|^2.$$

引理 7<sup>[20, 23]</sup> 如果矩阵  $S_p$  非奇异,那么矩阵方程(69)有唯一解

$$\text{col} [X_1, X_2, \dots, X_p] = S_p^{-1} \text{col} [F_1, F_2, \dots, F_p];$$

如果  $F_i = 0 (i = 1, 2, \dots, p)$ ,那么对应的齐次方程(69)有唯一零解  $X_i = 0 (i = 1, 2, \dots, p)$ .

为了建立一般耦合矩阵(69)的递阶梯度迭代算法,需要把耦合西尔维斯特矩阵方程(58)广义化成下列一般形式:

$$\begin{cases} AXI_B + I_A YB = C, \\ DXI_E + I_D YE = F, \end{cases}$$

其递阶梯度迭代解可以表示为

$$X(t) = X(t-1) + \mu \begin{bmatrix} A \\ D \end{bmatrix}^T \left\{ \begin{bmatrix} C - AX(t-1)I_B - I_A Y(t-1)B \\ F - DX(t-1)I_E - I_D Y(t-1)E \end{bmatrix} * \begin{bmatrix} I_B & I_E \end{bmatrix}^T \right\},$$

$$Y(t) = Y(t-1) + \mu \begin{bmatrix} I_A \\ I_D \end{bmatrix}^T \left\{ \begin{bmatrix} C - AX(t-1)I_B - I_A Y(t-1)B \\ F - DX(t-1)I_E - I_D Y(t-1)E \end{bmatrix} * \begin{bmatrix} B & E \end{bmatrix}^T \right\}.$$

如果  $I_A, I_B, I_D$  和  $I_E$  均为适当维数的单位阵,那么这个迭代算法完全等价于(66)——(67).

令  $X_i(t)$  为  $X_i$  的估计或迭代解.有了上述推广,能够得到计算一般耦合矩阵方程(69)迭代解  $X_i (i = 1, 2, \dots, p)$  的递阶梯度迭代算法<sup>[20]</sup>:

$$X_i(t) = X_i(t-1) +$$

$$\mu \begin{bmatrix} A_{1i} \\ A_{2i} \\ \vdots \\ A_{pi} \end{bmatrix}^T \begin{bmatrix} F_1 - \sum_{j=1}^p A_{1j} X_j(t-1) B_{1j} \\ F_2 - \sum_{j=1}^p A_{2j} X_j(t-1) B_{2j} \\ \vdots \\ F_p - \sum_{j=1}^p A_{pj} X_j(t-1) B_{pj} \end{bmatrix} * [B_{1i} \ B_{2i} \ ; \ \dots \ B_{pi}]^T, \quad (70)$$

$$\mu \leq 2 \left( \sum_{i=1}^p \sum_{j=1}^p \|A_{ij} B_{ij}\|^2 \right)^{-1}. \quad (71)$$

定理 8 如果矩阵方程(69)有唯一解  $X_i, i=1, 2, \dots, p$ , 那么对任意有限初值, 算法(70) — (71) 给出的迭代解  $X_i(t)$  收敛于  $X_i$ , 即

$$\lim_{t \rightarrow \infty} X_i(t) = X_i, \quad i=1, 2, \dots, p.$$

如果令

$$X(t) := \begin{bmatrix} X_1(t) \\ X_2(t) \\ \vdots \\ X_p(t) \end{bmatrix} \in \mathbf{R}^{(mp) \times n}, \quad F := \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_p \end{bmatrix} \in \mathbf{R}^{(mp) \times n},$$

那么式(69)可以简单写为

$$S_A * X * S_B I_{np \times n} = F.$$

使用 star 积性质, 算法(70)可以写成下列更紧凑形式:

$$X(t) = X(t-1) +$$

$$\mu S_A^T \begin{bmatrix} F_1 - \sum_{j=1}^p A_{1j} X_j(t-1) B_{1j} \\ F_2 - \sum_{j=1}^p A_{2j} X_j(t-1) B_{2j} \\ \vdots \\ F_p - \sum_{j=1}^p A_{pj} X_j(t-1) B_{pj} \end{bmatrix} * S_B^T =$$

$$X(t-1) + \mu S_A^T [F - S_A * X(t-1) * S_B I_{np \times n}] * S_B^T.$$

耦合矩阵方程的最小二乘迭代解算法可参见文献[23, 36]

### 4.3 其他矩阵方程

假设下面矩阵方程的最小二乘解存在, 可以借助于递阶辨识原理研究其递阶梯度迭代算法和递阶最小二乘迭代算法及其收敛性.

1) 研究下列矩阵方程的迭代求解算法:

$$A_i X = b_i, \quad A_i \in \mathbf{R}^{m_i \times n}, \quad b_i \in \mathbf{R}^{m_i}, \quad i=1, 2, \dots, N.$$

2) 求解下列矩阵方程, 并推导其梯度迭代算法和最小二乘迭代算法:

$$\textcircled{1} A_i x = b, \quad i=1, 2, \dots, N;$$

$$\textcircled{2} Ax = b_i, \quad i=1, 2, \dots, N;$$

$$\textcircled{3} \begin{cases} A_i x = b, & i=1, 2, \dots, N, \\ Ax = b_j, & j=1, 2, \dots, N. \end{cases}$$

3) 假设最小二乘解存在, 推导下列矩阵方程的梯度迭代算法和最小二乘迭代算法:

$$A_1 X_1 B_1 + A_2 X_2 B_2 + \dots + A_N X_N B_N = F.$$

4) 求解下列矩阵方程, 并推导其梯度迭代算法和最小二乘迭代算法:

$$\textcircled{1} A_i X B = F, \quad i=1, 2, \dots, N;$$

$$\textcircled{2} A X B_i = F_i, \quad i=1, 2, \dots, N;$$

$$\textcircled{3} \begin{cases} A_i X B = F, & i=1, 2, \dots, N, \\ A X B_j = F_j, & j=1, 2, \dots, N. \end{cases}$$

5) 假设最小二乘解存在, 推导下列矩阵方程的梯度迭代算法和最小二乘迭代算法:

$$A_1 X_1 B_1 + A_2 X_2 B_2 + \dots + A_N X_N B_N = F.$$

6) 假设最小二乘解存在, 借鉴矩阵方程  $A X B = F$  的迭代求解算法, 推导下列矩阵方程的梯度迭代算法和最小二乘迭代算法<sup>[26]</sup>:

$$\begin{cases} A_1 X B_1 = F_1, \\ A_2 X B_2 = F_2, \\ \vdots \\ A_N X B_N = F_N. \end{cases}$$

7) 求解下列矩阵方程, 并推导其梯度迭代算法和最小二乘迭代算法:

$$\begin{cases} A X + X B = C, \\ D X + X E = F \end{cases}$$

和

$$\begin{cases} A_1 X + X B_1 = F_1, \\ A_2 X + X B_2 = F_2, \\ \vdots \\ A_N X + X B_N = F_N. \end{cases}$$

8) 假设最小二乘解存在, 借鉴耦合西尔维斯特耦合矩阵方程

$$\begin{cases} A X + Y B = C, \\ D X + Y E = F \end{cases}$$

的迭代求解算法, 推导下列矩阵方程的梯度迭代算法和最小二乘迭代算法:

$$\begin{cases} A_1 X + Y B_1 = F_1, \\ A_2 X + Y B_2 = F_2, \\ \vdots \\ A_N X + Y B_N = F_N. \end{cases}$$

9) 假设最小二乘解存在, 推导下列耦合矩阵方程的梯度迭代算法和最小二乘迭代算法:

$$\begin{cases} A_1XB_1 + A_2XB_2 = F_1, \\ C_1XD_1 + C_2XD_2 = F_2 \end{cases}$$

和

$$\begin{cases} A_1XB_1 + A_2XB_2 + \cdots + A_qXB_q = F_1, \\ C_1XD_1 + C_2XD_2 + \cdots + C_qXD_q = F_2. \end{cases}$$

10) 假设最小二乘解存在, 推导下列耦合矩阵方程的梯度迭代算法和最小二乘迭代算法:

$$\begin{cases} A_{11}XB_{11} + A_{12}XB_{12} + \cdots + A_{1q}XB_{1q} = F_1, \\ A_{21}XB_{21} + A_{22}XB_{22} + \cdots + A_{2q}XB_{2q} = F_2, \\ \vdots \\ A_{p1}XB_{p1} + A_{p2}XB_{p2} + \cdots + A_{pq}XB_{pq} = F_p. \end{cases}$$

11) 假设最小二乘解存在, 推导下列耦合矩阵方程的梯度迭代算法和最小二乘迭代算法:

$$\begin{cases} A_{11}X_1B_{11} + A_{12}X_2B_{12} + \cdots + A_{1q}X_qB_{1q} = F_1, \\ A_{21}X_1B_{21} + A_{22}X_2B_{22} + \cdots + A_{2q}X_qB_{2q} = F_2, \\ \vdots \\ A_{p1}X_1B_{p1} + A_{p2}X_2B_{p2} + \cdots + A_{pq}X_qB_{pq} = F_p. \end{cases}$$

12) 假设最小二乘解存在, 推导下列矩阵方程的梯度迭代算法和最小二乘迭代算法<sup>[25 27]</sup>:

$$\begin{aligned} \textcircled{1} \quad & AX + X^T B = F. \\ \textcircled{2} \quad & AXB + CX^T D = F. \\ \textcircled{3} \quad & A_1XB_1 + \cdots + A_pXB_p + C_1X^T D_1 + \cdots + C_qX^T D_q = F. \end{aligned}$$

13) 假设最小二乘解存在, 推导下列耦合矩阵方程的梯度迭代算法和最小二乘迭代算法:

$$\begin{cases} A_1XB_1 + C_1X^T D_1 = F_1, \\ A_2XB_2 + C_2X^T D_2 = F_2. \end{cases}$$

14) 假设最小二乘解存在, 推导下列耦合矩阵方程的梯度迭代算法和最小二乘迭代算法:

$$\begin{cases} A_{11}XB_{11} + \cdots + A_{1q}XB_{1q} + C_{11}X^T D_{11} + \cdots + C_{1q}X^T D_{1q} = F_1, \\ A_{21}XB_{21} + \cdots + A_{2q}XB_{2q} + C_{21}X^T D_{21} + \cdots + C_{2q}X^T D_{2q} = F_2, \\ \vdots \\ A_{p1}XB_{p1} + \cdots + A_{pq}XB_{pq} + C_{p1}X^T D_{p1} + \cdots + C_{pq}X^T D_{pq} = F_p. \end{cases}$$

15) 假设最小二乘解存在, 推导下列一般耦合矩阵方程的梯度迭代算法和最小二乘迭代算法:

$$\begin{cases} A_{11}X_1B_{11} + \cdots + A_{1q}X_qB_{1q} + C_{11}X_1^T D_{11} + \cdots + C_{1q}X_q^T D_{1q} = F_1, \\ A_{21}X_1B_{21} + \cdots + A_{2q}X_qB_{2q} + C_{21}X_1^T D_{21} + \cdots + C_{2q}X_q^T D_{2q} = F_2, \\ \vdots \\ A_{p1}X_1B_{p1} + \cdots + A_{pq}X_qB_{pq} + C_{p1}X_1^T D_{p1} + \cdots + C_{pq}X_q^T D_{pq} = F_p. \end{cases}$$

16) 上述一些矩阵方程引入了矩阵转置, 当然也可以引入共轭矩阵, 共轭转置矩阵等, 导出一系列矩阵方程, 这里仅列出 3 例:

$$\textcircled{1} \quad AX + \bar{X}B = F;$$

$$\textcircled{2} \quad AXB + C \bar{X}D = F;$$

$$\textcircled{3} \quad \begin{cases} A_1XB_1 + C_1 \bar{X}D_1 = F_1, \\ A_2XB_2 + C_2 \bar{X}D_2 = F_2. \end{cases}$$

读者可以推广到多个未知矩阵情况.

17) 借助于矩阵分块和递阶辨识原理, 也可研究上述一些矩阵方程的迭代算法. 例如, 对于方程

$$Ax = b, \quad A \in \mathbf{R}^{m \times n}, \quad x \in \mathbf{R}^n, \quad b \in \mathbf{R}^m,$$

假设  $m \geq n$ ,  $A$  满秩. 把  $A$  分成  $2 \times 2$  块,

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad A_{ij} \in \mathbf{R}^{m_i \times n_j},$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

$x_i \in \mathbf{R}^{n_i}$ ,  $b_i \in \mathbf{R}^{m_i}$ ,  $m_1 + m_2 = m$ ,  $n_1 + n_2 = n$ , 得到

$$\begin{cases} A_{11}x_1 + A_{12}x_2 = b_1, \\ A_{21}x_1 + A_{22}x_2 = b_2. \end{cases}$$

应用递阶辨识原理, 推导相应的迭代解算法.

18) 假设矩阵方程

$$AXB = F, \quad A \in \mathbf{R}^{m \times n}, \quad B \in \mathbf{R}^{n \times r}, \quad F \in \mathbf{R}^{m \times r}$$

的最小二乘解存在, 这个解可以表示为

$$X = (A^T A)^{-1} A^T F B^T (B B^T)^{-1} \in \mathbf{R}^{n \times n}.$$

一类新问题的提法: 通常这个解不是对称的, 如何求这个矩阵方程的对称解, 即求解一个满足下列优化问题的解:

$$\min_{X=X^T} \|F - AXB\|^2.$$

再如, 假设对于某个特殊矩阵  $J$  在满足  $J^T X J = X$  所有解  $X$  的集合中, 找出一个解满足  $AXB = F$ , 即求下列条件优化问题的解:

$$\min_{X=J^T X J} \|F - AXB\|^2.$$

这种思想可派生出很多值得研究的问题, 可用于本文所列出的其他矩阵方程的解.

19) 众所周知, 高斯消元方法 (Gauss elimination) 可以用于求解矩阵方程  $AX = B$ , 试将高斯消元法加以推广用于求解矩阵方程  $(A \otimes B)X = F$ , 并考虑  $A$  或  $B$  为对角阵或单位阵情形.

20) 研究下列矩阵方程的迭代解:

$$\textcircled{1} \quad X + AX^{-1}B = F,$$

$$\textcircled{2} \quad AXB + CX^{-1}D = F.$$

这些矩阵方程还可进行横向和纵向推广, 如

$$\textcircled{1} \quad A_1XB_1 + \cdots + A_pXB_p + C_1X^{-1}D_1 + \cdots + C_qX^{-1}D_q = F.$$

$$\textcircled{2} A_1 X_1 B_1 + \dots + A_p X_p B_p + C_1 X_1^{-1} D_1 + \dots + C_q X_q^{-1} D_q = F.$$

21) 研究下列矩阵方程的迭代解:

$$\textcircled{1} X + AX^{-1}B = F.$$

$$\textcircled{2} AXB + CX^{-1}D = F.$$

## 5 方程误差模型的两阶段递推最小二乘法

本节的两阶段递推最小二乘法是下节递阶最小二乘法的一个特殊情形. 两阶段辨识方法也可称为交互估计方法. 最近, Duan 等<sup>[37]</sup>研究了输出误差模型的两阶段递推最小二乘参数估计算法; Yao 等<sup>[38]</sup>讨论了受控自回归滑动平均模型(CARMA)的两阶段最小二乘迭代辨识算法; Xiao 等<sup>[39]</sup>提出了基于残差的交互最小二乘算法; Wang 等<sup>[40]</sup>提出了基于残差的受控滑动平均模型的交互随机梯度算法.

### 5.1 系统描述与问题构成

考虑受控自回归模型(CAR)描述的系统:

$$A(z)y(t) = B(z)u(t) + v(t), \quad (72)$$

其中 $\{u(t)\}$ 和 $\{y(t)\}$ 分别是系统的输入和输出序列,  $\{v(t)\}$ 是零均值方差为 $\sigma^2$ 的随机白噪声序列,  $z^{-1}$ 为单位后移算子 $[z^{-1}y(t) = y(t-1)$   $zy(t) = y(t+1)$ ]  $A(z)$ 和 $B(z)$ 是单位后移算子 $z^{-1}$ 的常系数时不变多项式:

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{n_a} z^{-n_a},$$

$$B(z) = b_1 z^{-1} + b_2 z^{-2} + \dots + b_{n_b} z^{-n_b}.$$

设阶次 $n_a$ 和 $n_b$ 已知, 记 $n = n_a + n_b$ , 且 $t \leq 0$ 时,  $y(t) = 0$   $\mu(t) = 0$   $p(t) = 0$ .

下面基于分解技术推导 CAR 系统的两阶段递推最小二乘辨识算法. 分别定义 AR 部分参数向量 $\theta_a$ 和 MA 部分参数向量 $\theta_b$ 如下:

$$\theta_a = [a_1 \ a_2 \ \dots \ a_{n_a}]^T \in \mathbb{R}^{n_a},$$

$$\theta_b = [b_1 \ b_2 \ \dots \ b_{n_b}]^T \in \mathbb{R}^{n_b}.$$

信息向量

$$\varphi(t) = \begin{bmatrix} \varphi_a(t) \\ \varphi_b(t) \end{bmatrix} \in \mathbb{R}^n,$$

$$\varphi_a(t) = [-y(t-1) \ -y(t-2) \ \dots \ -y(t-n_a)]^T \in \mathbb{R}^{n_a},$$

$$\varphi_b(t) = [u(t-1) \ \mu(t-2) \ \dots \ \mu(t-n_b)]^T \in \mathbb{R}^{n_b}.$$

参数向量 $\theta = \begin{bmatrix} \theta_a \\ \theta_b \end{bmatrix} \in \mathbb{R}^n$ 包含了系统所有参数.

式(72)可以写为

$$y(t) = [1 - A(z)]y(t) + B(z)u(t) + v(t) =$$

$$\begin{aligned} & (-a_1 z^{-1} - a_2 z^{-2} - \dots - a_{n_a} z^{-n_a})y(t) + \\ & (b_1 z^{-1} + b_2 z^{-2} + \dots + b_{n_b} z^{-n_b})u(t) + v(t) = \\ & -a_1 y(t-1) - a_2 y(t-2) - \dots - a_{n_a} y(t-n_a) + \\ & b_1 u(t-1) + b_2 u(t-2) + \dots + b_{n_b} u(t-n_b) + \\ & v(t) = \varphi_a^T(t)\theta_a + \varphi_b^T(t)\theta_b + v(t) = \\ & \varphi^T(t)\theta + v(t). \end{aligned} \quad (73)$$

### 5.2 两阶段递推辨识算法

两阶段递推最小二乘辨识方法的基本思想是: 将辨识模型(73)分解为两个子系统, 分别辨识每个子系统的参数向量. 定义两个虚拟输出变量:

$$y_1(t) = y(t) - \varphi_b^T(t)\theta_b, \quad (74)$$

$$y_2(t) = y(t) - \varphi_a^T(t)\theta_a. \quad (75)$$

系统(73)可以分解为下列两个虚拟子系统(或称子辨识模型):

$$y_1(t) = \varphi_a^T(t)\theta_a + v(t),$$

$$y_2(t) = \varphi_b^T(t)\theta_b + v(t).$$

这两个子系统分别包含了 AR 部分参数向量 $\theta_a$ 和 MA 部分参数向量 $\theta_b$ . 定义两个准则函数,

$$J_1(\theta_a) = \sum_{j=1}^t [y_1(j) - \varphi_a^T(j)\theta_a]^2,$$

$$J_2(\theta_b) = \sum_{j=1}^t [y_2(j) - \varphi_b^T(j)\theta_b]^2.$$

令 $J_1(\theta_a)$ 和 $J_2(\theta_b)$ 分别关于参数向量 $\theta_a$ 和 $\theta_b$ 的偏导数为零, 得到

$$\frac{\partial J_1(\theta_a)}{\partial \theta_a} = -2\varphi_a(j) \sum_{j=1}^t [y_1(j) - \varphi_a^T(j)\theta_a] = 0,$$

$$\frac{\partial J_2(\theta_b)}{\partial \theta_b} = -2\varphi_b(j) \sum_{j=1}^t [y_2(j) - \varphi_b^T(j)\theta_b] = 0.$$

令 $\hat{\theta}(t) = \begin{bmatrix} \hat{\theta}_a(t) \\ \hat{\theta}_b(t) \end{bmatrix} \in \mathbb{R}^n$ 是 $\theta = \begin{bmatrix} \theta_a \\ \theta_b \end{bmatrix} \in \mathbb{R}^n$ 在时刻 $t$

的估计. 由上两式可以得到下列递推最小二乘算法:

$$\hat{\theta}_a(t) = \hat{\theta}_a(t-1) + L_a(t) [y_1(t) - \varphi_a^T(t)\hat{\theta}_a(t-1)], \quad (76)$$

$$L_a(t) = P_a(t-1)\varphi_a(t) [1 + \varphi_a^T(t)P_a(t-1)\varphi_a(t)]^{-1}, \quad (77)$$

$$P_a(t) = [I - L_a(t)\varphi_a^T(t)]P_a(t-1), \quad P_a(0) = p_0 I, \quad (78)$$

$$\hat{\theta}_b(t) = \hat{\theta}_b(t-1) + L_b(t) [y_2(t) - \varphi_b^T(t)\hat{\theta}_b(t-1)], \quad (79)$$

$$L_b(t) = P_b(t-1)\varphi_b(t) [1 + \varphi_b^T(t)P_b(t-1)\varphi_b(t)]^{-1}, \quad (80)$$

$$P_b(t) = [I - L_b(t)\varphi_b^T(t)]P_b(t-1), \quad P_b(0) = p_0 I. \quad (81)$$

将式(74)和(75)分别代入式(76)和(79)得到

$$\begin{aligned} \hat{\theta}_a(t) = & \hat{\theta}_a(t-1) + L_a(t) [y(t) - \\ & \varphi_b^T(t)\theta_b - \varphi_a^T(t)\hat{\theta}_a(t-1)], \end{aligned} \quad (82)$$

$$\begin{aligned} \hat{\theta}_b(t) = & \hat{\theta}_b(t-1) + L_b(t) [y(t) - \\ & \varphi_a^T(t)\theta_a - \varphi_b^T(t)\hat{\theta}_b(t-1)]. \end{aligned} \quad (83)$$

由于式(82)和(83)右边分别包含了未知参数向量 $\theta_b$

和  $\theta_a$  使得这个递推算法无法实现. 解决的办法是: 式 (82) 中未知  $\theta_b$  和式 (83) 中未知  $\theta_a$  分别用其在  $(t-1)$  时刻的估计  $\hat{\theta}_b(t-1)$  和  $\hat{\theta}_a(t-1)$  代替, 得到

$$\hat{\theta}_a(t) = \hat{\theta}_a(t-1) + L_a(t) [y(t) - \varphi_a^T(t) \hat{\theta}_b(t-1) - \varphi_a^T(t) \hat{\theta}_a(t-1)] = \hat{\theta}_a(t-1) + L_a(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)], \quad (84)$$

$$\hat{\theta}_b(t) = \hat{\theta}_b(t-1) + L_b(t) [y(t) - \varphi_b^T(t) \hat{\theta}_a(t-1) - \varphi_b^T(t) \hat{\theta}_b(t-1)] = \hat{\theta}_b(t-1) + L_b(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)]. \quad (85)$$

联立式 (84), (77) — (78), 以及式 (85), (80) — (81) 给出 CAR 模型的两阶段递推最小二乘算法 (Two-Stage Recursive Least Squares identification algorithm, TS-RLS 算法):

$$\hat{\theta}(t) = \hat{\theta}(t-1) + L_a(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)], \quad (86)$$

$$L_a(t) = P_a(t-1) \varphi_a(t) [1 + \varphi_a^T(t) P_a(t-1) \varphi_a(t)]^{-1}, \quad (87)$$

$$P_a(t) = [I - L_a(t) \varphi_a^T(t)] P_a(t-1), \quad P_a(0) = p_0 I, \quad (88)$$

$$\varphi_a(t) = [-y(t-1), -y(t-2), \dots, -y(t-n_d)]^T, \quad (89)$$

$$\hat{\theta}_b(t) = \hat{\theta}_b(t-1) + L_b(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)], \quad (90)$$

$$L_b(t) = P_b(t-1) \varphi_b(t) [1 + \varphi_b^T(t) P_b(t-1) \varphi_b(t)]^{-1}, \quad (91)$$

$$P_b(t) = [I - L_b(t) \varphi_b^T(t)] P_b(t-1), \quad P_b(0) = p_0 I, \quad (92)$$

$$\varphi_b(t) = [u(t-1), u(t-2), \dots, u(t-n_b)]^T. \quad (93)$$

$L_a(t) \in \mathbf{R}^{n_a}$  和  $L_b(t) \in \mathbf{R}^{n_b}$  是两个增益向量,  $P_a(t) \in \mathbf{R}^{n_a \times n_a}$  和  $P_b(t) \in \mathbf{R}^{n_b \times n_b}$  是两个协方差阵.

两阶段递推最小二乘算法 (86) — (93) 计算参数估计向量  $\hat{\theta}_a(t)$  和  $\hat{\theta}_b(t)$  的步骤如下.

1) 令  $t=1$ , 置初值  $P_a(0) = p_0 I$ ,  $P_b(0) = p_0 I$ ,  $\hat{\theta}_a(0) = \mathbf{1}_{n_a}/p_0$ ,  $\hat{\theta}_b(0) = \mathbf{1}_{n_b}/p_0$ ,  $u(i) = 0$ ,  $y(i) = 0$  ( $i \leq 0$ ).  $p_0$  是一个大常数, 如取  $p_0 = 10^6$ .

2) 收集输入输出数据  $u(t)$  和  $y(t)$ , 由式 (89) 构成输出信息向量  $\varphi_a(t)$ , 由式 (93) 构成输入信息向量  $\varphi_b(t)$ .

3) 由式 (87) 和 (91) 分别计算增益向量  $L_a(t)$  和  $L_b(t)$ , 由式 (88) 和 (92) 分别计算协方差阵  $P_a(t)$  和  $P_b(t)$ .

4) 由 (86) 和 (90) 分别刷新参数估计向量  $\hat{\theta}_a(t)$  和  $\hat{\theta}_b(t)$ .

5)  $t$  增 1, 转到第 2 步继续进行递推计算.

两阶段递推最小二乘算法 (86) — (93) 计算参数估计向量  $\hat{\theta}_a(t)$  和  $\hat{\theta}_b(t)$  的流程如图 1 所示.

### 5.3 仿真例子

例 1 考虑下列仿真对象:

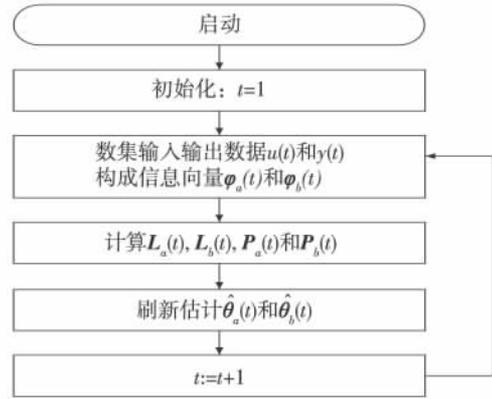


图 1 计算 TS-RLS 参数估计流程  
Fig. 1 The flowchart of computing the TS-RLS estimates  $\hat{\theta}_a(t)$  and  $\hat{\theta}_b(t)$

$$A(z) y(t) = B(z) u(t) + v(t),$$

$$A(z) = 1 - 1.60z^{-1} + 0.80z^{-2},$$

$$B(z) = 0.40z^{-1} + 0.30z^{-2},$$

$$\theta = [a_1 \ a_2 \ b_1 \ b_2]^T = [-1.60 \ 0.80 \ 0.40 \ 0.30]^T.$$

仿真时, 输入  $\{u(t)\}$  零均值单位方差不相关随机信号序列  $\{v(t)\}$  采用零均值方差为  $\sigma^2$  的白噪声序列. 分别应用 RLS 算法和 TS-RLS 算法估计这个系统的参数, 不同噪声方差下, 参数估计及其估计误差如表 1—2 所示. 相对参数估计误差  $\delta = \|\hat{\theta}(t) - \theta\| / \|\theta\|$  随  $t$  变化曲线如图 2—3. 当噪声方差  $\sigma^2 = 0.50^2$  和  $\sigma^2 = 1.00^2$  时, 系统的噪声比分别为  $\delta_{ns} = 73.45\%$  和  $\delta_{ns} = 146.90\%$ .

从表 1—2 和图 2—3, 可以看出: 随着数据长度的增加, 两阶段递推最小二乘算法参数估计很接近递推最小二乘算法参数估计, 但计算量要小.

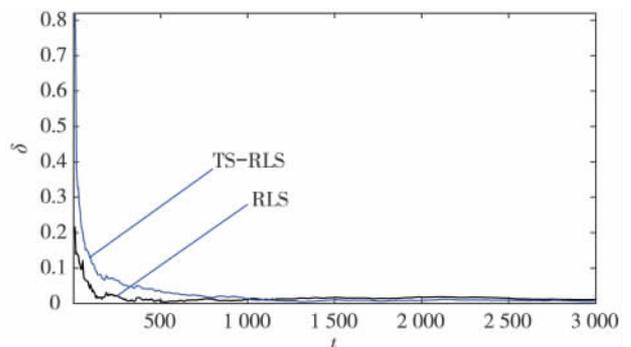


图 2 TS-RLS 和 RLS 参数估计误差  $\delta$  随  $t$  变化曲线 ( $\sigma^2 = 0.50^2$ )

Fig. 2 The TS-RLS and RLS estimation errors  $\delta$  versus  $t$  ( $\sigma^2 = 0.50^2$ )

表 1 TS-RLS 和 RLS 估计及其误差 ( $\sigma^2 = 0.50^2$ )

Table 1 The TS-RLS and RLS estimates and errors ( $\sigma^2 = 0.50^2$ )

算法	$t$	$a_1$	$a_2$	$b_1$	$b_2$	$\delta/\%$
TS-RLS	100	-1.712 11	0.883 28	0.320 57	0.133 66	13.300 39
	200	-1.667 97	0.852 48	0.320 85	0.238 30	7.201 52
	500	-1.637 41	0.824 89	0.378 63	0.261 18	3.673 90
	1 000	-1.618 62	0.811 40	0.390 60	0.292 75	1.373 13
	2 000	-1.605 89	0.801 04	0.394 16	0.313 81	0.865 14
	3 000	-1.606 49	0.803 27	0.395 41	0.308 52	0.666 38
	RLS	100	-1.638 88	0.833 62	0.397 76	0.238 51
200		-1.618 19	0.819 53	0.366 94	0.310 16	2.351 79
500		-1.610 13	0.804 86	0.396 72	0.298 23	0.637 18
1 000		-1.598 20	0.795 88	0.401 25	0.319 38	1.073 33
2 000		-1.591 45	0.789 57	0.398 20	0.331 16	1.830 78
3 000		-1.598 03	0.796 61	0.398 11	0.319 58	1.079 89
真值			-1.600 00	0.800 00	0.400 00	0.300 00

表 2 TS-RLS 和 RLS 估计及其误差 ( $\sigma^2 = 1.00^2$ )

Table 2 The TS-RLS and RLS estimates and errors ( $\sigma^2 = 1.00^2$ )

算法	$t$	$a_1$	$a_2$	$b_1$	$b_2$	$\delta/\%$
TS-RLS	100	-1.674 91	0.861 14	0.296 26	0.094 91	15.168 57
	200	-1.645 69	0.841 30	0.273 01	0.262 31	7.930 05
	500	-1.622 60	0.811 90	0.370 77	0.271 33	3.062 03
	1 000	-1.609 68	0.803 16	0.389 18	0.317 51	1.308 31
	2 000	-1.598 72	0.792 65	0.390 68	0.346 65	2.590 25
	3 000	-1.599 37	0.796 66	0.392 15	0.331 40	1.765 71
	RLS	100	-1.620 05	0.827 97	0.415 45	0.210 35
200		-1.613 31	0.821 77	0.338 47	0.331 24	3.961 32
500		-1.604 39	0.799 73	0.394 86	0.303 01	0.398 61
1 000		-1.596 31	0.793 69	0.402 81	0.339 06	2.144 56
2 000		-1.589 58	0.785 78	0.396 24	0.359 69	3.357 16
3 000		-1.594 51	0.793 14	0.396 12	0.339 61	2.194 40
真值			-1.600 00	0.800 00	0.400 00	0.300 00

把下列 Matlab 程序写到 TwoStage\_RLS\_ARX\_ex1.m 文件中,依次运行  $\sigma = 0.50$  和  $\sigma = 1.00$ ,可得到上述例子的仿真结果(参数估计表和误差曲线图)。

```

1 %-----*
2 % Filename: TwoStage_RLS_ARX_ex1.m for ARX
  models *
3 %  $A(z)y(t) = B(z)u(t) + v(t)$  *
4 % The noise variance  $\sigma^2 = 0.50^2$  and  $1.00^2$  *
5 %-----*
6 clear; format short g
7
8 M = Two - Stage RLS algorithm for ARX models'

```

```

9 FF = 1; % The Forgetting Factor
10 sigma = 1; % The noise variance  $\sigma^2 = 0.50^2$  and  $\sigma^2 = 1.00^2$ 
11 PlotLength = 3000; length1 = PlotLength + 100;
12 na = 2; nb = 2; n = na + nb;
13 a = [-1.6 0.80]; b = [0.40 0.30]; par0 = [a b];
14 p0 = 1e6; P1 = eye(na) * p0; P2 = eye(nb) * p0; P = eye
  (n) * p0;
15 par1 = ones(na, 1) / p0; par2 = ones(nb, 1) / p0;
16 parLS = [par1; par2];
17 %-----Computer the noise - to - signal ratio
18 a1 = [1 a]; b1 = [0 b]; d = 1
19 sy = f_integral(a1 b1); sv = f_integral(a1 d);

```

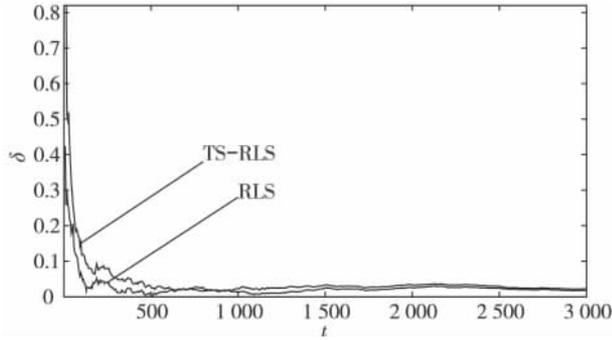


图3 TS-RLS 和 RLS 参数估计误差  $\delta$   
随  $t$  变化曲线 ( $\sigma^2 = 1.00^2$ )  
Fig. 3 The TS-RLS and RLS estimation  
errors  $\delta$  versus  $t$  ( $\sigma^2 = 1.00^2$ )

```

20 delta_ns = sqrt( sv/sy ) * 100 * sigma;
21 [sy sv delta_ns ]
22 %——Generate the input output data
23 rand( 'state' 2 ); randn( 'state' 3 );
24 u = ( rand( length1 , 1 ) - 0.5 ) * sqrt( 12 ); v = randn
    ( length1 , 1 ) * sigma;
25 y = ones( n , 1 ) / p0;
26 for t = n : length1
27     y( t ) = par0 * [ - y( t - 1 : - 1 : t - na ); u( t - 1 : - 1 : t
        - nb ) ] + v( t );
28 end
29 jj = 0; j1 = 0;
30 foit = n : length1
31     jj = jj + 1;
32     varphia = - y( t - 1 : - 1 : t - na ); varphib = u( t - 1 : -
        1 : t - nb );
33     varphi = [ varphia; varphib ];
34     % The RLS estimates
35     L = P * varphi / ( FF + varphi * P * varphi );
36     P = ( P - L * varphi * P ) / FF;
37     parLS = parLS + L * ( y( t ) - varphi * parLS );
38     deltaLS = norm( parLS - par0 ) / norm( par0 );
39     lsLS( jj , : ) = [ jj parLS' deltaLS ];
40
41     par12 = [ par1; par2 ];
42     L1 = P1 * varphia / ( FF + varphia * P1 * varphia );
43     P1 = ( P1 - L1 * varphia * P1 ) / FF;
44     par1 = par1 + L1 * ( y( t ) - varphi * par12 );
45
46     L2 = P2 * varphib / ( FF + varphib * P2 * varphib );
47     P2 = ( P2 - L2 * varphib * P2 ) / FF;
48     par2 = par2 + L2 * ( y( t ) - varphi * par12 );
49     delta = norm( par12 - par0 ) / norm( par0 );

```

```

50     ls( jj , : ) = [ jj par12' delta ];
51     if ( jj = 100 ) | ( jj = 200 ) | ( jj = 500 ) | mod( jj ,
        1000 ) = 0
52         j1 = j1 + 1;
53         ls_100LS( j1 , : ) = [ jj parLS' deltaLS * 100 ];
54         ls_100( j1 , : ) = [ jj par1' par2' delta * 100 ];
55     end
56 end
57 ls_100LS( j1 + 1 , : ) = [ 0 par0' rho ];
58 ls_100( j1 + 1 , : ) = [ 0 par0' rho ];
59
60 fprintf( '\n Two - Stage RLS )
61 fprintf( '\n %s\n' , t a_1 a_2 b_1 b_2 \delta \ ( \% ) \
    \ \ );
62 fprintf( '%5d %10.5f %10.5f %10.5f %10.5f %10.5f %10.5f
    \ \ \ \n' , ls_100 );
63
64 fprintf( '\n RLS )
65 fprintf( '\n %s\n' , t a_1 a_2 b_1 b_2 \delta \ ( \% ) \
    \ \ );
66 fprintf( '%5d %10.5f %10.5f %10.5f %10.5f %10.5f %10.5f
    \ \ \ \n' , ls_100LS );
67
68 figure ( 1 )
69 k = ( 2 * n : 5 : PlotLength - 1 );
70 jk = ls( k , 1 );
71 plot( jk , lsLS( k , n + 2 ) , k' , jk , ls( k , n + 2 ) , b' )
72 xlabel( '\tit t ); ylabel( '{ \it \delta } );
73 axis( [ 0 , PlotLength 0 0.82 ] )
74 text( 800 0.4 , 'Two - Stage RLS )
75 line( [ 100 800 ] , [ ls( 100 , n + 2 ) 0.38 ] )
76 text( 1000 0.3 , 'RLS )
77 line( [ 250 1000 ] , [ lsLS( 250 , n + 2 ) 0.28 ] )

```

## 6 线性回归模型的递阶最小二乘辨识方法

本节主要内容选自文献[8]。最近, Ding 等<sup>[28]</sup>利用多项式变换技术讨论了双率采样数据系统的递阶最小二乘辨识算法。递阶辨识原理已被发展用于多变量系统的辨识。Ding 等提出了双率采样数据系统状态空间模型的递阶辨识方法<sup>[13 29]</sup>, 多变量系统的递阶梯度迭代方法和递阶随机梯度辨识方法<sup>[9]</sup>, 以及递阶最小二乘迭代方法和递阶最小二乘辨识方法<sup>[10]</sup>。相关的讨论见下节的有色噪声干扰多变量系统的递阶辨识方法。

### 6.1 辨识模型与问题构成

考虑用 CAR/ARX 模型描述的动态(大)系统:

$$A(z) y(t) = B(z) u(t) + v(t), \quad (94)$$

其中  $\{u(t)\}$  和  $\{y(t)\}$  分别是系统的输入和输出序列  $\{v(t)\}$  是零均值白噪声序列  $A(z)$  和  $B(z)$  是后移算子  $z^{-1}$  的多项式:

$$A(z) := 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{n_a} z^{-n_a},$$

$$B(z) := b_1 z^{-1} + b_2 z^{-2} + \dots + b_{n_b} z^{-n_b}.$$

$(a_i, b_i)$  为待辨识的参数, 设阶次  $n_a$  和  $n_b$  已知, 维数  $n := n_a + n_b$  很大.  $\varphi(t) \in \mathbf{R}^n$  是由直到  $(t-1)$  时刻系统的输出  $y(t)$  和输入  $u(t)$  构成的回归信息向量.

定义参数向量  $\theta$  和信息向量  $\varphi(t)$  分别为

$$\theta := [a_1 \ a_2 \ \dots \ a_{n_a} \ b_1 \ b_2 \ \dots \ b_{n_b}]^T \in \mathbf{R}^n,$$

$$\varphi(t) := [-y(t-1) \ -y(t-2) \ \dots \ -y(t-n_a) \ u(t-1) \ u(t-2) \ \dots \ u(t-n_b)]^T \in \mathbf{R}^n.$$

则式(94)可以写成如下(总)辨识模型:

$$y(t) = \varphi^T(t) \theta + v(t). \quad (95)$$

虽然标准递推辨识算法, 如递推最小二乘法(RLS):

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t) \varphi(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)], \quad (96)$$

$$P(t) = P(t-1) - \frac{P(t-1) \varphi(t) \varphi^T(t) P(t-1)}{1 + \varphi^T(t) P(t-1) \varphi(t)} \quad (97)$$

可用于大系统(95)的参数估计, 但由于大系统的维数高, 变量和待估计的参数数目多, 使得估计算法的计算量和存储量急剧增加, 以致在算法的实现上造成极大的困难, 这就是所谓的“维数灾”. 在这种情况下, 研究大系统的分解辨识算法, 提出计算量小的辨识算法已成为必然.

## 6.2 递阶最小二乘算法的推导

下面应用递阶辨识原理研究系统(95)的辨识问题, 具体做法如下.

### 1) 辨识模型分解

将大系统(95)的信息向量和参数向量分解为  $N$  个维数为  $n_i$  的子信息向量(sub-information vector)  $\varphi_i(t)$  和子参数向量(sub-parameter vector)  $\theta_i$  如下:

$$\varphi(t) = \begin{bmatrix} \varphi_1(t) \\ \varphi_2(t) \\ \vdots \\ \varphi_N(t) \end{bmatrix}, \quad \varphi_i(t) \in \mathbf{R}^{n_i},$$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{bmatrix}, \quad \theta_i \in \mathbf{R}^{n_i}, \quad n_1 + n_2 + \dots + n_N = n.$$

于是, 可以把系统(95)分解为  $N$  个虚拟子系统(fic-

ticious subsystem), 即子辨识模型(sub-identification model, Sub-ID), 如图4. 它们可以表示为

$$y_i(t) = \varphi_i^T(t) \theta_i + v(t), \quad i = 1, 2, \dots, N, \quad (98)$$

其中

$$y_i(t) := y(t) - \alpha_i(t) = y(t) - \sum_{j=1, j \neq i}^N \varphi_j^T(t) \theta_j, \quad (99)$$

$$\alpha_i(t) := \sum_{j=1, j \neq i}^N \varphi_j^T(t) \theta_j, \quad i = 1, 2, \dots, N, \quad (100)$$

$\alpha_i(t)$  称为各子系统间的关联项(associate items).

2) 子模型辨识或子系统辨识递阶辨识的目标就是用系统的输入输出数据  $\{u(t), y(t)\}$  或  $\{y(t), \varphi(t)\}$  来估计每一个子系统(98)的参数  $\theta_i$ . 由于  $v(t)$  是白噪声, 故根据最小二乘原理, 可获得参数  $\theta_i$  的无偏估计. 估计子系统(98)参数  $\theta_i$  的递推最小二乘算法为

$$\hat{\theta}_i(t) = \hat{\theta}_i(t-1) + L_i(t) [y_i(t) - \varphi_i^T(t) \hat{\theta}_i(t-1)], \quad (101)$$

$$L_i(t) = \frac{P_i(t-1) \varphi_i(t)}{1 + \varphi_i^T(t) P_i(t-1) \varphi_i(t)}, \quad (102)$$

$$P_i^{-1}(t) = P_i^{-1}(t-1) + \varphi_i(t) \varphi_i^T(t), \quad i = 1, 2, \dots, N. \quad (103)$$

这个算法中  $y_i(t)$  是未知的, 把式(99)中  $y_i(t)$  代入式(101)可得

$$\hat{\theta}_i(t) = \hat{\theta}_i(t-1) + L_i(t) [y(t) - \sum_{j=1, j \neq i}^N \varphi_j^T(t) \theta_j - \varphi_i^T(t) \hat{\theta}_i(t-1)]. \quad (104)$$

这个算法中  $\varphi_i(t)$  是已知的, 但式(104)右边包含了其他子系统的未知子参数向量  $\theta_j (j \neq i)$ , 所以无法实现这个算法.

### 3) 子系统辨识算法间关联项协调

为了进行递推计算, 根据递阶辨识原理: 式(104)中未知的  $\theta_j$  用它在  $(t-1)$  时刻的估计值  $\hat{\theta}_j(t-1)$  代替, 可得

$$\hat{\theta}_i(t) = \hat{\theta}_i(t-1) + L_i(t) [y(t) - \sum_{j=1, j \neq i}^N \varphi_j^T(t) \hat{\theta}_j(t-1) - \varphi_i^T(t) \hat{\theta}_i(t-1)] = \hat{\theta}_i(t-1) + L_i(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)] \quad (105)$$

综上所述, 能够得到估计子参数向量  $\theta_i$  的递阶最小二乘辨识算法(Hierarchical Least Squares identification algorithm, HLS 算法)<sup>[8, 28, 31]</sup>:

$$\hat{\theta}_i(t) = \hat{\theta}_i(t-1) + L_i(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)], \quad (106)$$

$$P_i^{-1}(t) = P_i^{-1}(t-1) + \varphi_i(t) \varphi_i^T(t), \quad i = 1, 2, \dots, N. \quad (107)$$

递阶最小二乘辨识算法可以等价写为

$$\hat{\theta}(t) = \hat{\theta}(t-1) + L_i(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)], \quad (108)$$

$$L_i(t) = P_i(t) \varphi_i(t) = \frac{P_i(t-1) \varphi_i(t)}{1 + \varphi_i^T(t) P_i(t-1) \varphi_i(t)}, \quad (109)$$

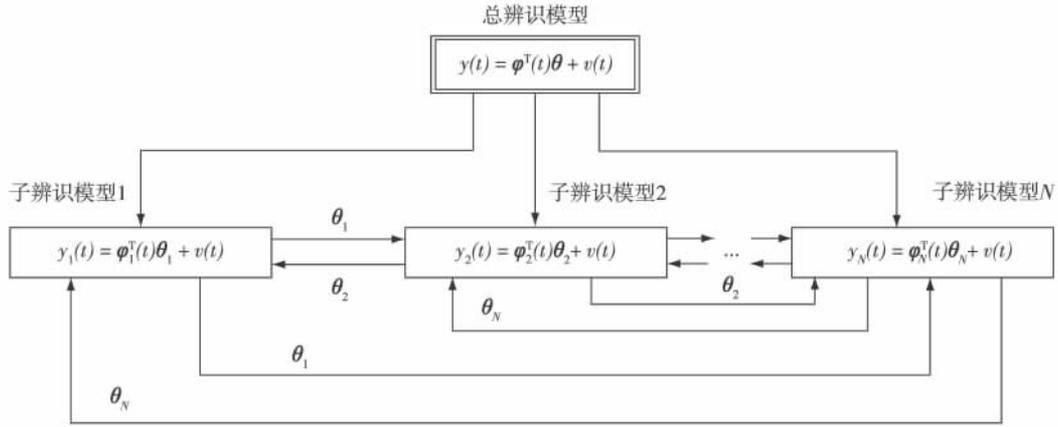


图4 (总) 辨识模型分解为子辨识模型的递阶结构

Fig. 4 The hierarchical structure of decomposition of the identification model

$P_i(t) = [I - L_i(t) \varphi_i^T(t)] P_i(t-1)$ ,  $i = 1, 2, \dots, N$ , (110)  
 算法初值可以选择为  $P_i(0) = p_0 I$ ,  $\hat{\theta}_i(0) = \mathbf{1}_{n_i} / p_0$ ,  
 $p_0 = 10^6$ .

若定义块对角阵 (block diagonal matrix):

$$P(t) := \text{blockdiag} [P_i(t) \quad i = 1, 2, \dots, N] = \begin{bmatrix} P_1(t) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & P_2(t) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & P_N(t) \end{bmatrix}. \quad (111)$$

注意: 这里协方差矩阵  $P(t)$  是重新定义的, 不是式 (96) — (97) 中的  $P(t)$ . 于是, 估计系统 (95) 参数向量  $\theta$  的递阶最小二乘算法可以等价改写为

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t) \varphi(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)], \quad (112)$$

$$P^{-1}(t) = P^{-1}(t-1) + \text{blockdiag} [\varphi_i(t) \varphi_i^T(t), \quad i = 1, 2, \dots, N], \quad P(0) = p_0 I, \quad (113)$$

$$\hat{\theta}(t) = \begin{bmatrix} \hat{\theta}_1(t) \\ \hat{\theta}_2(t) \\ \vdots \\ \hat{\theta}_N(t) \end{bmatrix}. \quad (114)$$

下面比较 RLS 算法和递阶辨识算法的计算量. RLS 算法 (96) — (97) 和递阶辨识算法每迭代计算一步的计算量大小如表 3 所示, 表中数值表示  $n = 100$ ,  $N = 10$ ,  $n_i = 10$ ,  $i = 1, 2, \dots, N$  时, 两种参数估计算法的计算量. 从表 3 可以看出: RLS 算法的计算量很大, 而递阶辨识算法的计算量小得多. 例如, 当  $n = 100$  时, RLS 算法每迭代一步需要的乘法次数为 20 400, 加法次数为 20 200. 把系统分为  $N = 10$  个子系统, 而递阶最小二乘算法每迭代一步需要的乘法

次数为 2 400, 加法次数为 2 200.

表 3 最小二乘辨识算法与递阶辨识算法计算量比较

Table 3 Comparison of computational efficiency of the RLS and HLS algorithms

参数估计方法	乘法次数	加法次数
递推最小二乘算法	$2n^2 + 4n = 20\,400$	$2n^2 + 2n = 20\,200$
递阶最小二乘算法	$\sum_{i=1}^N (2n_i^2 + 4n_i) = 2\,400$	$\sum_{i=1}^N (2n_i^2 + 2n_i) = 2\,200$

### 6.3 递阶最小二乘算法的收敛性

证明递阶最小二乘辨识算法的收敛性, 需要用到下列引理.

引理 8 (对称阵对角块矩阵引理)<sup>[8]</sup> 若分块对称矩阵

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{bmatrix}, \quad A_{ij} = A_{ji}^T \in \mathbf{R}^{n_i \times n_j}$$

满足  $\alpha I \leq A \leq \beta I$ , 其中  $\alpha$  和  $\beta$  为正常数, 则有

$$\alpha I \leq \begin{bmatrix} A_{11} & 0 & \cdots & 0 \\ 0 & A_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{NN} \end{bmatrix} \leq \beta I.$$

引理 9<sup>[8]</sup> 对于大系统 (95), 假设存在正整数  $p$ , 正常数  $\alpha$  和  $\beta$  使得下列强持续激励条件 (SPE) 成立:

$$\alpha I \leq \frac{1}{p} \sum_{i=0}^{p-1} \varphi(t-i) \varphi^T(t-i) \leq \beta I, \quad p \geq n.$$

令

$$H^{-1}(t) := \sum_{j=1}^t \varphi(j) \varphi^T(j) + H^{-1}(0), \quad H(0) := p_0 I,$$

那么协方差矩阵  $H(t)$  满足

$$H^{-1}(t-p) + p\alpha I \leq H^{-1}(t) \leq H^{-1}(t-p) + p\beta I$$

和

$$(t-p)\alpha I \leq H^{-1}(t) \leq [\beta(t+p) + 1/p_0] I.$$

证明 根据  $H(t)$  的定义可得

$$\begin{aligned} H^{-1}(t) &= H^{-1}(t-1) + \varphi(t) \varphi^T(t) = \\ &= H^{-1}(t-2) + \varphi(t-1) \varphi^T(t-1) + \\ &= \varphi(t) \varphi^T(t) = H^{-1}(t-p) + \\ &= \sum_{k=0}^{p-1} \varphi(t-k) \varphi^T(t-k). \end{aligned} \quad (115)$$

利用持续激励条件(SPE), 可得

$$H^{-1}(t-p) + p\alpha I \leq H^{-1}(t) \leq H^{-1}(t-p) + p\beta I.$$

令  $t = pi + k, 0 \leq k < p$ , 于是有

$$\begin{aligned} H^{-1}(t = pi + k) &\leq H^{-1}[p(i-1) + k] + p\beta I = \\ &= H^{-1}[p(i-2) + k] + 2p\beta I = \\ &= H^{-1}(k) + ip\beta I \leq H^{-1}(0) + (i+1)p\beta I \leq \\ &= H^{-1}(0) + (t+p)\beta I = [\beta(t+p) + 1/p_0] I. \end{aligned}$$

同理

$$\begin{aligned} H^{-1}(t = pi + k) &\geq H^{-1}(k) + ip\alpha I \geq \\ &= ip\alpha I \geq (t-p)\alpha I, \quad t > p. \end{aligned}$$

引理 9 证毕.

定理 9<sup>[8]</sup> 对于大系统(95), 如果强持续激励条件(SPE)成立  $\{v(t)\}$  满足假设:

- (A1)  $E[v(t) | \mathcal{F}_{t-1}] = 0$  a. s. ,
- (A2)  $E[v^2(t) | \mathcal{F}_{t-1}] = \sigma^2(t) \leq \sigma^2 < \infty$  a. s. ,
- (A3)  $\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{i=1}^t v^2(i) \leq \sigma^2 < \infty$  a. s. ,

其中  $\{v(t), \mathcal{F}_t\}$  是鞅差序列,  $\mathcal{F}_t$  是由直到  $t$  时刻的观测生成的  $\sigma$  代数. 递阶辨识算法(106) — (107) 或 (108) — (110) 给出的参数估计  $\hat{\theta}(t)$  一致(a. s.)收敛于真参数向量  $\theta$ , 或参数估计误差一致收敛于零, 即  $\lim_{t \rightarrow \infty} \hat{\theta}(t) = \theta$  a. s.

## 7 类多变量 CARMA 系统的递阶辨识方法

递阶辨识原理是研究复杂多变量系统辨识方法的一个有效工具<sup>[9-10, 13]</sup>. 最近 Zhang 等<sup>[41]</sup> 研究了多变量输出误差滑动平均系统的递阶梯度迭代参数估计算法, Xiang 等<sup>[42]</sup> 研究了单输入多输出系统的辅助模型递阶最小二乘算法. 下面介绍 Han 等<sup>[43]</sup> 提出的滑动平均噪声干扰的多变量系统递阶梯度迭代方

法和递阶最小二乘迭代算法.

考虑类多变量 CARMA 系统(即多变量 CARMA-like 系统)<sup>[9-10, 43]</sup>,

$$\alpha(z) y(t) = Q(z) u(t) + D(z) v(t), \quad (116)$$

其中  $\alpha(z)$  是系统的  $n$  次特征多项式, 它定义为传递矩阵的各元最小公分母首一多项式,  $Q(z)$  和  $D(z)$  是  $z^{-1}$  的矩阵多项式:

$$\begin{aligned} \alpha(z) &:= 1 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \cdots + \alpha_n z^{-n}, \quad \alpha_i \in \mathbf{R}^1, \\ Q(z) &:= Q_1 z^{-1} + Q_2 z^{-2} + \cdots + Q_n z^{-n}, \quad Q_i \in \mathbf{R}^{m \times r}, \\ D(z) &:= I + D_1 z^{-1} + D_2 z^{-2} + \cdots + D_n z^{-n}, \quad D_i \in \mathbf{R}^{m \times m}. \end{aligned}$$

特征多项式  $\alpha(z)$  的辨识具有重要的意义, 如系统极点配置、观测器设计等<sup>[9-10]</sup>.

首先考虑一种特殊情形:  $D(z) = d(z) I$ , 然后考虑  $D(z)$  为矩阵情形, 这里  $d(z)$  为一多项式

$$d(z) := 1 + d_1 z^{-1} + d_2 z^{-2} + \cdots + d_n z^{-n}, \quad d_i \in \mathbf{R},$$

则模型(116)也可以为

$$\alpha(z) y(t) = Q(z) u(t) + d(z) v(t), \quad (117)$$

其系统结构如图 5 所示.  $G(z) := \frac{Q(z)}{\alpha(z)} \in \mathbf{R}^{m \times r}$  为系统的传递函数矩阵.

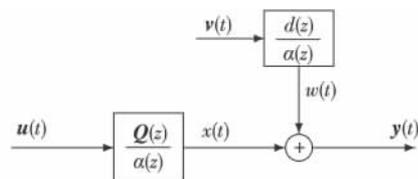


图 5 CARMA-like 模型描述的多变量系统

Fig. 5 A multivariable system described by CARMA-like models

### 7.1 递阶梯度迭代辨识算法

定义参数向量  $\vartheta$ 、参数矩阵  $\theta$ 、输入信息向量  $\varphi(t)$  和输出信息矩阵  $\psi(t)$  如下:

$$\begin{aligned} \vartheta &:= [\alpha_1 \ \alpha_2 \ \cdots \ \alpha_n \ d_1 \ d_2 \ \cdots \ d_{n_d}]^T \in \mathbf{R}^{n+n_d}, \\ \theta^T &:= [Q_1 \ Q_2 \ \cdots \ Q_n] \in \mathbf{R}^{m \times (nr)}, \end{aligned}$$

$$\varphi(t) := \begin{bmatrix} u(t-1) \\ u(t-2) \\ \vdots \\ u(t-n) \end{bmatrix} \in \mathbf{R}^{(nr)},$$

$$\begin{aligned} \psi(t) &:= [y(t-1) \ y(t-2) \ \cdots \ y(t-n), \\ &= -v(t-1) \ -v(t-2) \ \cdots \ -v(t-n_d)] \in \\ &= \mathbf{R}^{m \times (n+n_d)}. \end{aligned}$$

于是, 从式(116)可以得到辨识模型:

$$y(t) + \psi(t) \vartheta = \theta^T \varphi(t) + v(t). \quad (118)$$

这个模型待辨识的参数包括两部分: 一个是参数向量  $\vartheta$ , 一个是由传递矩阵分子多项式矩阵系数构成

的参数矩阵  $\theta$ , 且信息矩阵  $\psi(t)$  中还包含了未知噪声项  $v(t-i)$ , 所以传统的最小二乘辨识方法不能直接应用, 这是辨识的困难. 这里基于递阶辨识原理和迭代搜索原理来解决这一困难. 递阶辨识的基本原理就是将一个系统分解为多个维数较小、变量较少的子系统, 然后分别辨识每个子系统的参数. 迭代搜索原理是将系统中信息矩阵所含的未知噪声项用其迭代估计值来代替<sup>[11, 36, 44]</sup>. 具体细节如下.

定义两个中间向量:

$$\begin{aligned} b_1(t) &:= \theta^T \varphi(t), \\ b_2(t) &:= \psi(t) \vartheta. \end{aligned}$$

分解式(118)为下列两个虚拟子系统:

$$\begin{aligned} S_1: \quad y(t) &= -\psi(t) \vartheta + b_1(t) + v(t), \\ S_2: \quad y(t) &= \theta^T \varphi(t) - b_2(t) + v(t). \end{aligned}$$

设数据长度  $t \gg mn_0 + n$ , 定义

$$\begin{aligned} Y_1(t) &:= \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(t) \end{bmatrix} \in \mathbf{R}^{(mt)}, \\ \Psi(t) &:= \begin{bmatrix} \psi(1) \\ \psi(2) \\ \vdots \\ \psi(t) \end{bmatrix} \in \mathbf{R}^{(mt) \times (n+n_d)}, \\ V_1(t) &:= \begin{bmatrix} v(1) \\ v(2) \\ \vdots \\ v(t) \end{bmatrix} \in \mathbf{R}^{(mt)}, \\ B_1(t) &:= \begin{bmatrix} b_1(1) \\ b_1(2) \\ \vdots \\ b_1(t) \end{bmatrix} = \begin{bmatrix} \theta^T \varphi(1) \\ \theta^T \varphi(2) \\ \vdots \\ \theta^T \varphi(t) \end{bmatrix} \in \mathbf{R}^{(mt)}, \quad (119) \end{aligned}$$

$$\begin{aligned} Y_2(t) &:= [y(1) \ y(2) \ \cdots \ y(t)] \in \mathbf{R}^{m \times t}, \\ \Phi(t) &:= [\varphi(1) \ \varphi(2) \ \cdots \ \varphi(t)] \in \mathbf{R}^{(nr) \times t}, \\ B_2(t) &:= [b_2(1) \ b_2(2) \ \cdots \ b_2(t)] = \\ &[\psi(1) \vartheta \ \psi(2) \vartheta \ \cdots \ \psi(t) \vartheta] \in \mathbf{R}^{m \times t}, \quad (120) \\ V_2(t) &:= [v(1) \ v(2) \ \cdots \ v(t)] \in \mathbf{R}^{m \times t}. \end{aligned}$$

因此可以得到

$$\begin{aligned} S_1: \quad Y_1(t) &= -\Psi(t) \vartheta + B_1(t) + V_1(t), \\ S_2: \quad Y_2(t) &= \theta^T \Phi(t) - B_2(t) + V_2(t). \end{aligned}$$

定义两个准则函数:

$$\begin{aligned} J_1(\vartheta) &= \|Y_1(t) + \Psi(t) \vartheta - B_1(t)\|^2, \\ J_2(\theta) &= \|Y_2(t) - \theta^T \Phi(t) + B_2(t)\|^2. \end{aligned}$$

令  $k=1, 2, \dots$  为迭代变量,  $\hat{\vartheta}_k$  和  $\hat{\theta}_k$  分别为  $\vartheta$  和  $\theta$  在

第  $k$  次迭代的参数估计,  $\mu_k(t) \geq 0$  为时变步长(时变收敛因子). 使用负梯度搜索, 极小化准则函数  $J_1(\vartheta)$  和  $J_2(\theta)$  得到  $\vartheta$  和  $\theta$  迭代解:

$$\begin{aligned} \hat{\vartheta}_k &= \hat{\vartheta}_{k-1} - \frac{\mu_k(t)}{2} \text{grad}[J_1(\hat{\vartheta}_{k-1})] = \\ &\hat{\vartheta}_{k-1} - \mu_k(t) \Psi^T(t) [Y_1(t) - B_1(t) + \Psi(t) \hat{\vartheta}_{k-1}], \\ \hat{\theta}_k &= \hat{\theta}_{k-1} - \frac{\mu_k(t)}{2} \text{grad}[J_2(\hat{\theta}_{k-1})] = \\ &\hat{\theta}_{k-1}^T + \mu_k(t) \Phi(t) [Y_2(t) - \hat{\theta}_{k-1}^T \Phi(t) + B_2(t)]. \end{aligned}$$

将式(119)中  $B_1(t)$  和式(120)中  $B_2(t)$  分别代入以上两式得到

$$\begin{aligned} \hat{\vartheta}_k &= \hat{\vartheta}_{k-1} - \mu_k(t) \Psi^T(t) \cdot \\ &\left( Y_1(t) - \begin{bmatrix} \theta^T \varphi(1) \\ \theta^T \varphi(2) \\ \vdots \\ \theta^T \varphi(t) \end{bmatrix} + \Psi(t) \hat{\vartheta}_{k-1} \right), \quad (121) \end{aligned}$$

$$\begin{aligned} \hat{\theta}_k &= \hat{\theta}_{k-1} + \mu_k(t) \Phi(t) \{ Y_2(t) - \hat{\theta}_{k-1}^T \Phi(t) + \\ &[\psi(1) \vartheta \ \psi(2) \vartheta \ \cdots \ \psi(t) \vartheta] \}^T. \quad (122) \end{aligned}$$

由于以上两式右边分别含有未知参数矩阵  $\theta$  和参数向量  $\vartheta$ , 故算法(121) — (122) 不能实现. 为解决这一困难, 利用递阶辨识原理<sup>[9-10, 36]</sup>, 式(121)中  $\theta$  和式(122)中  $\vartheta$  分别用它们在前一次迭代的估计  $\hat{\theta}_{k-1}$  和  $\hat{\vartheta}_{k-1}$  代替, 得到

$$\begin{aligned} \hat{\vartheta}_k &= \hat{\vartheta}_{k-1} - \mu_k(t) \Psi^T(t) \cdot \\ &\left( Y_1(t) - \begin{bmatrix} \hat{\theta}_{k-1}^T \varphi(1) \\ \hat{\theta}_{k-1}^T \varphi(2) \\ \vdots \\ \hat{\theta}_{k-1}^T \varphi(t) \end{bmatrix} + \Psi(t) \hat{\vartheta}_{k-1} \right), \quad (123) \end{aligned}$$

$$\begin{aligned} \hat{\theta}_k &= \hat{\theta}_{k-1} + \mu_k(t) \Phi(t) \{ Y_2(t) - \hat{\theta}_{k-1}^T \Phi(t) + \\ &[\psi(1) \hat{\vartheta}_{k-1} \ \psi(2) \hat{\vartheta}_{k-1} \ \cdots \ \psi(t) \hat{\vartheta}_{k-1}] \}^T. \quad (124) \end{aligned}$$

由于  $\Psi(t)$ , 也就是  $\psi(t)$  中包含了不可测噪声项  $v(t-i)$ , 根据迭代搜索原理, 将  $\psi(t)$  中的  $v(t-i)$  用其第  $k-1$  次迭代估计值  $\hat{v}_{k-1}(t-i)$  代替, 代替后的  $\psi(t)$  记作

$$\begin{aligned} \hat{\psi}_k(t) &:= [y(t-1) \ y(t-2) \ \cdots \ y(t-n), \\ &-\hat{v}_{k-1}(t-1) \ \cdots \ -\hat{v}_{k-1}(t-2) \ \cdots \ , \\ &-\hat{v}_{k-1}(t-n_d)] \in \mathbf{R}^{m \times (n+n_d)}. \end{aligned}$$

由式(118)可得

$$v(t) = y(t) + \psi(t) \vartheta - \theta^T \varphi(t).$$

用  $\hat{\psi}_k(t)$  和  $\hat{\theta}_k$  代替上式中的  $\psi(t)$  和  $\theta$ , 那么  $v(t)$  第  $k$  次迭代估计值  $\hat{v}_k(t)$  可由下式计算:

$$\hat{v}_k(t) = y(t) + \hat{\psi}_k(t) \hat{\vartheta}_k - \hat{\theta}_k^T \varphi(t).$$

定义

$$\hat{\Psi}_k(t) = \begin{bmatrix} \hat{\psi}_k(1) \\ \hat{\psi}_k(2) \\ \vdots \\ \hat{\psi}_k(t) \end{bmatrix} \in \mathbf{R}^{(ml) \times (n+n_d)}$$

用  $\hat{\Psi}_k(t)$  和  $\hat{\psi}_k(t)$  分别代替式(123)和(124)中的  $\Psi(t)$  和  $\psi(t)$  得到

$$\hat{\theta}_k = \hat{\theta}_{k-1} - \mu_k(t) \hat{\Psi}_k^T(t) \cdot \left( \begin{array}{c} Y_1(t) - \begin{bmatrix} \hat{\theta}_{k-1}^T \varphi(1) \\ \hat{\theta}_{k-1}^T \varphi(2) \\ \vdots \\ \hat{\theta}_{k-1}^T \varphi(t) \end{bmatrix} + \hat{\psi}_k(t) \hat{\theta}_{k-1} \end{array} \right), \quad (125)$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \mu_k(t) \Phi(t) \{ Y_2(t) - \hat{\theta}_{k-1}^T \Phi(t) + [\hat{\psi}_k(1) \hat{\theta}_{k-1} \hat{\psi}_k(2) \hat{\theta}_{k-1} \cdots \hat{\psi}_k(t) \hat{\theta}_{k-1}]^T \}, \quad (126)$$

或

$$\hat{\theta}_k = [I - \mu_k(t) \hat{\Psi}_k^T(t) \hat{\Psi}_k(t)] \hat{\theta}_{k-1} - \mu_k(t) \hat{\Psi}_k^T(t) \cdot \left( \begin{array}{c} Y_1(t) - \begin{bmatrix} \hat{\theta}_{k-1}^T \varphi(1) \\ \hat{\theta}_{k-1}^T \varphi(2) \\ \vdots \\ \hat{\theta}_{k-1}^T \varphi(t) \end{bmatrix} \end{array} \right),$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} [I - \mu_k(t) \Phi(t) \Phi^T(t)] + \mu_k(t) \Phi(t) \{ Y_2(t) + [\hat{\psi}_k(1) \hat{\theta}_{k-1} \hat{\psi}_k(2) \hat{\theta}_{k-1} \cdots \hat{\psi}_k(t) \hat{\theta}_{k-1}]^T \},$$

其中  $I$  表示适当维数的单位向量。以上两式可看作是二个参数估计作为状态的离散状态方程,其收敛的必要条件是矩阵  $[I - \mu_k(t) \hat{\Psi}_k^T(t) \hat{\Psi}_k(t)]$  和  $[I - \mu_k(t) \Phi(t) \Phi^T(t)]$  的特征值在单位圆内,所以收敛因子必须满足

$$\mu_k(t) \leq \frac{2}{\lambda_{\max} [\hat{\Psi}_k^T(t) \hat{\Psi}_k(t)]},$$

$$\mu_k(t) \leq \frac{2}{\lambda_{\max} [\Phi(t) \Phi^T(t)]},$$

其中  $\lambda_{\max} [X]$  为方阵  $X$  的最大特征值,它们的交集为

$$\mu_k(t) \leq \frac{2}{\max\{\lambda_{\max} [\hat{\Psi}_k^T(t) \hat{\Psi}_k(t)] \lambda_{\max} [\Phi(t) \Phi^T(t)]\}}$$

一种保守的选择是

$$\mu_k(t) \leq \frac{2}{\lambda_{\max} [\hat{\Psi}_k^T(t) \hat{\Psi}_k(t)] + \lambda_{\max} [\Phi(t) \Phi^T(t)]},$$

或

$$\mu_k(t) \leq \frac{2}{\|\hat{\Psi}_k(t)\|^2 + \|\Phi(t)\|^2}. \quad (127)$$

综合以上各式 把  $Y_1(t)$ ,  $\hat{\Psi}_k(t)$ ,  $Y_2(t)$ ,  $\Phi(t)$  的

定义式代入式(125) (126) 和(127),可总结出多变量 CARMA-like 模型的递阶梯度迭代辨识算法 (Hierarchical Gradient based Iterative identification algorithm, HGI):

$$\hat{\theta}_k = \hat{\theta}_{k-1} - \mu_k(t) \sum_{j=1}^t \hat{\psi}_k^T(j) [y(j) + \hat{\psi}_k(j) \hat{\theta}_{k-1} - \hat{\theta}_{k-1}^T \varphi(j)], \quad (128)$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \mu_k(t) \sum_{j=1}^t \varphi(t) [y(j) + \hat{\psi}_k(j) \hat{\theta}_{k-1} - \hat{\theta}_{k-1}^T \varphi(j)]^T, \quad (129)$$

$$\varphi(t) = [u^T(t-1) \ u^T(t-2) \ \cdots \ u^T(t-n)]^T \quad (130)$$

$$\hat{\psi}_k(t) = [y(t-1) \ y(t-2) \ \cdots \ y(t-n) \ -\hat{y}_{k-1}(t-1) \ -\hat{y}_{k-1}(t-2) \ \cdots \ -\hat{y}_{k-1}(t-n_d)], \quad (131)$$

$$\hat{y}_k(t) = y(t) + \hat{\psi}_k(t) \hat{\theta}_k - \hat{\theta}_k^T \varphi(t), \quad (132)$$

$$\mu_k(t) \leq 2 \left( \sum_{j=1}^t [\|\hat{\psi}_k(j)\|^2 + \|\varphi(j)\|^2] \right)^{-1} = \mu_0(t). \quad (133)$$

在计算参数估计时,可以取算法中  $t=L$ ,  $L$  为数据长度. HGI 算法(128) — (133) 的计算步骤如下.

1) 令  $k=1$ , 置初值  $\hat{y}_0(t) = 0$ ,  $\theta_0 = \mathbf{1}_{n+n_d}/p_0$ ,  $\theta_0^T = \mathbf{1}_{m \times (nr)}/p_0$ ,  $\mathbf{1}_{n+n_d}$  是元素全为 1 的  $\{(n+n_d)\}$  维列向量,  $\mathbf{1}_{m \times (nr)}$  是元素全为 1 的  $m \times (nr)$  维矩阵,  $p_0 = 10^6$ .

2) 收集输入输出数据  $\{u(t) \ y(t) : t=1, 2, \dots, L\}$ ,  $L$  为数据长度,用式(11)构造  $\varphi(t)$ . 给定小正数  $\epsilon$ .

3) 用式(131)构造  $\hat{\psi}_k(t)$ , 根据式(133)选择  $\mu_k(t)$ . 为使算法有快的收敛速度,可取尽可能大的  $\mu_k(t)$ , 如取  $\mu_k(t) = \mu_0(t)$ .

4) 根据式(128)和式(129)刷新估计向量  $\hat{\theta}_k$  和参数估计矩阵  $\hat{\theta}_k$ , 用式(132)计算  $\hat{y}_k(t)$ .

5) 如果  $\hat{\theta}_k$  与  $\hat{\theta}_{k-1}$ ,  $\hat{\theta}_k$  与  $\hat{\theta}_{k-1}$  很接近, 或者  $\|\hat{\theta}_k - \hat{\theta}_{k-1}\| + \|\hat{\theta}_k - \hat{\theta}_{k-1}\| \leq \epsilon$ , 就结束计算, 获得参数估计  $\hat{\theta}_k$  和  $\hat{\theta}_k$ , 否则  $k$  增 1, 转到步骤 3.

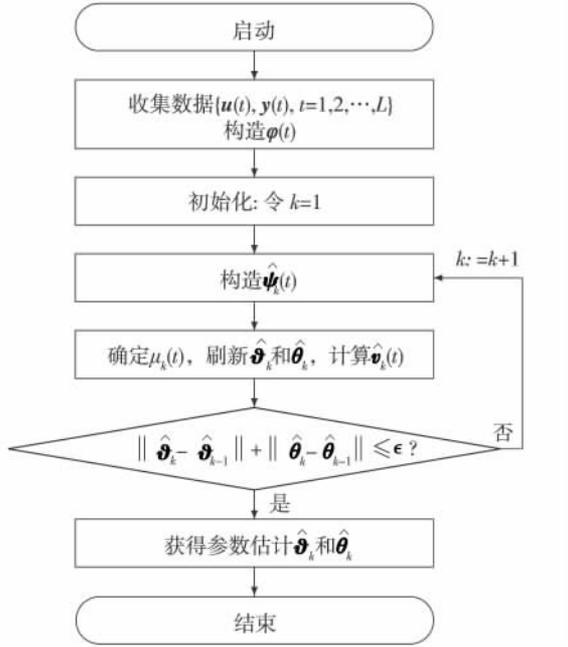
HGI 算法的流程如图 6 所示.

## 7.2 递阶最小二乘迭代算法

本节的多变量 CARMA-like 系统的递阶最小二乘迭代算法参见文献[43].

运用推论 2 极小化上述两个准则函数,可得到  $\theta$  和  $\theta$  的迭代表达式:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \mu_k(t) [\Psi^T(t) \Psi(t)]^{-1} \Psi^T(t) [B_1(t) - Y_1(t) - \Psi(t) \hat{\theta}_{k-1}], \quad (134)$$

图 6 计算 HGI 参数估计  $\hat{\boldsymbol{\theta}}_k$  和  $\hat{\boldsymbol{\vartheta}}_k$  的流程Fig. 6 The flowchart of computing the HGI estimates  $\hat{\boldsymbol{\theta}}_k$  and  $\hat{\boldsymbol{\vartheta}}_k$ 

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \mu_k(t) \{ [\mathbf{B}_2(t) + \mathbf{Y}_2(t) - \hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\Phi}(t)] \boldsymbol{\Phi}^T(t) [\boldsymbol{\Phi}(t) \boldsymbol{\Phi}^T(t)]^{-1} \}^T, \quad (135)$$

其中  $k=1, 2, \dots, \mu_k(t) \geq 0$ ,  $\hat{\boldsymbol{\vartheta}}_k$  和  $\hat{\boldsymbol{\theta}}_k$  与递阶梯度迭代算法中的定义相同. 将式 (119) 中  $\mathbf{B}_1(t)$  和式 (120) 中  $\mathbf{B}_2(t)$  分别代入式 (134) 和 (135) 得到

$$\hat{\boldsymbol{\vartheta}}_k = \hat{\boldsymbol{\vartheta}}_{k-1} + \mu_k(t) [\boldsymbol{\Psi}^T(t) \boldsymbol{\Psi}(t)]^{-1} \boldsymbol{\Psi}^T(t) \cdot \left( \begin{bmatrix} \boldsymbol{\theta}^T \boldsymbol{\varphi}(1) \\ \boldsymbol{\theta}^T \boldsymbol{\varphi}(2) \\ \vdots \\ \boldsymbol{\theta}^T \boldsymbol{\varphi}(t) \end{bmatrix} - \mathbf{Y}_1(t) - \boldsymbol{\Psi}(t) \hat{\boldsymbol{\vartheta}}_{k-1} \right), \quad (136)$$

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \mu_k(t) \{ ([\boldsymbol{\psi}(1) \boldsymbol{\vartheta} \boldsymbol{\psi}(2) \boldsymbol{\vartheta} \cdots, \boldsymbol{\psi}(t) \boldsymbol{\vartheta}] + \mathbf{Y}_2(t) - \hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\Phi}(t)) \boldsymbol{\Phi}^T(t) [\boldsymbol{\Phi}(t) \boldsymbol{\Phi}^T(t)]^{-1} \}^T. \quad (137)$$

根据递阶辨识原理<sup>[9]</sup>, 式 (136) 中  $\boldsymbol{\theta}$  和式 (137) 中  $\boldsymbol{\vartheta}$  分别用它们前一次的迭代估计  $\hat{\boldsymbol{\theta}}_{k-1}$  和  $\hat{\boldsymbol{\vartheta}}_{k-1}$  代替, 得到

$$\hat{\boldsymbol{\vartheta}}_k = \hat{\boldsymbol{\vartheta}}_{k-1} + \mu_k(t) [\boldsymbol{\Psi}^T(t) \boldsymbol{\Psi}(t)]^{-1} \boldsymbol{\Psi}^T(t) \cdot \left( \begin{bmatrix} \hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\varphi}(1) \\ \hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\varphi}(2) \\ \vdots \\ \hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\varphi}(t) \end{bmatrix} - \mathbf{Y}_1(t) - \boldsymbol{\Psi}(t) \hat{\boldsymbol{\vartheta}}_{k-1} \right), \quad (138)$$

$$\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \mu_k(t) \{ ([\boldsymbol{\psi}(1) \hat{\boldsymbol{\vartheta}}_{k-1} \boldsymbol{\psi}(2) \hat{\boldsymbol{\vartheta}}_{k-1} \cdots, \boldsymbol{\psi}(t) \hat{\boldsymbol{\vartheta}}_{k-1}] + \mathbf{Y}_2(t) -$$

$$\hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\Phi}(t)) \boldsymbol{\Phi}^T(t) [\boldsymbol{\Phi}(t) \boldsymbol{\Phi}^T(t)]^{-1} \}^T. \quad (139)$$

将  $\mathbf{Y}_1(t)$ ,  $\mathbf{Y}_2(t)$ ,  $\boldsymbol{\Psi}(t)$  和  $\boldsymbol{\Phi}(t)$  的定义式分别代入式 (138) 和 (139) 可以得到

$$\begin{aligned} \hat{\boldsymbol{\vartheta}}_k &= \hat{\boldsymbol{\vartheta}}_{k-1} - \mu_k(t) [\boldsymbol{\Psi}^T(t) \boldsymbol{\Psi}(t)]^{-1} \sum_{j=1}^t \boldsymbol{\Psi}^T(j) [\mathbf{y}(j) + \boldsymbol{\psi}(j) \hat{\boldsymbol{\vartheta}}_{k-1} - \hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\varphi}(j)] = \\ &= \hat{\boldsymbol{\vartheta}}_{k-1} - \mu_k(t) \left[ \sum_{j=1}^t \boldsymbol{\Psi}^T(j) \boldsymbol{\psi}(j) \right]^{-1} \sum_{j=1}^t \boldsymbol{\Psi}^T(j) [\mathbf{y}(j) + \boldsymbol{\psi}(j) \hat{\boldsymbol{\vartheta}}_{k-1} - \hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\varphi}(j)] = \\ &= [1 - \mu_k(t)] \hat{\boldsymbol{\vartheta}}_{k-1} - \mu_k(t) \left[ \sum_{j=1}^t \boldsymbol{\Psi}^T(j) \boldsymbol{\psi}(j) \right]^{-1} \cdot \\ &= \sum_{j=1}^t \boldsymbol{\Psi}^T(j) [\mathbf{y}(j) - \hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\varphi}(j)], \quad (140) \end{aligned}$$

$$\begin{aligned} \hat{\boldsymbol{\theta}}_k &= \hat{\boldsymbol{\theta}}_{k-1} + \mu_k(t) [\boldsymbol{\Phi}(t) \boldsymbol{\Phi}^T(t)]^{-1} \sum_{j=1}^t \boldsymbol{\varphi}(j) [\mathbf{y}(j) + \boldsymbol{\psi}(j) \hat{\boldsymbol{\vartheta}}_{k-1} - \hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\varphi}(j)] = \hat{\boldsymbol{\theta}}_{k-1} + \\ &= \mu_k(t) \left[ \sum_{j=1}^t \boldsymbol{\varphi}(j) \boldsymbol{\varphi}^T(j) \right]^{-1} \sum_{j=1}^t \boldsymbol{\varphi}(j) \cdot \\ &= [\mathbf{y}(j) + \boldsymbol{\psi}(j) \hat{\boldsymbol{\vartheta}}_{k-1} - \hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\varphi}(j)]^T = \\ &= [1 - \mu_k(t)] \hat{\boldsymbol{\theta}}_{k-1} + \mu_k(t) \left[ \sum_{j=1}^t \boldsymbol{\varphi}(j) \boldsymbol{\varphi}^T(j) \right]^{-1} \cdot \\ &= \sum_{j=1}^t \boldsymbol{\varphi}(j) [\mathbf{y}(j) + \boldsymbol{\psi}(j) \hat{\boldsymbol{\vartheta}}_{k-1}]^T. \quad (142) \end{aligned}$$

与递阶梯度迭代辨识算法类似,  $\boldsymbol{\psi}(t)$  中包含的不可测噪声项  $\mathbf{v}(t-i)$  用其第  $k-1$  次迭代的估计值  $\hat{\mathbf{v}}_{k-1}(t-i)$  代替, 取  $\mu_k(t) = 1$ , 可得多变量 CARMA-like 系统的递阶最小二乘迭代辨识算法 (Hierarchical Least Squares based Iterative identification algorithm, HLSI 算法):

$$\hat{\boldsymbol{\vartheta}}_k = - \left[ \sum_{j=1}^t \hat{\boldsymbol{\psi}}_k^T(j) \hat{\boldsymbol{\psi}}_k(j) \right]^{-1} \sum_{j=1}^t \hat{\boldsymbol{\psi}}_k^T(j) [\mathbf{y}(j) - \hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\varphi}(j)], \quad (142)$$

$$\hat{\boldsymbol{\theta}}_k = \left[ \sum_{j=1}^t \boldsymbol{\varphi}(j) \boldsymbol{\varphi}^T(j) \right]^{-1} \sum_{j=1}^t \boldsymbol{\varphi}(j) [\mathbf{y}(j) + \hat{\boldsymbol{\psi}}_k(j) \hat{\boldsymbol{\vartheta}}_{k-1}]^T, \quad (143)$$

$$\begin{aligned} \boldsymbol{\varphi}(t) &= [\mathbf{u}^T(t-1) \ \mathbf{u}^T(t-2) \ \cdots \ \mathbf{u}^T(t-n)]^T, \quad (144) \\ \hat{\boldsymbol{\psi}}_k(t) &= [\mathbf{y}(t-1) \ \mathbf{y}(t-2) \ \cdots \ \mathbf{y}(t-n), \\ &= -\hat{\mathbf{v}}_{k-1}(t-1) \ \cdots \ -\hat{\mathbf{v}}_{k-1}(t-n_d)], \quad (145) \end{aligned}$$

$$\hat{\mathbf{v}}_k(t) = \mathbf{y}(t) + \hat{\boldsymbol{\psi}}_k(t) \hat{\boldsymbol{\vartheta}}_k - \hat{\boldsymbol{\theta}}_k^T \boldsymbol{\varphi}(t). \quad (146)$$

在计算参数估计时, 可取这个算法中的  $t=L$ ,  $L$  为数据长度. 递阶最小二乘迭代算法 (142) — (146) 的计算步骤如下.

1) 收集输入输出数据  $\{u(t) \ y(t) : t=1, 2, \dots, L\}$   $L$  为数据长度. 用式(144) 构造  $\varphi(t)$  给定小正数  $\epsilon$ .

2) 令  $k=1$ , 置初值  $\hat{\theta}_0 = \mathbf{1}_{n+n_d}$   $\hat{\theta}_0^T = \mathbf{1}_{m \times (nr)}$   $\mathbf{1}_{n+n_d}$  是元素全为 1 的  $(n+n_d)$  维列向量  $\mathbf{1}_{m \times (nr)}$  是元素全为 1 的  $m \times (nr)$  维矩阵  $\hat{\rho}_0(t)$  为随机向量.

3) 用式(145) 构造  $\hat{\psi}_k(t)$ .

4) 用式(142) 和式(143) 刷新参数估计  $\hat{\theta}_k$  和  $\hat{\theta}_k$ .

5) 用式(146) 计算  $\hat{v}_k(t)$ .

6) 如果  $\hat{\theta}_k$  与  $\hat{\theta}_{k-1}$   $\hat{\theta}_k$  与  $\hat{\theta}_{k-1}$  很接近, 满足

$$\|\hat{\theta}_k - \hat{\theta}_{k-1}\| + \|\hat{\theta}_k - \hat{\theta}_{k-1}\| \leq \epsilon$$

就结束计算, 获得参数估计  $\hat{\theta}_k$  和  $\hat{\theta}_k$ , 否则  $k$  增 1, 转到步骤 3.

HLSI 算法计算参数的流程如图 7 所示.

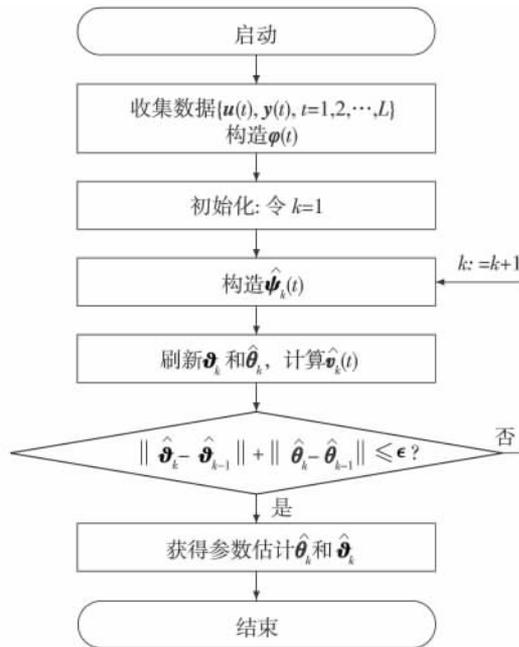


图 7 计算 HLSI 参数估计  $\hat{\theta}_k$  和  $\hat{\theta}_k$  的流程

Fig.7 The flowchart of computing the HLSI estimates  $\hat{\theta}_k$  and  $\hat{\theta}_k$

### 7.3 交互噪声干扰的情形

下面考虑  $D(z)$  多项式矩阵情形, 重写交互噪声干扰下的多变量 CARMA-like 模型(116) 如下:

$$\alpha(z) y(t) = Q(z) u(t) + D(z) v(t), \quad (147)$$

式(147) 中各变量的定义同上.

交互噪声干扰情形, 即  $D(z)$  为多项式矩阵情形, 推导递阶辨识方法的思路与系统(117) 类似, 只是参数向量  $\vartheta$ 、参数矩阵  $\theta$ 、信息向量  $\varphi(t)$  和信息矩阵  $\psi(t)$  需重新定义:

$$\vartheta = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_n]^T \in \mathbf{R}^n,$$

$$\theta^T = [Q_1 \ Q_2 \ \dots \ Q_n \ D_1 \ D_2 \ \dots \ D_{n_d}] \in \mathbf{R}^{m \times (nr+mn_d)},$$

$$\varphi(t) = [u^T(t-1) \ \mu^T(t-2) \ \dots \ \mu^T(t-n) \ v^T(t-1) \ y^T(t-2) \ \dots \ y^T(t-n_d)]^T \in \mathbf{R}^{(nr+mn_d)},$$

$$\psi(t) = [y(t-1) \ y(t-2) \ \dots \ y(t-n)] \in \mathbf{R}^{m \times n}.$$

于是, 从式(147) 可以得到辨识模型:

$$y(t) + \psi(t) \vartheta = \theta^T \varphi(t) + v(t). \quad (148)$$

这个模型与辨识模型(118) 类似, 因此同样可以通过定义两个中间向量, 将模型(148) 分解为两个虚拟子系统, 然后根据递阶辨识原理和梯度搜索原理, 就可以得到交互噪声干扰情形下的多变量系统的递阶迭代辨识算法.

#### 1) 递阶梯度迭代算法

$$\hat{\theta}_k = \hat{\theta}_{k-1} - \mu_k(t) \sum_{j=1}^t \psi^T(j) [y(j) + \psi(j) \hat{\theta}_{k-1} - \hat{\theta}_{k-1}^T \hat{\varphi}_k(j)], \quad (149)$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \mu_k(t) \sum_{j=1}^t \hat{\varphi}_k^T(j) [y(j) + \psi(j) \hat{\theta}_{k-1} - \hat{\theta}_{k-1}^T \hat{\varphi}_k(j)]^T, \quad (150)$$

$$\hat{\varphi}_k(t) = [u^T(t-1) \ \mu^T(t-2) \ \dots \ \mu^T(t-n) \ \hat{\rho}_{k-1}^T(t-1) \ \hat{\rho}_{k-1}^T(t-2) \ \dots \ \hat{\rho}_{k-1}^T(t-n_d)]^T, \quad (151)$$

$$\psi(t) = [y(t-1) \ y(t-2) \ \dots \ y(t-n)], \quad (152)$$

$$\hat{v}_k(t) = y(t) + \psi(t) \hat{\theta}_k - \hat{\theta}_k^T \hat{\varphi}_k(t), \quad (153)$$

$$\mu_k(t) \leq 2 \left( \sum_{j=1}^t [\|\psi(j)\|^2 + \|\hat{\varphi}_k(j)\|^2] \right)^{-1}. \quad (154)$$

#### 2) 递阶最小二乘迭代算法

$$\hat{\theta}_k = - \left[ \sum_{j=1}^t \psi^T(j) \psi(j) \right]^{-1} \sum_{j=1}^t \psi^T(j) [y(j) - \hat{\theta}_{k-1}^T \hat{\varphi}_k(j)], \quad (155)$$

$$\hat{\theta}_k = \left[ \sum_{j=1}^t \hat{\varphi}_k(j) \hat{\varphi}_k^T(j) \right]^{-1} \sum_{j=1}^t \hat{\varphi}_k(j) [y(j) + \psi_k(j) \hat{\theta}_{k-1}], \quad (156)$$

$$\hat{\varphi}_k(t) = [u^T(t-1) \ \mu^T(t-2) \ \dots \ \mu^T(t-n) \ \hat{\rho}_{k-1}^T(t-1) \ \hat{\rho}_{k-1}^T(t-2) \ \dots \ \hat{\rho}_{k-1}^T(t-n_d)]^T, \quad (157)$$

$$\hat{v}_k(t) = y(t) + \psi(t) \hat{\theta}_k - \hat{\theta}_k^T \hat{\varphi}_k(t), \quad (158)$$

$$\hat{\Psi}_k(t) = [\hat{\varphi}_k(1) \ \hat{\varphi}_k(2) \ \dots \ \hat{\varphi}_k(t)]. \quad (159)$$

交互噪声干扰下的多变量 CARMA-like 系统的递阶梯度迭代算法的计算步骤与流程图在此不再一一列出.

## 8 结语

递阶辨识是基于辨识模型分解而发展起来的一

种新型辨识方法,是为解决结构复杂、维数高、大规模系统辨识问题提出的.基本思想是通过将辨识模型的分解,使得子辨识问题的规模变小,相对于原问题要简单,从而减小辨识方法的计算量.递阶辨识原理也可以用于研究一般矩阵方程和耦合矩阵方程的递阶梯度迭代求解方法和递阶最小二乘迭代求解方法.本文讨论了线性回归模型的两阶段递阶递推最小二乘方法、递阶最小二乘辨识方法,讨论了多变量 CARMA-like 系统的递阶梯度迭代算法和递阶最小二乘迭代算法.递阶辨识原理可推广用于研究有色噪声干扰的各类多变量系统、双线性参数系统、非线性系统以及反馈非线性系统的辨识问题.

提出的基于递阶辨识原理的多变量系统递阶辨识方法,解决了国际著名专家、加拿大的 Sinha and Kwong 教授在《Automatica》(1979年,第4卷,第471—475页)中提出的存在交互干扰噪声的耦合多变量系统领域30多年的辨识难题.本文讨论的递阶辨识方法可以推广到其他多变量系统和类多变量系统<sup>[2]</sup>,以及非线性系统<sup>[14,16]</sup>.

## 参考文献

### References

- [1] 丁锋. 系统辨识(1): 辨识导引[J]. 南京信息工程大学学报: 自然科学版, 2011, 3(1): 1-22  
DING Feng. System identification. Part A: Introduction to the identification [J]. Journal of Nanjing University of Information Science & Technology: Natural Science Edition 2011, 3(1): 1-22
- [2] 丁锋. 系统辨识(2): 系统描述的基本模型[J]. 南京信息工程大学学报: 自然科学版, 2011, 3(2): 97-117  
DING Feng. System identification. Part B: Basic models for system description [J]. Journal of Nanjing University of Information Science & Technology: Natural Science Edition 2011, 3(2): 97-117
- [3] 丁锋. 系统辨识(3): 辨识精度与辨识基本问题[J]. 南京信息工程大学学报: 自然科学版, 2011, 3(3): 193-226  
DING Feng. System identification. Part C: Identification accuracy and basic problems [J]. Journal of Nanjing University of Information Science & Technology: Natural Science Edition 2011, 3(3): 193-226
- [4] 丁锋. 系统辨识(4): 辅助模型辨识思想与方法[J]. 南京信息工程大学学报: 自然科学版, 2011, 3(4): 289-318  
DING Feng. System identification. Part D: Auxiliary model identification idea and methods [J]. Journal of Nanjing University of Information Science & Technology: Natural Science Edition 2011, 3(4): 289-318
- [5] 丁锋. 系统辨识(5): 迭代搜索原理与辨识方法[J]. 南京信息工程大学学报: 自然科学版, 2011, 3(6): 481-510  
DING Feng. System identification. Part E: Iterative search principle and identification methods [J]. Journal of Nanjing University of Information Science & Technology: Natural Science Edition 2011, 3(6): 481-510
- [6] 丁锋. 系统辨识(6): 多新息辨识理论与方法[J]. 南京信息工程大学学报: 自然科学版, 2012, 4(1): 1-28  
DING Feng. System identification. Part F: Multi-innovation identification theory and methods [J]. Journal of Nanjing University of Information Science & Technology: Natural Science Edition 2012, 4(1): 1-28
- [7] 丁锋. 基于输出估计的多输入系统随机梯度估计算法[J]. 南京信息工程大学学报: 自然科学版, 2010, 2(6): 481-488  
DING Feng. Stochastic gradient estimation algorithm for multiple-input systems based on the output estimation [J]. Journal of Nanjing University of Information Science & Technology: Natural Science Edition, 2010, 2(6): 481-488
- [8] 丁锋, 杨家本. 大系统的递阶辨识[J]. 自动化学报, 1999, 25(5): 647-654  
DING Feng, YANG Jiaben. Hierarchical identification of large scale systems [J]. Acta Automatica Sinica, 1999, 25(5): 647-654
- [9] Ding F, Chen T. Hierarchical gradient-based identification of multivariable discrete-time systems [J]. Automatica, 2005, 41(2): 315-325
- [10] Ding F, Chen T. Hierarchical least squares identification methods for multivariable systems [J]. IEEE Transactions on Automatic Control, 2005, 50(3): 397-402
- [11] Wang L Y, Ding F, Liu X P. Consistency of HLS estimation algorithms for MIMO ARX-like systems [J]. Applied Mathematics and Computation, 2007, 190(2): 1081-1093
- [12] 丁锋, 萧德云. 多变量系统状态空间模型的递阶辨识[J]. 控制与决策, 2005, 20(8): 848-853, 859  
DING Feng, XIAO Deyun. Hierarchical identification of state space models for multivariable systems [J]. Control and Decision, 2005, 20(8): 848-853, 859
- [13] Ding F, Chen T. Hierarchical identification of lifted state-space models for general dual-rate systems [J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2005, 52(6): 1179-1187
- [14] Ding F, Chen T. Identification of Hammerstein nonlinear ARMAX systems [J]. Automatica, 2005, 41(9): 1479-1489
- [15] Ding F, Shi Y, Chen T. Gradient-based identification methods for Hammerstein nonlinear ARMAX models [J]. Nonlinear Dynamics, 2006, 45(1/2): 31-43
- [16] Ding F, Shi Y, Chen T. Auxiliary model based least-squares identification methods for Hammerstein output-error systems [J]. Systems & Control Letters, 2007, 56(5): 373-380
- [17] Ding F, Chen T. Adaptive digital control of Hammerstein nonlinear systems with limited output sampling [J]. SIAM Journal on Control and Optimization, 2007, 45(6): 2257-2276.
- [18] Ding F, Liu X P, Liu G. Identification methods for Hammerstein nonlinear systems [J]. Digital Signal Process-

- ing 2011 21(2): 215-238
- [19] Ding F, Chen T. A gradient based adaptive control algorithm for dual-rate systems [J]. Asian Journal of Control, 2006 8(4): 314-323
- [20] Ding F, Chen T. On iterative solutions of general coupled matrix equations [J]. SIAM Journal on Control and Optimization 2006 44(6): 2269-2284
- [21] Ding F, Chen T. Gradient based iterative algorithms for solving a class of matrix equations [J]. IEEE Transactions on Automatic Control 2005 50(8): 1216-1221
- [22] Ding F, Liu X P, Ding J. Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle [J]. Applied Mathematics and Computation 2008 197(1): 41-50
- [23] Ding F, Chen T. Iterative least squares solutions of coupled Sylvester matrix equations [J]. Systems & Control Letters 2005 54(2): 95-107
- [24] 丁锋,黎明,代冀阳. 递阶辨识原理与迭代方法族 [C]//第25届中国控制会议论文集. 哈尔滨, 2006: 418-422  
DING Feng, LI Ming, Dai Jiyang. Hierarchical identification principle and a family of iterative methods [C]// Proceedings of the 25th Chinese Control Conference. Harbin 2006: 418-422
- [25] Xie L, Ding J, Ding F. Gradient based iterative solutions for general linear matrix equations [J]. Computers Mathematics with Applications 2009 58(7): 1441-1448.
- [26] Ding J, Liu Y J, Ding F. Iterative solutions to matrix equations of the form  $A_i X B_i = F_i$  [J]. Computers & Mathematics with Applications 2010 59(11): 3500-3507
- [27] Xie L, Liu Y J, Yang H Z. Gradient based and least squares based iterative algorithms for matrix equations  $AXB + CX^T D = F$  [J]. Applied Mathematics and Computation 2010 217(5): 2191-2199
- [28] Ding J, Ding F, Liu X P, et al. Hierarchical least squares identification for linear SISO systems with dual-rate sampled-data [J]. IEEE Transactions on Automatic Control, 2011 56(11): 2677-2683
- [29] 丁锋,陈通文,萧德云. 一般双率系统状态空间模型及其递阶辨识 [J]. 自动化学报 2004 30(5): 652-663  
DING Feng, CHEN Tongwen, XIAO Deyun. State-space modeling and hierarchical identification for general dual-rate stochastic systems [J]. Acta Automatica Sinica, 2004 30(5): 652-663
- [30] 丁锋,陈通文,萧德云. 非均匀周期采样多率系统的一种递阶辨识方法 [J]. 电子学报, 2004, 32(9): 1414-1420  
DING Feng, CHEN Tongwen, XIAO Deyun. Hierarchical identification of non-uniformly periodically sampled multirate systems [J]. Acta Electronica Sinica 2004 32(9): 1414-1420
- [31] 丁锋,丁韬,杨家本,等. 衰减激励条件下递阶最小二乘辨识的均方收敛性 [J]. 控制与决策 2002 17(1): 6-10  
DING Feng, DING Tao, YANG Jiaben, et al. Mean square convergence of hierarchical least squares identification under the attenuating excitation [J]. Control and Decision 2002 17(1): 6-10
- [32] Ding Feng, Yang Jiaben, Xu Yongmao. Convergence of hierarchical stochastic gradient identification for transfer function matrix models [J]. Control Theory and Applications 2001 18(6): 949-953
- [33] 赵霞,姚郁,方强. 递阶辨识方法在转台伺服系统调试中的应用研究 [J]. 控制理论与应用, 2002, 19(2): 229-234  
ZHAO Xia, YAO Yu, FANG Qiang. The study on the hierarchical identification method in the debugging of turntable servo systems [J]. Control Theory and Applications, 2002 19(2): 229-234
- [34] 丁锋,谢新民. 系统参数和状态联合估计 [J]. 控制与决策, 1994 9(3): 223-225  
DING Feng, XIE Xinmin. Combined estimation of system parameters and states [J]. Control and Decision, 1994 9(3): 223-225
- [35] Ding F, Qiu L, Chen T. Reconstruction of continuous-time systems from their non-uniformly sampled discrete-time systems [J]. Automatica 2009 45(2): 324-332
- [36] 丁锋. 系统辨识理方法论 [M]. 北京: 电力出版社 2012  
DING Feng. System identification theory and methods [M]. Beijing: China Electric Power Press 2012
- [37] Duan H H, Jia J, Ding R F. Two-stage recursive least squares parameter estimation algorithm for output error models [J]. Mathematical and Computer Modelling, 2012 55(3/4): 1151-1159
- [38] Yao G Y, Ding R F. Two-stage least squares based iterative identification algorithm for controlled autoregressive moving average (CARMA) systems [J]. Computers & Mathematics with Applications 2012 63(5): 975-984
- [39] Xiao Y S, Zhang Y, Ding J, et al. The residual based interactive least squares algorithms and simulation studies [J]. Computers & Mathematics with Applications 2009, 58(6): 1190-1197
- [40] Wang L Y, Xie L, Wang X F. The residual based interactive stochastic gradient algorithms for controlled moving average models [J]. Applied Mathematics and Computation 2009 211(2): 442-449
- [41] Zhang Z N, Ding F, Liu X G. Hierarchical gradient based iterative parameter estimation algorithm for multivariable output error moving average systems [J]. Computers & Mathematics with Applications 2011 61(3): 672-682
- [42] Xiang L L, Xie L B, Ding R F. Hierarchical least squares algorithms for single-input multiple-output systems based on the auxiliary model [J]. Mathematical and Computer Modelling 2010 52(5/6): 918-924
- [43] Han H Q, Xie L, Ding F, et al. Hierarchical least squares based iterative identification for multivariable systems with moving average noises [J]. Mathematical and Computer Modelling 2010 51(9/10): 1213-1220
- [44] Ding F, Chen T. Combined parameter and output estimation of dual-rate systems using an auxiliary model [J]. Automatica 2004 40(10): 1739-1748

## System identification. Part G: Hierarchical identification principle and methods

DING Feng<sup>1 2 3</sup>

1 School of Internet of Things Engineering, Jiangnan University, Wuxi 214122

2 Control Science and Engineering Research Center, Jiangnan University, Wuxi 214122

3 Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122

**Abstract** Hierarchical identification is an important branch of system identification. The hierarchical identification principle is developed on the basis of the “decomposition-coordination principle” in the hierarchical control for a large-scale system. It is able to not only solve problems that the identification algorithms require heavy computational burden for a large-scale systems with many parameters and high dimensions problem, but also solve identification problems for bilinear-parameter systems, multi-linear-parameter systems and nonlinear systems with complex structures. In this paper, firstly we describe the hierarchical identification principle, the Jacobi iteration and Gauss-Seidel iteration for linear systems with a set of equations  $\mathbf{AX} = \mathbf{b}$  and give the family of iterative methods for linear equations; secondly, we study hierarchical least squares based and hierarchical gradient based iterative algorithms for general matrix equations and coupled matrix equations in the light of the Jacobi iteration and the hierarchical identification principle; thirdly, we present a two-stage recursive least squares algorithm (i. e. a simple hierarchical least squares algorithm) for equation error models and a hierarchical least squares identification algorithm for linear regression models. Finally, the hierarchical identification methods are introduced for multivariable CARMA-like systems using the hierarchical identification principle.

**Key words** iterative identification; recursive identification; parameter estimation; FIR model; equation error model; CAR model; CARMA model; CARAR model; CARARMA model; output error model; OEMA model; OEAR model; auxiliary model identification; multi-innovation identification; hierarchical identification; coupled identification