文章编号:1674-7070(2012)05-0385-17

辨识方法的计算效率(2):迭代算法

丁锋[1,2,3]

摘要

讨论了最小二乘迭代辨识算法及其计算效率问题.最小二乘迭代算法由于涉及矩阵求逆运算,为减小计算量,提出了基于块矩阵求逆的最小二乘迭代辨识算法.基于块矩阵求逆的最小二乘迭代辨识算法不是一种新算法,只是从辨识算法的实现方式上降低计算负担,它与最小二乘迭代算法产生相同的参数估计,但计算量小.文中研究了伪线性回归系统、多元伪线性回归系统、多变量伪线性回归系统的最小二乘迭代辨识算法及其基于块矩阵求逆的最小二乘迭代算法.

关键词

递推辨识;迭代辨识;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 2012-10-09
资助项目 国家自然科学基金(61273194);江苏省自然科学基金(BK2012549);高等学校学科创新引智计划(B12018)

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

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

0 引言

数学模型有很悠久的研究历史.在以数学模型为基础的控制论诞生以及自动化控制科学蓬勃发展的时代,动态系统的数学模型起到了举足轻重的作用,进而形成了研究和建立系统数学模型的理论与方法——系统辨识.数十年来,系统辨识得到了长足发展,从实验建模、辨识建模、辨识方法的提出,到辨识的基本问题,再到辨识方法的性能分析等都取得了许多光辉的研究成果[1-4].最近诞生的一些新型辨识方法,如辅助模型辨识方法[5]、迭代辨识方法[6]、多新息辨识方法[7]、递阶辨识方法[8]、耦合辨识方法[9]等都出现在控制领域国际权威期刊上[10-27].

辨识方法(估计方法)有很多类别.辨识方法的类别按其计算方式可分为一次完成算法,即直接算法(direct algorithm)、递推估计方法(递推辨识方法)和迭代估计方法(迭代辨识方法);按其实时性可分为在线估计方法(在线辨识方法)和离线估计方法(离线辨识方法);按其属性特征可分为最小二乘估计算法、最小均方估计算法、梯度估计算法、随机逼近估计算法、辅助模型辨识方法、多新息辨识方法、递阶辨识方法、耦合辨识方法、极大似然辨识方法、贝叶斯辨识方法(Bayesian identification method)等.也可分为时不变参数估计方法和时变参数估计方法,以及随机参数估计方法等[1].

辨识算法的计算量是评价计算效率的一个重要指标.文献[28]讨论了一些基本矩阵运算的计算量和典型递推辨识算法的计算量.矩阵乘法计算量与其计算次序有很大关系,本文讨论矩阵乘法、矩阵逆、块矩阵逆的计算量,仍然采用flop数,即浮点运算数表示.一次加法运算称为一个flop,一次乘法运算也称为一个flop[29].除法作为乘法对待,减法作为加法对待.

文献[28]讨论了线性回归系统、多元线性回归系统、多变量系统的随机梯度辨识算法、最小二乘辨识算法、递推最小二乘辨识算法的计算量.本文讨论最小二乘迭代算法的计算量,并利用块矩阵求逆引理来提高计算效率,提出了基于块矩阵求逆的最小二乘迭代辨识算法.基于块矩阵求逆的最小二乘迭代辨识算法不是一种新算法,只是从辨识算法的实现方式上降低计算负担,它们与最小二乘迭代算法产生相同的参数估计,但计算量小.本文针对伪线性回归系统、多元伪线性回归系统、多变量伪线性回归系统,推导了最小二

乘迭代辨识算法及其基于块矩阵求逆的最小二乘迭代算法.

1 矩阵乘法与矩阵逆的计算量

1.1 矩阵乘法

设 $\boldsymbol{A}\in\mathbf{R}^{m\times n}$, $\boldsymbol{B}\in\mathbf{R}^{n\times p}$, $\boldsymbol{C}\in\mathbf{R}^{p\times q}$, $\boldsymbol{\Lambda}\in\mathbf{R}^{n\times n}$, $\boldsymbol{H}\in\boldsymbol{R}^{L\times n}(L\gg n)$. 因为 $\boldsymbol{AB}\in\mathbf{R}^{m\times p}$ 有 mnp 次乘法运算和 $m(n-1)p$ 次加法运算, 其计算量为 $(2mnp-mp)$ flops. 也就是说, 矩阵乘法的计算量数量级为 $O(mnp)$; $\boldsymbol{\Lambda}^2$ 有 n^3 次乘法运算和 (n^3-n^2) 次加法运算, 其计算量为 $(2n-1)n^2$ flops, 计算量数量级为 $O(n^3)$. 按照这样的计算方法, 可以推算出矩阵乘法与逆的计算量, 如表 1 所示, 其中矩阵逆的计算过程见下节.

从表 1 可以看出: 矩阵乘法的计算量与计算顺序有很大关系(括号表示计算顺序), 特别是当矩阵维数有很大差异时. 对于辨识问题, 数据长度 L 一般远远大于参数的数目 n, 即 $L\gg n$, 表 1 中 $\boldsymbol{S}_1=\boldsymbol{S}_2$, $\boldsymbol{S}_1$ 的计算量是 $O(L)$, $\boldsymbol{S}_2$ 的计算量是 $O(L^2)$; 同样 $\boldsymbol{W}_1=\boldsymbol{W}_2$, $\boldsymbol{W}_1$ 的计算量是 $O(L^2)$, $\boldsymbol{W}_2$ 的计算量是 $O(L^3)$, 后者的计算量要远远大于前者. 由此可见, 辨识算法的计算量与算法的实现方式有很大关系.

1.2 矩阵逆

计算矩阵逆有多种方法. 可以通过计算矩阵 $\boldsymbol{A}\in\mathbf{R}^{n\times n}$ 的伴随矩阵(adjoint matrix) $\mathrm{adj}[\boldsymbol{A}]$, 除以其行列式 $\det[\boldsymbol{A}]:=|\boldsymbol{A}|$ 得到, 即

$$\boldsymbol{A}^{-1}=\frac{\mathrm{adj}[\boldsymbol{A}]}{\det[\boldsymbol{A}]}\in\mathbf{R}^{n\times n}.$$

但是这种方法的计算量大. 另一种方法是通过线性变换求矩阵的逆.

计算矩阵逆的一种方法是构造增广矩阵(augmented matrix) $[\boldsymbol{A}|\boldsymbol{I}]\in\mathbf{R}^{n\times(2n)}$, 然后使用高斯消元法(Gaussian elimination), 把它左半部分变换成单位阵 $\boldsymbol{I}$(假设 $\boldsymbol{A}$ 的逆存在), 那么变换后增广矩阵的右半部分就是 $\boldsymbol{A}^{-1}$, 也就是变换到最后的矩阵为 $[\boldsymbol{I}|\boldsymbol{A}^{-1}]$. 这种方法实际上是用高斯消元法求解矩阵方程 $\boldsymbol{AX}=\boldsymbol{I}_n$, 因为解为 $\boldsymbol{X}=\boldsymbol{A}^{-1}$. 假设增广矩阵 $[\boldsymbol{A}|\boldsymbol{I}]\in\mathbf{R}^{n\times(2n)}$ 有下列形式:

$$\left[\begin{array}{cccc|cccc} a_{11} & a_{12} & \cdots & a_{1n} & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2n} & 0 & 1 & & \vdots \\ \vdots & \vdots & & \vdots & \vdots & & \ddots & 0 \\ a_{n1} & a_{n2} & \cdots & a_{nn} & 0 & \cdots & 0 & 1 \end{array}\right],$$

其第 i 行记为 r_i, 第 (i,j) 元记为 a_{ij}, $i=1,2,\cdots,n$, $j=1,2,\cdots,2n$.

表 1 矩阵乘法与逆的计算量

Table 1 The computational efficiency of matrix multiplications and inverse

序号	计算次序	乘法次数	加法次数	flop 数
1	$\boldsymbol{\Lambda}^2\in\mathbf{R}^{n\times n}$	n^3	$n^2(n-1)$	$(2n-1)n^2$
2	$\boldsymbol{\Lambda}^3=\boldsymbol{\Lambda}^2\boldsymbol{\Lambda}\in\mathbf{R}^{n\times n}$	$2n^3$	$2n^2(n-1)$	$2(2n-1)n^2$
3	$\boldsymbol{\Lambda}^4=\boldsymbol{\Lambda}^3\boldsymbol{\Lambda}\in\mathbf{R}^{n\times n}$	$3n^3$	$3n^2(n-1)$	$3(2n-1)n^2$
4	$\boldsymbol{\Lambda}^4=(\boldsymbol{\Lambda}^2)(\boldsymbol{\Lambda}^2)\in\mathbf{R}^{n\times n}$	$2n^3$	$2n^2(n-1)$	$2(2n-1)n^2$
5	$\boldsymbol{\Lambda}^8=(\boldsymbol{\Lambda}^4)(\boldsymbol{\Lambda}^4)\in\mathbf{R}^{n\times n}$	$3n^3$	$3n^2(n-1)$	$3(2n-1)n^2$
6	$\boldsymbol{\Lambda}^k=(((\boldsymbol{\Lambda}^2\boldsymbol{\Lambda})\boldsymbol{\Lambda})\cdots\boldsymbol{\Lambda})\boldsymbol{\Lambda}\in\mathbf{R}^{n\times n}$	$(k-1)n^3$	$(k-1)n^2(n-1)$	$(k-1)(2n-1)n^2$
7	$\boldsymbol{AB}\in\mathbf{R}^{m\times p}$	mnp	$m(n-1)p$	$2mnp-mp$
8	$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\in\mathbf{R}^{n\times n}$	mn^2	$(m-1)n^2$	$(2m-1)n^2$
9	$\boldsymbol{AA}^{\mathrm{T}}\in\mathbf{R}^{m\times m}$	m^2n	$(n-1)m^2$	$(2n-1)m^2$
10	$(\boldsymbol{AB})\boldsymbol{C}\in\mathbf{R}^{m\times q}$	$mnp+mpq$	$m(n-1)p+m(p-1)q$	$2mp(n+q)-m(p+q)$
11	$\boldsymbol{A}(\boldsymbol{BC})\in\mathbf{R}^{m\times q}$	$mnq+npq$	$n(p-1)q+m(n-1)q$	$2(m+p)nq-(m+n)q$
12	$\boldsymbol{S}:=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\in\mathbf{R}^{n\times n}$	n^2L	$n^2(L-1)$	$2n^2L-n^2$
13	$\boldsymbol{W}:=\boldsymbol{HH}^{\mathrm{T}}\in\mathbf{R}^{L\times L}$	nL^2	$(n-1)L^2$	$(2n-1)L^2$
14	$\boldsymbol{S}_1:=(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})\in\mathbf{R}^{n\times n}$	n^2L+n^3	$n^2(L-1)+n^2(n-1)$	$2n^2L+2(n-1)n^2$
15	$\boldsymbol{S}_2:=\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{HH}^{\mathrm{T}})\boldsymbol{H}\in\mathbf{R}^{n\times n}$	$2nL^2+n^2L$	$(2n-1)L^2+(n^2-n)L-n^2$	$(4n-1)L^2+(2n^2-n)L-n^2$
16	$\boldsymbol{W}_1:=\boldsymbol{H}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})\boldsymbol{H}^{\mathrm{T}}\in\mathbf{R}^{L\times L}$	$2nL^2+n^2L$	$(2n-2)L^2+(n^2-n)L$	$(4n-2)L^2+(2n^2-n)L$
17	$\boldsymbol{W}_2:=(\boldsymbol{HH}^{\mathrm{T}})(\boldsymbol{HH}^{\mathrm{T}})\in\mathbf{R}^{L\times L}$	nL^2+L^3	$(n-1)L^2+(L-1)L^2$	$2L^3+(2n-2)L^2$
18	$\boldsymbol{\Lambda}':=\boldsymbol{\Lambda}^{-1}\in\mathbf{R}^{n\times n}$	$\frac{4}{3}n^3+\frac{2}{3}n$	$\frac{4}{3}n^3-\frac{3}{2}n^2+\frac{1}{6}n$	$\frac{8}{3}n^3-\frac{3}{2}n^2+\frac{5}{6}n$

通过线性变换，先把 $\boldsymbol{A}$ 的左边变成上三角阵，步骤如下.

1）将 a_{11} 元变为1，将 a_{i1} 元变为0（$i=2,3,\cdots,n$）. 变换如下：第1行除以 a_{11}（假设其不为零，否则进行行交换，下同），即 $r_1/a_{11}\to r_1$，其乘法运算次数为 $2n$ 次（除法作为乘法对待）；第1行乘以 $-a_{i1}$ 加到第 i 行，即 $-a_{i1}r_1+r_i\to r_i$（$i=2,3,\cdots,n$）（因为只关心矩阵逆，所以第(1,1)元下方的元不用计算为0，以节省运算次数），每行的乘法运算次数为 $2n-1$，加法运算次数为 $2n-1$（减法作为加法对待），共有 $n-1$ 行. 变换后的矩阵结构为（为了表述方便，经过变换后，增广矩阵的第 (i,j) 元依然记为 a_{ij}，* 代表可能非零的元）

$$\left[\begin{array}{cccc|cccc}1 & a_{12} & \cdots & a_{1n} & * & 0 & \cdots & 0\\ 0 & a_{22} & \cdots & a_{2n} & * & 1 & & \vdots\\ \vdots & \vdots & & \vdots & \vdots & & \ddots & 0\\ 0 & a_{n2} & \cdots & a_{nn} & * & 0 & \cdots & 1\end{array}\right].$$

2）将 a_{22} 元变为1，将 a_{i2} 元变为0（$i=3,4,\cdots,n$）. 变换如下：第2行除以 a_{22}，即 $r_2/a_{22}\to r_2$，其乘法运算次数为 $2n-1$ 次；第2行乘以 $-a_{i2}$ 加到第 i 行，即 $-a_{i2}r_2+r_i\to r_i$（$i=3,4,\cdots,n$）（同样，第(2,2)元下方的元不用计算为0），每行的乘法运算次数为 $2n-2$，加法运算次数为 $2n-2$，共有 $n-2$ 行. 变换后的矩阵结构为

$$\left[\begin{array}{ccccc|ccccc}1 & a_{12} & a_{13} & \cdots & a_{1n} & * & 0 & 0 & \cdots & 0\\ 0 & 1 & a_{23} & \cdots & a_{2n} & * & * & 0 & \cdots & 0\\ 0 & 0 & a_{33} & \cdots & a_{3n} & * & * & 1 & & \vdots\\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \ddots & 0\\ 0 & 0 & a_{n3} & \cdots & a_{nn} & * & * & 0 & \cdots & 1\end{array}\right].$$

3）$r_3/a_{33}\to r_3$，其乘法运算次数为 $2n-2$ 次；$-a_{i3}r_3+r_i\to r_i$（$i=4,5,\cdots,n$）（第(3,3)元下方的元不用计算为0），每行的乘法运算次数为 $2n-3$，加法运算次数为 $2n-3$，共有 $n-3$ 行. 变换后的矩阵结构为

$$\left[\begin{array}{ccccc|ccccc}1 & a_{12} & a_{13} & \cdots & a_{1n} & * & 0 & 0 & \cdots & 0\\ 0 & 1 & a_{23} & \cdots & a_{2n} & * & * & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & a_{3n} & * & * & 1 & & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & 0\\ 0 & 0 & 0 & \cdots & a_{nn} & * & * & * & \cdots & 1\end{array}\right].$$

4）$r_{n-1}/a_{n-1,n-1}\to r_{n-1}$，其乘法运算次数为 $n+2$ 次；$-a_{n,n-1}r_{n-1}+r_n\to r_n$，乘法运算次数为 $n+1$，加法运算次数为 $n+1$. 第 n 行各元素除以 a_{nn}，乘法运算次数为 $n+1$ 次. 变换后的矩阵结构为

$$\left[\begin{array}{ccccc|ccccc}1 & a_{12} & a_{13} & \cdots & a_{1n} & * & 0 & 0 & \cdots & 0\\ 0 & 1 & a_{23} & \cdots & a_{2n} & * & * & 0 & \cdots & 0\\ 0 & 0 & \ddots & & \vdots & \vdots & \vdots & \ddots & & \vdots\\ \vdots & \vdots & & 1 & \vdots & * & * & \cdots & * & 0\\ 0 & 0 & \cdots & 0 & 1 & * & * & \cdots & * & *\end{array}\right].\quad(1)$$

至此，增广矩阵的左边变成了单位上三角阵，加法运算次数为

$$\begin{aligned}&1\cdot(n+1)+2\cdot(n+2)+\cdots+\\&(n-2)(2n-2)+(n-1)(2n-1)=\\&1\cdot(n+1)+2\cdot(n+2)+\cdots+\\&(n-2)(n+n-2)+(n-1)(n+n-1)=\\&1\cdot n+1^2+2\cdot n+2^2+\cdots+(n-2)\cdot n+\\&(n-2)^2+(n-1)\cdot n+(n-1)^2=\\&\frac{n(n-1)}{2}\cdot n+\frac{n(n-1)(2n-1)}{6}=\\&\frac{5}{6}n^3-n^2+\frac{1}{6}n,\end{aligned}\quad(2)$$

乘法运算次数为

$$\begin{aligned}&\left(\frac{5}{6}n^3-n^2+\frac{1}{6}n\right)+(n+1)+(n+2)+\cdots+\\&(2n-1)+(2n)=\frac{5}{6}n^3-n^2+\frac{1}{6}n+n^2+\\&\frac{n(n+1)}{2}=\frac{5}{6}n^3+\frac{1}{2}n^2+\frac{2}{3}n.\end{aligned}\quad(3)$$

再把增广矩阵(1)的左边变成单位矩阵，步骤如下.

1）对前 $n-1$ 行进行线性变换：$-a_{jn}r_n+r_j\to r_j$（$j=n-1,n-2,\cdots,2,1$），因为只关心增广矩阵右边矩阵，左边的运算次数不计入（即不必计算），故乘法运算次数为 n，加法运算次数为 n，共 $n-1$ 行. 变换后的矩阵为

$$\left[\begin{array}{ccccc|cccc}1 & a_{12} & a_{13} & \cdots & 0 & * & * & \cdots & *\\ 0 & 1 & a_{23} & \cdots & 0 & * & * & \cdots & *\\ 0 & 0 & \ddots & & \vdots & \vdots & \vdots & & \vdots\\ \vdots & \vdots & & 1 & 0 & * & * & \cdots & *\\ 0 & 0 & \cdots & 0 & 1 & * & * & \cdots & *\end{array}\right].$$

2）对前 $n-2$ 行进行线性变换：$-a_{j,n-1}r_{n-1}+r_j\to r_j$（$j=n-2,n-1,\cdots,2,1$），乘法运算次数为 n，加法运算次数为 n，共 $n-2$ 行.

3）以此类推，对前2行进行线性变换：$-a_{j,3}r_3+r_j\to r_j$（$j=2,1$），乘法运算次数为 n，加法运算次数为 n，共2行. 变换矩阵为

$$\left[\begin{array}{ccccc|cccc} 1 & a_{12} & 0 & \cdots & 0 & * & * & \cdots & * \\ 0 & 1 & 0 & \cdots & 0 & * & * & \cdots & * \\ 0 & 0 & 1 & & \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \ddots & 0 & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 1 & * & * & \cdots & * \end{array}\right].$$

4) 最后对第 1 行进行变换: $-a_{12}r_2+r_1\to r_1$, 乘法运算次数为 n, 加法运算次数为 n, 共 1 行. 变换后的矩阵为

$$\left[\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & * & * & \cdots & * \\ 0 & 1 & & \vdots & * & * & \cdots & * \\ \vdots & & \ddots & 0 & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 1 & * & * & \cdots & * \end{array}\right].$$

增广矩阵左边从上三角阵变换为单位阵, 加法运算次数为

$$n+2n+3n+\cdots+(n-2)n+(n-1)n=\frac{n^2(n-1)}{2}, \tag{4}$$

乘法运算次数为

$$\frac{n^2(n-1)}{2}. \tag{5}$$

因此, 式(2)与式(4)的次数相加, 得到矩阵求逆运算的加法次数:

$$\left(\frac{5}{6}n^3-n^2+\frac{1}{6}n\right)+\frac{n^2(n-1)}{2}=\frac{4}{3}n^3-\frac{3}{2}n^2+\frac{1}{6}n, \tag{6}$$

式(3)与式(5)的次数相加, 得到矩阵求逆运算的乘法次数:

$$\left(\frac{5}{6}n^3+\frac{1}{2}n^2+\frac{2}{3}n\right)+\frac{n^2(n-1)}{2}=\frac{4}{3}n^3+\frac{2}{3}n. \tag{7}$$

两式相加得到矩阵求逆运算的总 flop 数:

$$\left(\frac{4}{3}n^3-\frac{3}{2}n^2+\frac{1}{6}n\right)+\left(\frac{4}{3}n^3+\frac{2}{3}n\right)=\frac{8}{3}n^3-\frac{3}{2}n^2+\frac{5}{6}n. \tag{8}$$

下面的 Matlab 程序使用高斯消元法计算矩阵 $\boldsymbol{A}$ 的逆.

```
%————————————————————————————————————————*
% Filename: matrix_inversion.m *
% Compute the inverse of matrix A *
%————————————————————————————————————————*
  clear; format short g
  n = 4; m = 2 * n;
%——Generate matrix A
  rand('state',1);
% a = rand(n,n);
  a = pascal(n);
  a1 = [a, eye(n)];
  fprintf('----Compute matrix inverse')
  for k = 1:n
     a1(k,:) = a1(k,:)/a1(k,k);
     for i = k + 1:n
        a1(i,:) = -a1(i,k) * a1(k,:) + a1(i,:);
     end
  end
  Uppertriangularmatrix = a1
  for k = n - 1: -1:1
     for j = k: -1:1
        a1(j,:) = -a1(j,k + 1) * a1(k + 1,:) + a1(j,:);
   end
  end
  Invmatrix = a1(:,n + 1:m)
  I2 = a * a1(:,n + 1:m);
```

接下来的 Matlab 程序是具有最小运算次数的高斯消元法计算矩阵 $\boldsymbol{A}$ 的逆, 其忽略了增广矩阵左边消除为零的运算, 因为不关心左边单位阵的运算, 程序运行最后的增广矩阵右边是 $\boldsymbol{A}$ 的逆, 右边不是单位阵, 这是与上面程序不同的地方.

```
%————————————————————————————————————————*
% Filename: matrix_inversion.m *
% Compute the inverse of matrix A *
%————————————————————————————————————————*
  clear; format short g
  n = 4; m = 2 * n;
%——Generate matrix A
  rand('state',1);
% a = rand(n,n);
  a = pascal(n);
  a1 = [a, eye(n)];
  fprintf('Compute matrix inverse with minimimun flops')
  for k = 1:n
     a1(k,k:m) = a1(k,k:m)/a1(k,k);
     for i = k + 1:n
        a1(i,k + 1:m) = -a1(i,k) * a1(k,k + 1:m) + a1(i,k + 1:m);
   end
  end
  Uppertriangularmatrix1 = a1
  for k = n - 1: -1:1
   for j = k: -1:1
        a1(j,n + 1:m) = -a1(j,k + 1) * a1(k + 1,n + 1:m) + a1(j,n + 1:m);
```

```
    end
  end
  Invmatrix1 = a1(:,n+1:m)
  I2 = a * a1(:,n+1:m);
```

1.3 块矩阵求逆的计算量

使用块矩阵求逆引理可以降低最小二乘迭代算法的计算量,下面不加证明地给出该引理.

引理 1 (块矩阵求逆引理)(Block matrix inversion lemma)假设矩阵 $\boldsymbol{A}\in\mathbf{R}^{m\times m}$ 和 $\boldsymbol{Q}:=\boldsymbol{D}-\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}\in\mathbf{R}^{n\times n}$ 是可逆矩阵,那么下列块矩阵逆关系成立:

$$\begin{bmatrix}\boldsymbol{A} & \boldsymbol{B}\\ \boldsymbol{C} & \boldsymbol{D}\end{bmatrix}^{-1}=\begin{bmatrix}\boldsymbol{A}^{-1}+\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{Q}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & -\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{Q}^{-1}\\ -\boldsymbol{Q}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{Q}^{-1}\end{bmatrix}. \quad (9)$$

证明比较简单,这里从略.

利用式(8)矩阵求逆的计算量公式,可知式(9)左边矩阵求逆的计算量的总 flop 数为

$$N_1:=\frac{8}{3}(m+n)^3-\frac{3}{2}(m+n)^2+\frac{5}{6}(m+n).$$

式(9)右边各子矩阵的乘法运算次数和加法运算次数如表 2 所示(在计算运算量时,括号里面的矩阵相乘作为一个整体对待),其总 flop 数为

$$N_2:=\frac{8}{3}(m^3+n^3)+6m^2n+6mn^2-\frac{5}{2}m^2-4mn-\frac{1}{2}n^2+\frac{5}{6}(m+n).$$

它们之差为

$$\begin{aligned}N_1-N_2={}&\frac{8}{3}(m+n)^3-\frac{3}{2}(m+n)^2+\frac{5}{6}(m+n)-\\&\left[\frac{8}{3}(m^3+n^3)+6m^2n+6mn^2-\frac{5}{2}m^2-4mn-\frac{1}{2}n^2+\frac{5}{6}(m+n)\right]=\\&\frac{8}{3}(3m^2n+3mn^2)-(6m^2n+6mn^2)-\frac{3}{2}(m+n)^2+\frac{5}{2}m^2+4mn+\frac{1}{2}n^2=\\&2m^2n+2mn^2+m^2+mn-n^2=\\&m^2(2n+1)+(2m-1)n^2+mn>0.\end{aligned}$$

因此,直接矩阵求逆的计算量比使用块矩阵求逆的计算量大. 例如当 $m=7$, $n=3$ 时, $N_1=2\ 529$ flops, $N_2=2\ 044$ flops, $N_1-N_2=481$ flops;当 $m=10$, $n=10$ 时, $N_1=20\ 750$ flops, $N_2=16\ 650$ flops, $N_1-N_2=4\ 100$ flops.

2 伪线性回归系统

为方便起见,设 $\{u(t)\}$ 为系统输入序列, $\{y(t)\}$ 为系统观测输出序列, $\{v(t)\}$ 为零均值随机白噪声序列, z^{-1} 为单位后移算子: $z^{-1}y(t)=y(t-1)$ 或 $zy(t)=y(t+1)$, $a(z)$, $b(z)$, $c(z)$, $d(z)$ 和 $f(z)$ 是算子 z^{-1} 的常系数时不变多项式,定义如下:

$$a(z):=1+a_1z^{-1}+a_2z^{-2}+\cdots+a_{n_a}z^{-n_a},\ a_i\in\mathbf{R},$$
$$b(z):=b_1z^{-1}+b_2z^{-2}+\cdots+b_{n_b}z^{-n_b},\ b_i\in\mathbf{R},$$
$$c(z):=1+c_1z^{-1}+c_2z^{-2}+\cdots+c_{n_c}z^{-n_c},\ c_i\in\mathbf{R},$$

表 2 块矩阵求逆的计算量

Table 2 The computational efficiency of computing the block matrix inverse

变量	计算次序	乘法次数	加法次数
$\boldsymbol{A}_1$	$\boldsymbol{A}_1:=\boldsymbol{A}'-(\boldsymbol{A}'\boldsymbol{B})\boldsymbol{C}_1\in\mathbf{R}^{m\times m}$	m^2n	$m^2n-m^2+n^2$
$\boldsymbol{B}_1$	$\boldsymbol{B}_1:=(\boldsymbol{A}'\boldsymbol{B})\boldsymbol{Q}'\in\mathbf{R}^{m\times n}$	mn^2	mn^2-mn
$\boldsymbol{C}_1$	$\boldsymbol{C}_1:=\boldsymbol{Q}'(\boldsymbol{C}\boldsymbol{A}')\in\mathbf{R}^{n\times m}$	mn^2	mn^2-mn
$\boldsymbol{Q}'$	$\boldsymbol{Q}':=\boldsymbol{Q}^{-1}\in\mathbf{R}^{n\times n}$	$\frac{4}{3}n^3+\frac{2}{3}n$	$\frac{4}{3}n^3-\frac{3}{2}n^2+\frac{1}{6}n$
$\boldsymbol{A}'$	$\boldsymbol{A}':=\boldsymbol{A}^{-1}\in\mathbf{R}^{m\times m}$	$\frac{4}{3}m^3+\frac{2}{3}m$	$\frac{4}{3}m^3-\frac{3}{2}m^2+\frac{1}{6}m$
$\boldsymbol{Q}$	$\boldsymbol{Q}=\boldsymbol{D}-(\boldsymbol{C}\boldsymbol{A}')\boldsymbol{B}\in\mathbf{R}^{n\times n}$	mn^2	mn^2
	$\boldsymbol{A}'\boldsymbol{B}\in\mathbf{R}^{m\times n}$	m^2n	m^2n-mn
	$\boldsymbol{C}\boldsymbol{A}'\in\mathbf{R}^{n\times m}$	m^2n	m^2n-mn
总数		$\frac{4}{3}(m^3+n^3)+3m^2n+3mn^2+\frac{2}{3}(m+n)$	$\frac{4}{3}(m^3+n^3)+3m^2n+3mn^2-\frac{5}{2}m^2-4mn-\frac{1}{2}n^2+\frac{1}{6}(m+n)$
总 flop 数		$\frac{8}{3}(m^3+n^3)+6m^2n+6mn^2-\frac{5}{2}m^2-4mn-\frac{1}{2}n^2+\frac{5}{6}(m+n)$	

$$d(z):=1+d_1z^{-1}+d_2z^{-2}+\cdots+d_{n_d}z^{-n_d},d_i\in\mathbf{R},$$

$$f(z):=1+f_1z^{-1}+f_2z^{-2}+\cdots+f_{n_f}z^{-n_f},f_i\in\mathbf{R}.$$

多项式系数(coefficient)a_i,b_i,c_i,d_i 和 f_i 为模型参数. 假设阶次(order)n_a,n_b,n_c,n_d 和 n_f 已知. 根据移位算子的性质,有

$$a(z)y(t)=(1+a_1z^{-1}+a_2z^{-2}+\cdots+a_{n_a}z^{-n_a})y(t)=$$
$$y(t)+a_1y(t-1)+a_2y(t-2)+\cdots+a_{n_a}y(t-n_a),$$
$$b(z)u(t)=(b_1z^{-1}+b_2z^{-2}+\cdots+b_{n_b}z^{-n_b})u(t)=$$
$$b_1u(t-1)+b_2u(t-2)+\cdots+b_{n_b}u(t-n_b),$$
$$d(z)v(t)=(1+d_1z^{-1}+d_2z^{-2}+\cdots+d_{n_d}z^{-n_d})v(t)=$$
$$v(t)+d_1v(t-1)+d_2v(t-2)+\cdots+d_{n_d}v(t-n_d),\text{等.}$$

2.1 递推增广最小二乘算法

考虑下列伪线性回归模型(Pseudo-Linear Regression model,PLR 模型):

$$y(t)=\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}+v(t),\tag{10}$$

$$\boldsymbol{\varphi}(t)=[\boldsymbol{\phi}^{\mathrm{T}}(t),\boldsymbol{\psi}^{\mathrm{T}}(t)]^{\mathrm{T}}\in\mathbf{R}^n,\tag{11}$$

其中$\{y(t)\}$是观测输出序列,$\{v(t)\}$是零均值不相关随机噪声序列,$\boldsymbol{\theta}\in\mathbf{R}^n$ 是要估计的参数向量,$\boldsymbol{\varphi}(t)\in\mathbf{R}^{n_1}$是由观测输入输出数据$\{u(t-i),y(t-i)\}$构成的线性或非线性回归信息向量,$\boldsymbol{\psi}(t)\in\mathbf{R}^{n_2}$是由不可测噪声$\{v(t-i)\}$构成的线性或非线性回归信息向量,它们可以表示为

$$\boldsymbol{\phi}(t):=\boldsymbol{\phi}(y(t-1),y(t-2),\cdots,y(t-n_a),$$
$$u(t-1),u(t-2),\cdots,u(t-n_b))\in\mathbf{R}^{n_1},$$
$$\boldsymbol{\psi}(t):=\boldsymbol{\psi}(v(t-1),v(t-2),\cdots,$$
$$v(t-n_d))\in\mathbf{R}^{n_2},$$

伪线性回归模型包括一大类线性或非线性、标量或多变量滑动平均系统,如(多输入)线性或非线性受控滑动平均系统、(多输入)受控自回归滑动平均系统等.

设 $\boldsymbol{I}$ 代表适当维数的单位阵,其对角元均为 1,其余元均为 0. $\boldsymbol{I}_n$ 代表 n 阶单位阵 $\boldsymbol{I}_n\in\mathbf{R}^{n\times n}$. 下列递推增广最小二乘算法(Recursive Extended Least Squares algorithm,RELS 算法)可以估计伪线性回归系统(10)的参数向量 $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{\theta}}(t)=\hat{\boldsymbol{\theta}}(t-1)+\boldsymbol{L}(t)[y(t)-\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t-1)],\tag{12}$$

$$\boldsymbol{L}(t)=\boldsymbol{P}(t-1)\hat{\boldsymbol{\varphi}}(t)[1+\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\boldsymbol{P}(t-1)\hat{\boldsymbol{\varphi}}(t)]^{-1},\tag{13}$$

$$\boldsymbol{P}(t)=[\boldsymbol{I}-\boldsymbol{L}(t)\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)]\boldsymbol{P}(t-1),\boldsymbol{P}(0)=p_0\boldsymbol{I},\tag{14}$$

$$\hat{\boldsymbol{\varphi}}(t)=[\boldsymbol{\phi}^{\mathrm{T}}(t),\hat{\boldsymbol{\psi}}^{\mathrm{T}}(t)]^{\mathrm{T}},\tag{15}$$

$$\boldsymbol{\phi}(t)=\boldsymbol{\phi}(y(t-1),y(t-2),\cdots,y(t-n_a),$$
$$u(t-1),u(t-2),\cdots,u(t-n_b)),\tag{16}$$

$$\hat{\boldsymbol{\psi}}(t)=\boldsymbol{\psi}(\hat{v}(t-1),\hat{v}(t-2),\cdots,\hat{v}(t-n_d)),\tag{17}$$

$$\hat{v}(t)=y(t)-\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t).\tag{18}$$

但是在相同数据长度下,最小二乘迭代算法比 RELS 算法能产生更高的参数估计精度,下面讨论最小二乘迭代辨识方法.

2.2 最小二乘迭代算法

设 L 为数据长度($L\gg n$),定义堆积输出向量 $\boldsymbol{Y}$ 和堆积信息矩阵 $\boldsymbol{H}$ 如下:

$$\boldsymbol{Y}:=\begin{bmatrix}y(1)\\y(2)\\\vdots\\y(L)\end{bmatrix}\in\mathbf{R}^L,\quad\boldsymbol{H}:=\begin{bmatrix}\boldsymbol{\varphi}^{\mathrm{T}}(1)\\\boldsymbol{\varphi}^{\mathrm{T}}(2)\\\vdots\\\boldsymbol{\varphi}^{\mathrm{T}}(L)\end{bmatrix}\in\mathbf{R}^{L\times n}.$$

定义矩阵 $\boldsymbol{X}$ 的范数 $\|\boldsymbol{X}\|^2:=\mathrm{tr}[\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}]$ 和二次准则函数:

$$J_1(\boldsymbol{\theta}):=\|\boldsymbol{Y}-\boldsymbol{H}\boldsymbol{\theta}\|^2.$$

假设信息向量 $\boldsymbol{\varphi}(t)$ 是持续激励的,即矩阵 $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$ 可逆,极小化 $J_1(\boldsymbol{\theta})$ 给出参数向量 $\boldsymbol{\theta}$ 的最小二乘估计:

$$\hat{\boldsymbol{\theta}}=[\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}]^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{Y}.\tag{19}$$

使用式(11)中 $\boldsymbol{\varphi}(t)$ 的定义,矩阵 $\boldsymbol{H}$ 可以表示为

$$\boldsymbol{H}=\begin{bmatrix}\boldsymbol{\phi}^{\mathrm{T}}(1)&\boldsymbol{\psi}^{\mathrm{T}}(1)\\\boldsymbol{\phi}^{\mathrm{T}}(2)&\boldsymbol{\psi}^{\mathrm{T}}(2)\\\vdots&\vdots\\\boldsymbol{\phi}^{\mathrm{T}}(L)&\boldsymbol{\psi}^{\mathrm{T}}(L)\end{bmatrix}\in\mathbf{R}^{L\times n}.$$

因为矩阵 $\boldsymbol{H}$ 中的 $\boldsymbol{\psi}(t)$ 是未知的($t=1,2,\cdots,L$),所以不可能通过式(19)计算参数向量 $\boldsymbol{\theta}$ 的估计 $\hat{\boldsymbol{\theta}}$. 解决的办法是使用递阶辨识原理[19-20]. 令 $k=1,2,3,\cdots$为迭代变量,$\hat{\boldsymbol{\theta}}_k$ 为第 k 次迭代 $\boldsymbol{\theta}$ 的估计,向量 $\boldsymbol{\psi}(t)$ 中未知噪声项 $v(t-i)$ 用其前一次迭代(即($k-1$)次迭代)估计 $\hat{v}_{k-1}(t-i)$ 代替,定义第 k 次迭代 $\boldsymbol{\varphi}(t)$ 和 $\boldsymbol{\psi}(t)$ 的估计为

$$\hat{\boldsymbol{\varphi}}_k(t):=[\boldsymbol{\phi}^{\mathrm{T}}(t),\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t)]^{\mathrm{T}}\in\mathbf{R}^n,$$
$$\hat{\boldsymbol{\psi}}_k(t):=\boldsymbol{\psi}(\hat{v}_{k-1}(t-1),\hat{v}_{k-1}(t-2),\cdots,$$
$$\hat{v}_{k-1}(t-n_d))\in\mathbf{R}^{n_2}.$$

从式(10)可得 $v(t)=y(t)-\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}$. 用 $\hat{\boldsymbol{\varphi}}_k(t)$ 和 $\hat{\boldsymbol{\theta}}_k$ 代替 $\boldsymbol{\varphi}(t)$ 和 $\boldsymbol{\theta}$,就给出 $v(t)$ 的估计:

$$\hat{v}_k(t)=y(t)-\hat{\boldsymbol{\varphi}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_k.$$

定义估计的信息矩阵:

$$\hat{\boldsymbol{H}}_k:=\begin{bmatrix}\hat{\boldsymbol{\varphi}}_k^{\mathrm{T}}(1)\\\hat{\boldsymbol{\varphi}}_k^{\mathrm{T}}(2)\\\vdots\\\hat{\boldsymbol{\varphi}}_k^{\mathrm{T}}(L)\end{bmatrix}=\begin{bmatrix}\boldsymbol{\phi}^{\mathrm{T}}(1)&\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(1)\\\boldsymbol{\phi}^{\mathrm{T}}(2)&\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(2)\\\vdots&\vdots\\\boldsymbol{\phi}^{\mathrm{T}}(L)&\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(L)\end{bmatrix}=$$
$$[\boldsymbol{\Phi},\hat{\boldsymbol{\Psi}}_k]\in\mathbf{R}^{L\times n},\tag{20}$$

其中

$$\boldsymbol{\Phi}:=\begin{bmatrix}\boldsymbol{\phi}^{\mathrm{T}}(1)\\ \boldsymbol{\phi}^{\mathrm{T}}(2)\\ \vdots\\ \boldsymbol{\phi}^{\mathrm{T}}(L)\end{bmatrix}\in\mathbf{R}^{L\times n_1},\quad \hat{\boldsymbol{\Psi}}_k:=\begin{bmatrix}\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(1)\\ \hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(2)\\ \vdots\\ \hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(L)\end{bmatrix}\in\mathbf{R}^{L\times n_2}.$$

定义数据乘积矩阵(data product moment matrix)：

$$\boldsymbol{S}_k:=\hat{\boldsymbol{H}}_k^{\mathrm{T}}\hat{\boldsymbol{H}}_k=\begin{bmatrix}\boldsymbol{\Phi}^{\mathrm{T}}\\ \hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\end{bmatrix}[\boldsymbol{\Phi},\hat{\boldsymbol{\Psi}}_k]=\begin{bmatrix}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} & \boldsymbol{\Phi}^{\mathrm{T}}\hat{\Psi}_k\\ \hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\Phi} & \hat{\Psi}_k^{\mathrm{T}}\hat{\Psi}_k\end{bmatrix}=$$

$$\begin{bmatrix}\boldsymbol{S} & \boldsymbol{\Phi}^{\mathrm{T}}\hat{\Psi}_k\\ \hat{\Psi}_k^{\mathrm{T}}\boldsymbol{\Phi} & \hat{\Psi}_k^{\mathrm{T}}\hat{\Psi}_k\end{bmatrix}\in\mathbf{R}^{(n_1+n_2)\times(n_1+n_2)},\qquad(21)$$

其中 $\boldsymbol{S}:=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\in\mathbf{R}^{n_1\times n_1}$.

用 $\hat{\boldsymbol{H}}_k$ 代替式(19)中 $\boldsymbol{H}$，并为参数估计向量 $\hat{\boldsymbol{\theta}}$ 加上下标 k，表示第 k 次迭代的参数估计 $\hat{\boldsymbol{\theta}}_k$，可以总结出估计伪线性回归系统(10)的最小二乘迭代辨识方法(Least Squares based Iterative identification algorithm，LSI 算法)：

$$\hat{\boldsymbol{\theta}}_k=[\hat{\boldsymbol{H}}_k^{\mathrm{T}}\hat{\boldsymbol{H}}_k]^{-1}\hat{\boldsymbol{H}}_k^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{S}_k^{-1}\hat{\boldsymbol{H}}_k^{\mathrm{T}}\boldsymbol{Y},\quad k=1,2,3,\cdots\qquad(22)$$

$$\boldsymbol{Y}=[y(1),y(2),\cdots,y(L)]^{\mathrm{T}},\qquad(23)$$

$$\hat{\boldsymbol{H}}_k=[\hat{\boldsymbol{\varphi}}_k(1),\hat{\boldsymbol{\varphi}}_k(2),\cdots,\hat{\boldsymbol{\varphi}}_k(L)]^{\mathrm{T}},\qquad(24)$$

$$\hat{\boldsymbol{\varphi}}_k(t)=[\boldsymbol{\phi}^{\mathrm{T}}(t),\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t)]^{\mathrm{T}},\qquad(25)$$

$$\boldsymbol{\phi}(t)=\boldsymbol{\phi}(y(t-1),y(t-2),\cdots,y(t-n_a),\; u(t-1),u(t-2),\cdots,u(t-n_b)),\qquad(26)$$

$$\hat{\boldsymbol{\psi}}_k(t)=\psi(\hat{v}_{k-1}(t-1),\hat{v}_{k-1}(t-2),\cdots,\; \hat{v}_{k-1}(t-n_d)),\qquad(27)$$

$$\hat{v}_k(t)=y(t)-\hat{\boldsymbol{\varphi}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_k,\quad t=1,2,\cdots,L.\qquad(28)$$

最小二乘迭代算法(22)—(28)每迭代计算一步的计算量如表 3 所示，其中乘法次数为 $\frac{4}{3}n^3+n^2+\frac{2}{3}n+(n^2+2n)L$，加法次数为 $\frac{4}{3}n^3-\frac{3}{2}n^2-\frac{11}{6}n+(n^2+2n)L$，总 flop 数为 $\frac{8}{3}n^3-\frac{1}{2}n^2-\frac{7}{6}n+2n(n+2)L$.

2.3 基于块矩阵求逆的最小二乘迭代算法

LSI 算法(22)—(28)在每一次迭代过程中，都要计算 $(n_1+n_2)\times(n_1+n_2)$ 维大矩阵 $\boldsymbol{S}_k=[\hat{\boldsymbol{H}}_k^{\mathrm{T}}\hat{\boldsymbol{H}}_k]$ 的逆，故计算量大. 观察式(21)中分块矩阵 $\boldsymbol{S}_k\in\mathbf{R}^{(n_1+n_2)\times(n_1+n_2)}$ 的结构，可以利用分块矩阵求逆引理来减小算法的计算量. 因为 $\boldsymbol{S}_k$ 是对称阵，故块矩阵求逆计算量会更小. 而且 $\boldsymbol{S}_k$ 的第(1,1)子块矩阵 $\boldsymbol{S}$ 不依赖迭代变量 k，只需在开始迭代计算前计算一次 $\boldsymbol{S}^{-1}$，所以基于块矩阵求逆的最小二乘迭代算法的计算量还要小.

应用引理 1 于式(21)给出

$$\boldsymbol{S}_k^{-1}=\begin{bmatrix}\boldsymbol{S} & \boldsymbol{\Phi}^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k\\ \hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\Phi} & \hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k\end{bmatrix}^{-1}=$$

$$\begin{bmatrix}\boldsymbol{S}^{-1}+\boldsymbol{S}^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k)\boldsymbol{Q}_k^{-1}(\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\Phi})\boldsymbol{S}^{-1} & -\boldsymbol{S}^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k)\boldsymbol{Q}_k^{-1}\\ -\boldsymbol{Q}_k^{-1}(\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\Phi})\boldsymbol{S}^{-1} & \boldsymbol{Q}_k^{-1}\end{bmatrix},\qquad(29)$$

$$\boldsymbol{Q}_k:=\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k-(\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\Phi})\boldsymbol{S}^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k)=\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}[\boldsymbol{I}-\boldsymbol{\Phi}\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}]\hat{\boldsymbol{\Psi}}_k.\qquad(30)$$

使用式(20)和(28)，从式(22)，有

$$\hat{\boldsymbol{\theta}}_k=\boldsymbol{S}^{-1}\hat{\boldsymbol{H}}_k^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{S}_k^{-1}\begin{bmatrix}\boldsymbol{\Phi}^{\mathrm{T}}\\ \hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\end{bmatrix}\boldsymbol{Y}=\boldsymbol{S}_k^{-1}\begin{bmatrix}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\ \hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{Y}\end{bmatrix}=$$

$$\begin{bmatrix}\boldsymbol{S}^{-1}+\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{S}^{-1} & -\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k\boldsymbol{Q}_k^{-1}\\ -\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{S}^{-1} & \boldsymbol{Q}_k^{-1}\end{bmatrix}\times\begin{bmatrix}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\ \hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{Y}\end{bmatrix}.$$

表 3 最小二乘迭代算法的计算量

Table 3 The computational efficiency of the least squares based iterative algorithm

变量	计算次序	乘法次数	加法次数
$\hat{\boldsymbol{\theta}}_k$	$\boldsymbol{\beta}_k:=\hat{\boldsymbol{H}}_k^{\mathrm{T}}\boldsymbol{Y}\in\mathbf{R}^n$	nL	$n(L-1)$
	$\boldsymbol{S}_k:=\hat{\boldsymbol{H}}_k^{\mathrm{T}}\hat{\boldsymbol{H}}_k\in\mathbf{R}^{n\times n}$	$n^2Ln^2(L-1)$	
	$\boldsymbol{S}'_k:=\boldsymbol{S}_k^{-1}\in\mathbf{R}^{n\times n}$	$\frac{4}{3}n^3+\frac{2}{3}n$	$\frac{4}{3}n^3-\frac{3}{2}n^2+\frac{1}{6}n$
	$\hat{\boldsymbol{\theta}}_k=\boldsymbol{S}'_k\boldsymbol{\beta}_k\in\mathbf{R}^n$	n^2	$n(n-1)$
$\hat{v}_k(t)$	$\hat{v}_k(t)=y(t)-\hat{\boldsymbol{\varphi}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_k\in\mathbf{R}$	nL	nL
总数		$\frac{4}{3}n^3+n^2+\frac{2}{3}n+(n^2+2n)L$	$\frac{4}{3}n^3-\frac{3}{2}n^2-\frac{11}{6}n+(n^2+2n)L$
总 flop 数		$\frac{8}{3}n^3-\frac{1}{2}n^2-\frac{7}{6}n+2n(n+2)L$	

再结合式(23)—(27),可以得到计算量小的基于块矩阵求逆的最小二乘迭代算法(多余的括号表示计算次序):

$$\hat{\boldsymbol{\theta}}_k=\begin{bmatrix}\boldsymbol{S}^{-1}+(\boldsymbol{R}_k\boldsymbol{Q}_k^{-1})\boldsymbol{R}_k^{\mathrm{T}} & -(\boldsymbol{R}_k\boldsymbol{Q}_k^{-1})\\ -(\boldsymbol{R}_k\boldsymbol{Q}_k^{-1})^{\mathrm{T}}] & \boldsymbol{Q}_k^{-1}\end{bmatrix}\begin{bmatrix}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y})\\ (\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{Y})\end{bmatrix},\tag{31}$$

$$\boldsymbol{Q}_k=\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k-(\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\Phi})\boldsymbol{S}^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k)=\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k-(\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\Phi})\boldsymbol{R}_k,\tag{32}$$

$$\boldsymbol{R}_k=\boldsymbol{S}^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k),\tag{33}$$

$$\boldsymbol{Y}=[y(1),y(2),\cdots,y(L)]^{\mathrm{T}},\tag{34}$$

$$\hat{\boldsymbol{H}}_k=[\hat{\boldsymbol{\varphi}}_k(1),\hat{\boldsymbol{\varphi}}_k(2),\cdots,\hat{\boldsymbol{\varphi}}_k(L)]^{\mathrm{T}},\tag{35}$$

$$\hat{\boldsymbol{\varphi}}_k(t)=[\boldsymbol{\phi}^{\mathrm{T}}(t),\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t)]^{\mathrm{T}},$$

$$\boldsymbol{\phi}(t)=\boldsymbol{\phi}(y(t-1),y(t-2),\cdots,y(t-n_a),u(t-1),u(t-2),\cdots,u(t-n_b)),\tag{36}$$

$$\hat{\boldsymbol{\psi}}_k(t)=\boldsymbol{\psi}(\hat{v}_{k-1}(t-1),\hat{v}_{k-1}(t-2),\cdots,\hat{v}_{k-1}(t-n_d)),\tag{37}$$

$$\hat{v}_k(t)=y(t)-\hat{\boldsymbol{\varphi}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_k,\quad t=1,2,\cdots,L.\tag{38}$$

读者可以写出这个算法的实现步骤,绘出流程图,列出实现该算法的乘法次数和加法次数. 尽管算法(31)—(38)比算法(22)—(28)形式复杂,但是计算量小.

定义常数矩阵和向量:

$$\boldsymbol{R}:=\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\in\mathbf{R}^{n_1\times L},\quad \boldsymbol{\alpha}:=\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{R}\boldsymbol{Y}\in\mathbf{R}^{n_1},$$

$$\boldsymbol{\beta}:=\boldsymbol{\Phi}\boldsymbol{\alpha}\in\mathbf{R}^{L},\quad \boldsymbol{M}:=\boldsymbol{I}-\boldsymbol{\Phi}\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\in\mathbf{R}^{L\times L}.$$

它们不依赖于迭代变量 k,只需在迭代循环前计算一次即可.

由式(31)有

$$\hat{\boldsymbol{\theta}}_k=\begin{bmatrix}\boldsymbol{\alpha}+\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\alpha}-\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{Y}\\ -\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\alpha}+\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{Y}\end{bmatrix}=$$

$$\begin{bmatrix}\boldsymbol{\alpha}+\boldsymbol{R}\hat{\boldsymbol{\Psi}}_k\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\beta}-\boldsymbol{R}\hat{\boldsymbol{\Psi}}_k\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}Y)\\ -\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\beta}+\boldsymbol{Q}_k^{-1}Y)\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{Y}\end{bmatrix}=$$

$$\begin{bmatrix}\boldsymbol{\alpha}+\boldsymbol{R}\hat{\boldsymbol{\Psi}}_k\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{Y})\\ -\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{Y})\end{bmatrix}.$$

于是,可以得到基于块矩阵求逆的最小二乘迭代算法的等价形式:

$$\hat{\boldsymbol{\theta}}_k=\begin{bmatrix}\boldsymbol{\alpha}+\boldsymbol{R}\hat{\boldsymbol{\Psi}}_k\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{Y})\\ -\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{Y})\end{bmatrix},\quad k=1,2,3,\cdots,\tag{39}$$

$$\hat{\boldsymbol{\Psi}}_k=[\hat{\boldsymbol{\psi}}_k(1),\hat{\boldsymbol{\psi}}_k(2),\cdots,\hat{\boldsymbol{\psi}}_k(L)]^{\mathrm{T}},\tag{40}$$

$$\boldsymbol{Q}_k=\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{M}\hat{\boldsymbol{\Psi}}_k=\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k-(\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{\Phi})\boldsymbol{S}^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\hat{\boldsymbol{\Psi}}_k),\tag{41}$$

$$\hat{\boldsymbol{\varphi}}_k(t)=[\boldsymbol{\phi}^{\mathrm{T}}(t),\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t)]^{\mathrm{T}},t=1,2,\cdots,L,\tag{42}$$

$$\hat{\boldsymbol{\psi}}_k(t)=\boldsymbol{\psi}(\hat{v}_{k-1}(t-1),\hat{v}_{k-1}(t-2),\cdots,\hat{v}_{k-1}(t-n_d)),\tag{43}$$

$$\hat{v}_k(t)=y(t)-\hat{\boldsymbol{\varphi}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_k,\tag{44}$$

$$\boldsymbol{\phi}(t)=\boldsymbol{\phi}(y(t-1),y(t-2),\cdots,y(t-n_a),u(t-1),u(t-2),\cdots,u(t-n_b)),\tag{45}$$

$$\boldsymbol{Y}=[y(1),y(2),\cdots,y(L)]^{\mathrm{T}},\tag{46}$$

$$\boldsymbol{\Phi}=[\boldsymbol{\phi}(1),\boldsymbol{\phi}(2),\cdots,\boldsymbol{\phi}(L)]^{\mathrm{T}},\tag{47}$$

$$\boldsymbol{S}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi},\quad \boldsymbol{R}=\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}},\quad \boldsymbol{M}=\boldsymbol{I}-\boldsymbol{\Phi}\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}},\quad \boldsymbol{\alpha}=\boldsymbol{R}\boldsymbol{Y},\quad \boldsymbol{\beta}=\boldsymbol{\Phi}\boldsymbol{\alpha}.\tag{48}$$

基于块矩阵求逆的最小二乘迭代算法(39)—(48)计算参数估计向量 $\hat{\boldsymbol{\theta}}_k$ 的步骤如下.

1）收集输入输出数据 $\{u(t),y(t):t=1,2,3,\cdots,L\}$.

2）初始化:令 $k=1$, $\hat{v}_0(t)$ 是一个随机数,给定小正数 ε.

3）用式(46)构成 $\boldsymbol{Y}$,用式(45)和(47)构成 $\boldsymbol{\phi}(t)$ 和 $\boldsymbol{\Phi}$.

4）用式(48)计算 $\boldsymbol{S},\boldsymbol{R},\boldsymbol{M},\boldsymbol{\alpha}$ 和 $\boldsymbol{\beta}$.

5）用式(43)构成 $\hat{\boldsymbol{\psi}}_k(t)$,用式(40)构成 $\hat{\boldsymbol{\Psi}}_k$,用式(41)计算 $\boldsymbol{Q}_k$.

6）用式(39)刷新参数估计向量 $\hat{\boldsymbol{\theta}}_k$,用式(44)计算 $\hat{v}_k(t)$.

7）如果 $\|\hat{\boldsymbol{\theta}}_k-\hat{\boldsymbol{\theta}}_{k-1}\|\leqslant\varepsilon$,那么中断迭代循环,获得迭代次数 k 和参数估计向量 $\hat{\boldsymbol{\theta}}_k$;否则 k 增 1,返回到第 5 步.

基于块矩阵求逆的最小二乘迭代算法(39)—(48)计算参数估计向量 $\hat{\boldsymbol{\theta}}_k$ 的流程如图 1 所示.

按照上述计算步骤,算法的乘法运算次数和加法运算次数如表 4—5 所示. 下面的解释似乎有道理,却大大增加了计算量. 注意到 $\boldsymbol{S},\boldsymbol{R},\boldsymbol{M},\boldsymbol{\alpha}$ 和 $\boldsymbol{\beta}$ 是不依赖于迭代变量 k 的常数矩阵或向量,只需在迭代循环开始前计算一次,所以计算量小. 但如果真是这样,计算量反而很大. 例如,尽管 $\boldsymbol{M}$ 是一个常数矩阵,只需在迭代循环外计算一次,但是 $\boldsymbol{M}=\boldsymbol{I}-\boldsymbol{\Phi}\boldsymbol{S}'\boldsymbol{\Phi}^{\mathrm{T}}\in\mathbf{R}^{L\times L}$ 的乘法运算次数 $n_1^2L+n_1L^2$,加法次数为 $(n_1^2-n_1+1)L+(n_1-1)L^2$,计算量是 $O(L^2)$;$\boldsymbol{Q}_k=\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{M}\hat{\boldsymbol{\Psi}}_k\in\mathbf{R}^{n_2\times n_2}$ 的乘法运算次数为 $n_2L^2+n_2^2L$,加法运算次数为 $n_2(L-1)L+n_2^2(L-1)$,计算量也是 $O(L^2)$. 计算量比表 3 最小二乘迭代算法的计算量 $O(L)$ 还要大得多. 原因是矩阵乘法的计算量大小与乘法次序关系很大. 读者可以找出计算量小的计算步骤和计算流程.

伪线性回归系统覆盖了伪线性回归模型Ⅰ、伪线性回归模型Ⅱ、伪线性回归模型Ⅲ. 它们是线性参数模型,可以是线性系统,也可以是非线性系统;可

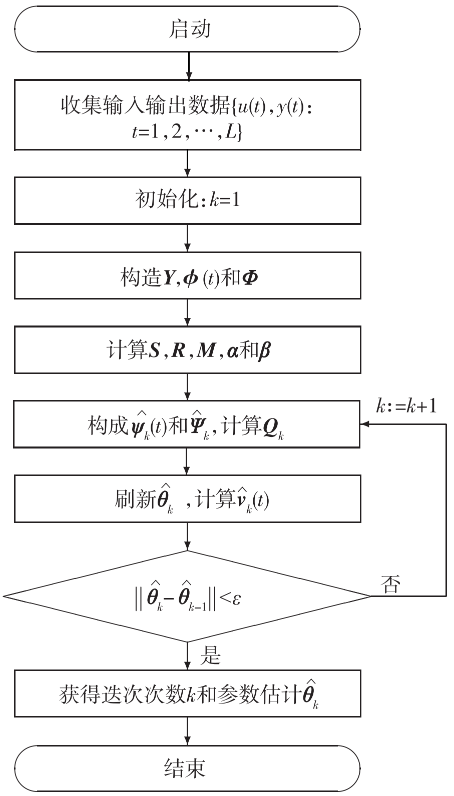

图1 计算参数估计向量 $\hat{\boldsymbol{\theta}}_k$ 的流程

Fig. 1 The flowchart of computing the parameter estimation vector $\hat{\boldsymbol{\theta}}_k$

以是标量系统,也可以是多变量系统. 前述的基于块矩阵求逆的最小二乘迭代辨识算法适用于线性系统的 CARMA 模型、CARAR 模型、CARARMA 模型、输出误差模型、OEMA 模型、OEAR 模型、Box-Jenkins 模型及其特殊情形. 例如,当信息向量为

$$\boldsymbol{\varphi}(t)=[u(t-1),u(t-2),\cdots,u(t-n_b)]^{\mathrm{T}}\in\mathbf{R}^{n_b},$$

$$\boldsymbol{\psi}(t)=[v(t-1),v(t-2),\cdots,v(t-n_d)]^{\mathrm{T}}\in\mathbf{R}^{n_d}$$

时,伪线性回归模型(10)退化为下列受控滑动平均模型(Controlled Moving Average model, CMA 模型),即有限脉冲响应滑动平均模型(Finite Impulse Response Moving Average model, FIR-MA 模型):

$$y(t)=b(z)u(t)+d(z)v(t). \tag{49}$$

因此,算法(39)—(48)退化为文献[30]中的基于信息矩阵分解的受控滑动平均系统迭代最小二乘估计算法.

当信息向量为

$$\boldsymbol{\varphi}(t)=[-y(t-1),-y(t-2),u(t-1),u^2(t-1),\ u(t-2)u(t-3),u(t-4)]^{\mathrm{T}}\in\mathbf{R}^6,$$

$$\boldsymbol{\psi}(t)=[v(t-1),v(t-2)v(t-3),v(t-3)]^{\mathrm{T}}\in\mathbf{R}^3,$$

伪线性回归模型(10)表示下列非线性系统:

$$y(t)+\theta_1y(t-1)+\theta_2y(t-2)=\theta_3u(t-1)+\ \theta_4u^2(t-1)+\theta_5u(t-2)u(t-3)+\theta_6u(t-4)+\ v(t)+\theta_7v(t-1)+\theta_8v(t-2)v(t-3)+\theta_9v(t-3).$$

只需将算法(39)—(48)的式(43)修改为

$$\hat{\boldsymbol{\psi}}_k(t)=[\hat{v}_{k-1}(t-1),\hat{v}_{k-1}(t-2)\hat{v}_{k-1}(t-3),\ \hat{v}_{k-1}(t-3)]^{\mathrm{T}}\in\mathbf{R}^3,$$

就可用于这个非线性系统的辨识. 基于块矩阵求逆的最小二乘迭代算法具有高的计算效率,适用于信息向量含有未知变量的系统参数辨识.

3 多元伪线性回归系统

定义移位算子 z 的矩阵多项式:

$$\boldsymbol{A}(z):=\boldsymbol{I}+\boldsymbol{A}_1z^{-1}+\boldsymbol{A}_2z^{-2}+\cdots+\boldsymbol{A}_{n_a}z^{-n_a},\boldsymbol{A}_i\in\mathbf{R}^{m\times m},$$

$$\boldsymbol{B}(z):=\boldsymbol{B}_1z^{-1}+\boldsymbol{B}_2z^{-2}+\cdots+\boldsymbol{B}_{n_b}z^{-n_b},\boldsymbol{B}_i\in\mathbf{R}^{m\times r},$$

表4 基于块矩阵求逆的最小二乘迭代算法的计算量

Table 4 The computational efficiency of the LSI algorithm based on the block matrix inversion

变量	计算次序	乘法次数	加法次数
$\hat{\boldsymbol{\theta}}_k$	$\boldsymbol{\lambda}_k:=\boldsymbol{R}\hat{\boldsymbol{\Psi}}_k\boldsymbol{\gamma}_k\in\mathbf{R}^{n_1}$	$n_1n_2(L+1)$	$n_1n_2L-n_1$
	$\boldsymbol{\gamma}_k:=\boldsymbol{Q}'_k\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{Y})\in\mathbf{R}^{n_2}$	$n_2L+n_1n_2$	$(n_2+1)L+n_2^2-2n_2$
	$\boldsymbol{Q}'_k:=\boldsymbol{Q}_k^{-1}\in\mathbf{R}^{n_2\times n_2}$	$\frac{4}{3}n_2^3+\frac{2}{3}n_2$	$\frac{4}{3}n_2^3-\frac{3}{2}n_2^2+\frac{1}{6}n_2$
	$\hat{\boldsymbol{\theta}}_k=\begin{bmatrix}\boldsymbol{\alpha}+\boldsymbol{\lambda}_k\\-\boldsymbol{\gamma}_k\end{bmatrix}\in\mathbf{R}^n$	0	n_1
$\boldsymbol{Q}_k$	$\boldsymbol{Q}_k=\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{M}\hat{\boldsymbol{\Psi}}_k\in\mathbf{R}^{n_2\times n_2}$	$n_2L^2+n_2^2L$	$n_2(L-1)L+n_2^2(L-1)$
$\hat{v}_k(t)$	$\hat{v}_k(t)=y(t)-\hat{\boldsymbol{\varphi}}_k^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}_k\in\mathbf{R}$	nL	nL
总数		$\frac{4}{3}n_2^3+\frac{2}{3}n_2+2n_1n_2+n_2L^2+(n_2^2+n_1n_2+n_2+n)L$	$\frac{4}{3}n_2^3-\frac{3}{2}n_2^2-\frac{11}{6}n_2+n_2L^2+(n_2^2+n_1n_2+n+1)$
总 flop 数		$\frac{8}{3}n_2^3-\frac{3}{2}n_2^2-\frac{7}{6}n_2+2n_1n_2+2n_2L^2+(2n_2^2+2n_1n_2+n_2+2n+1)L$	

表5 常数矩阵向量运算的计算量

Table 5 The computational efficiency of the constant matrices and vectors

变量	计算次序	乘法次数	加法次数
$\boldsymbol{S}$	$\boldsymbol{S}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\in\mathbf{R}^{n_1\times n_1}$	n_1^2L	$n_1^2(L-1)$
$\boldsymbol{S}'$	$\boldsymbol{S}':=\boldsymbol{S}^{-1}\in\mathbf{R}^{n_1\times n_1}$	$\frac{4}{3}n_1^3+\frac{2}{3}n_1$	$\frac{4}{3}n_1^3-\frac{3}{2}n_1^2+\frac{1}{6}n_1$
$\boldsymbol{R}$	$\boldsymbol{R}=\boldsymbol{S}'\boldsymbol{\Phi}^{\mathrm{T}}\in\mathbf{R}^{n_1\times L}$	n_1^2L	$n_1(n_1-1)L$
$\boldsymbol{M}$	$\boldsymbol{M}=\boldsymbol{I}-\boldsymbol{\Phi}\boldsymbol{S}'\boldsymbol{\Phi}^{\mathrm{T}}\in\mathbf{R}^{L\times L}$	$n_1^2L+n_1L^2$	$(n_1^2-n_1+1)L+(n_1-1)L^2$
$\boldsymbol{\alpha}$	$\boldsymbol{\alpha}=\boldsymbol{R}\boldsymbol{Y}\in\mathbf{R}^{n_1}$	n_1L	$n_1(L-1)$
$\boldsymbol{\beta}$	$\boldsymbol{\beta}=\boldsymbol{\Phi}\boldsymbol{\alpha}\in\mathbf{R}^{L}$	n_1L	$(n_1-1)L$
总数		$\frac{4}{3}n_1^3+\frac{2}{3}n_1+(3n_1^2+2n_1)L+n_1L^2$	$\frac{4}{3}n_1^3-\frac{5}{2}n_1^2-\frac{5}{6}n_1+3n_1^2L+(n_1-1)L^2$
总 flop 数		$\frac{8}{3}n_1^3-\frac{5}{2}n_1^2-\frac{1}{6}n_1+(6n_1^2+2n_1)L+(2n_1-1)L^2$	

$$\boldsymbol{C}(z):=\boldsymbol{I}+\boldsymbol{C}_1z^{-1}+\boldsymbol{C}_2z^{-2}+\cdots+\boldsymbol{C}_{n_c}z^{-n_c},\boldsymbol{C}_i\in\mathbf{R}^{m\times m},$$
$$\boldsymbol{D}(z):=\boldsymbol{I}+\boldsymbol{D}_1z^{-1}+\boldsymbol{D}_2z^{-2}+\cdots+\boldsymbol{D}_{n_d}z^{-n_d},\boldsymbol{D}_i\in\mathbf{R}^{m\times m},$$
$$\boldsymbol{F}(z):=\boldsymbol{I}+\boldsymbol{F}_1z^{-1}+\boldsymbol{F}_2z^{-2}+\cdots+\boldsymbol{F}_{n_f}z^{-n_f},\boldsymbol{F}_i\in\mathbf{R}^{m\times m}.$$

假设阶次(order)n_a,n_b,n_c,n_d 和 n_f 已知,移位算子 z^{-1} 多项式矩阵 $\boldsymbol{A}(z),\boldsymbol{B}(z),\boldsymbol{C}(z),\boldsymbol{D}(z)$ 和 $\boldsymbol{F}(z)$ 的系数矩阵(coefficient matrix)$\boldsymbol{A}_i\in\mathbf{R}^{m\times m},\boldsymbol{B}_i\in\mathbf{R}^{m\times r},\boldsymbol{C}_i\in\mathbf{R}^{m\times m},\boldsymbol{D}_i\in\mathbf{R}^{m\times m}$ 和 $\boldsymbol{F}_i\in\mathbf{R}^{m\times n}$ 是待辨识的系统参数矩阵.

3.1 多元伪线性滑动平均系统

考虑下列多元伪线性滑动平均系统(multivariate pseudo-linear regression system),

$$\boldsymbol{y}(t)=\boldsymbol{\Phi}(t)\boldsymbol{\theta}+d(z)\boldsymbol{v}(t),\tag{50}$$

其中 $\boldsymbol{y}(t)=[y_1(t),y_2(t),\cdots,y_m(t)]^{\mathrm{T}}\in\mathbf{R}^m$ 为 m 维系统输出向量,$\boldsymbol{\Phi}(t)\in\mathbf{R}^{m\times n_1}$ 是由系统输入输出数据构成的回归信息矩阵(通常 $n_1>m$),$\boldsymbol{\theta}\in\mathbf{R}^{n_1}$ 是待辨识的系统模型参数向量,$\boldsymbol{v}(t)=[v_1(t),v_2(t),\cdots,v_m(t)]^{\mathrm{T}}\in\mathbf{R}^m$ 是零均值白噪声向量. 假设 $t\leqslant 0$ 时,$\boldsymbol{y}(t)=\mathbf{0},\boldsymbol{\Phi}(t)=\mathbf{0}$ 和 $\boldsymbol{v}(t)=\mathbf{0}$.

定义参数向量 $\boldsymbol{\vartheta}$ 和信息矩阵 $\boldsymbol{\Gamma}(t)$ 如下:

$$\boldsymbol{\vartheta}:=[\boldsymbol{\theta}^{\mathrm{T}},d_1,d_2,\cdots,d_{n_d}]^{\mathrm{T}}\in\mathbf{R}^n,n:=n_1+n_2,$$
$$\boldsymbol{\Gamma}(t):=[\boldsymbol{\Phi}(t),\boldsymbol{\Psi}(t)]\in\mathbf{R}^{m\times n},\tag{51}$$
$$\boldsymbol{\Psi}(t):=[\boldsymbol{v}(t-1),\boldsymbol{v}(t-2),\cdots,\boldsymbol{v}(t-n_d)]\in\mathbf{R}^{m\times n_2},$$
$$n_2:=n_d.$$

式(50)可以写为

$$\boldsymbol{y}(t)=\boldsymbol{\Gamma}(t)\boldsymbol{\vartheta}+\boldsymbol{v}(t).\tag{52}$$

1)最小二乘迭代算法

比较式(10)—(11)与式(51)—(52),模仿上节算法的推导方法,定义和极小化准则函数[31]:

$$J_2(\boldsymbol{\vartheta}):=\sum_{t=1}^{L}\|\boldsymbol{y}(t)-\boldsymbol{\Gamma}(t)\boldsymbol{\vartheta}\|^2,$$

可以得到估计多元线性回归滑动平均系统的最小二乘迭代辨识算法:

$$\hat{\boldsymbol{\vartheta}}_k=[\hat{\boldsymbol{H}}_k^{\mathrm{T}}\hat{\boldsymbol{H}}_k]^{-1}\hat{\boldsymbol{H}}_k^{\mathrm{T}}\boldsymbol{Y},\quad k=1,2,3,\cdots,\tag{53}$$
$$\boldsymbol{Y}=[\boldsymbol{y}^{\mathrm{T}}(1),\boldsymbol{y}^{\mathrm{T}}(2),\cdots,\boldsymbol{y}^{\mathrm{T}}(L)]^{\mathrm{T}},\tag{54}$$
$$\hat{\boldsymbol{H}}_k=\begin{bmatrix}\boldsymbol{\Phi}(1) & \hat{\boldsymbol{\Psi}}_k(1)\\ \boldsymbol{\Phi}(2) & \hat{\boldsymbol{\Psi}}_k(2)\\ \vdots & \vdots\\ \boldsymbol{\Phi}(L) & \hat{\boldsymbol{\Psi}}_k(L)\end{bmatrix},\tag{55}$$
$$\hat{\boldsymbol{\Gamma}}_k(t)=[\boldsymbol{\Phi}(t),\hat{\boldsymbol{\Psi}}_k(t)],\tag{56}$$
$$\hat{\boldsymbol{\Psi}}_k(t)=[\hat{\boldsymbol{v}}_{k-1}(t-1),\hat{\boldsymbol{v}}_{k-1}(t-2),\cdots,\hat{\boldsymbol{v}}_{k-1}(t-n_d)],\tag{57}$$
$$\hat{\boldsymbol{v}}_k(t)=\boldsymbol{y}(t)-\hat{\boldsymbol{\Gamma}}_k(t)\hat{\boldsymbol{\vartheta}}_k,\quad t=1,2,\cdots,L.\tag{58}$$

2)基于块矩阵求逆的最小二乘迭代算法

可以总结出基于块矩阵求逆的最小二乘迭代算法:

$$\hat{\boldsymbol{\vartheta}}_k=\begin{bmatrix}\boldsymbol{\alpha}+\boldsymbol{R}\hat{\boldsymbol{\Psi}}_k\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{Y})\\ -\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{Y})\end{bmatrix},\quad k=1,2,3,\cdots,\tag{59}$$
$$\hat{\boldsymbol{\Psi}}_k=[\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(1),\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(2),\cdots,\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(L)]^{\mathrm{T}},\tag{60}$$
$$\boldsymbol{Q}_k=\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{M}\hat{\boldsymbol{\Psi}}_k,\tag{61}$$
$$\hat{\boldsymbol{\Gamma}}_k(t)=[\boldsymbol{\Phi}(t),\hat{\boldsymbol{\Psi}}_k(t)],\quad t=1,2,\cdots,L,\tag{62}$$
$$\hat{\boldsymbol{\Psi}}_k(t)=[\hat{\boldsymbol{v}}_{k-1}(t-1),\hat{\boldsymbol{v}}_{k-1}(t-2),\cdots,\hat{\boldsymbol{v}}_{k-1}(t-n_d)],\tag{63}$$
$$\hat{\boldsymbol{v}}_k(t)=\boldsymbol{y}(t)-\hat{\boldsymbol{\Gamma}}_k(t)\hat{\boldsymbol{\vartheta}}_k,\tag{64}$$
$$\boldsymbol{Y}=[\boldsymbol{y}^{\mathrm{T}}(1),\boldsymbol{y}^{\mathrm{T}}(2),\cdots,\boldsymbol{y}^{\mathrm{T}}(L)]^{\mathrm{T}},\tag{65}$$
$$\boldsymbol{\Phi}=[\boldsymbol{\Phi}^{\mathrm{T}}(1),\boldsymbol{\Phi}^{\mathrm{T}}(2),\cdots,\boldsymbol{\Phi}^{\mathrm{T}}(L)]^{\mathrm{T}},\tag{66}$$
$$\boldsymbol{S}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi},\quad \boldsymbol{R}=\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}},\quad \boldsymbol{M}=\boldsymbol{I}-\boldsymbol{\Phi}\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}},$$
$$\boldsymbol{\alpha}=\boldsymbol{R}\boldsymbol{Y},\quad \boldsymbol{\beta}=\boldsymbol{\Phi}\boldsymbol{\alpha}.\tag{67}$$

3.2 交互干扰噪声多元伪线性滑动平均系统

考虑交互干扰噪声多元伪线性滑动平均系统(multivariate pseudo-linear regression system with interactive noise):

$$\boldsymbol{y}(t)=\boldsymbol{\Phi}(t)\boldsymbol{\theta}+\boldsymbol{D}(z)\boldsymbol{v}(t). \tag{68}$$

定义参数向量 $\boldsymbol{\theta}_2$ 和信息向量 $\boldsymbol{\psi}(t)$ 如下:

$$\boldsymbol{\theta}_2^{\mathrm{T}}:=[\boldsymbol{D}_1,\boldsymbol{D}_2,\cdots,\boldsymbol{D}_{n_d}]\in\mathbf{R}^{m\times(mn_d)},$$

$$\boldsymbol{\psi}(t):=[\boldsymbol{v}^{\mathrm{T}}(t-1),\boldsymbol{v}^{\mathrm{T}}(t-2),\cdots,\boldsymbol{v}^{\mathrm{T}}(t-n_d)]^{\mathrm{T}}\in\mathbf{R}^{mn_d}.$$

设 col[$\boldsymbol{X}$]表示将矩阵 $\boldsymbol{X}$ 的列按次序排成的向量,如 $\boldsymbol{X}:=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]\in\mathbf{R}^{m\times n}$, $\boldsymbol{x}_i\in\mathbf{R}^m$, $i=1,2,\cdots,n$, 那么

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

若 $\boldsymbol{A}=[a_{ij}]\in\mathbf{R}^{m\times n}$, $\boldsymbol{B}=[b_{ij}]\in\mathbf{R}^{p\times q}$, 则 Kronecker 积定义为 $\boldsymbol{A}\otimes\boldsymbol{B}=[a_{ij}\boldsymbol{B}]\in\mathbf{R}^{(mp)\times(nq)}$.

式(68)可以等价写为

$$\begin{aligned}&\boldsymbol{y}(t)=\boldsymbol{\Phi}(t)\boldsymbol{\theta}+\boldsymbol{D}(z)\boldsymbol{v}(t)=\\&\boldsymbol{\Phi}(t)\boldsymbol{\theta}+\boldsymbol{v}(t)+\boldsymbol{D}_1\boldsymbol{v}(t-1)+\boldsymbol{D}_2\boldsymbol{v}(t-2)+\cdots+\boldsymbol{D}_{n_d}\boldsymbol{v}(t-n_d)=\\&\boldsymbol{\Phi}(t)\boldsymbol{\theta}+\boldsymbol{\theta}_2^{\mathrm{T}}\boldsymbol{\psi}(t)+\boldsymbol{v}(t)=\\&\boldsymbol{\Phi}(t)\boldsymbol{\theta}+[\boldsymbol{\psi}^{\mathrm{T}}(t)\otimes\boldsymbol{I}_m]\mathrm{col}[\boldsymbol{\theta}_2]+\boldsymbol{v}(t)=\\&[\boldsymbol{\Phi}(t),\boldsymbol{\psi}^{\mathrm{T}}(t)\otimes\boldsymbol{I}_m]\begin{bmatrix}\boldsymbol{\theta}\\ \mathrm{col}[\boldsymbol{\theta}_2]\end{bmatrix}+\boldsymbol{v}(t).\end{aligned} \tag{69}$$

式(69)与式(52)具有类似的形式,因此算法(53)—(58)和算法(59)—(67)可以应用,只需将式(57)和式(63)改为

$$\hat{\boldsymbol{\Psi}}_k(t)=\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t)\otimes\boldsymbol{I}_m,$$

$$\hat{\boldsymbol{\psi}}_k(t)=[\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-1),\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-2),\cdots,\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-n_d)]^{\mathrm{T}}.$$

3.3 多元伪线性自回归滑动平均系统

考虑多元伪线性自回归滑动平均系统(multivariate pseudo-linear autoregressive moving average system):

$$\boldsymbol{y}(t)=\boldsymbol{\Phi}(t)\boldsymbol{\theta}+\boldsymbol{C}^{-1}(z)\boldsymbol{D}(z)\boldsymbol{v}(t). \tag{70}$$

定义参数向量 $\boldsymbol{\theta}_2$ 和信息向量 $\boldsymbol{\psi}(t)$ 如下:

$$\boldsymbol{\theta}_2^{\mathrm{T}}:=[\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_{n_c},\boldsymbol{D}_1,\boldsymbol{D}_2,\cdots,\boldsymbol{D}_{n_d}]\in\mathbf{R}^{m\times(mn_c+mn_d)},$$

$$\begin{aligned}\boldsymbol{\psi}(t):=[&-\boldsymbol{w}^{\mathrm{T}}(t-1),-\boldsymbol{w}^{\mathrm{T}}(t-2),\cdots,-\boldsymbol{w}^{\mathrm{T}}(t-n_d),\\&\boldsymbol{v}^{\mathrm{T}}(t-1),\boldsymbol{v}^{\mathrm{T}}(t-2),\cdots,\boldsymbol{v}^{\mathrm{T}}(t-n_d)]^{\mathrm{T}}\in\mathbf{R}^{mn_c+mn_d}.\end{aligned}$$

令

$$\boldsymbol{w}(t):=\boldsymbol{C}^{-1}(z)\boldsymbol{D}(z)\boldsymbol{v}(t)\in\mathbf{R}^m, \tag{71}$$

或

$$\begin{aligned}\boldsymbol{w}(t)&=[\boldsymbol{I}-\boldsymbol{C}(z)]\boldsymbol{w}(t)+\boldsymbol{D}(z)\boldsymbol{v}(t)=\\&\boldsymbol{\theta}_2^{\mathrm{T}}\boldsymbol{\psi}(t)+\boldsymbol{v}(t).\end{aligned} \tag{72}$$

令 $\hat{\boldsymbol{\theta}}(t)$ 和 $\hat{\boldsymbol{\theta}}_2(t)$ 是 $\boldsymbol{\theta}$ 和 $\boldsymbol{\theta}_2$ 在时刻 t 的估计. 由式(70)—(72)可得

$$\begin{aligned}&\boldsymbol{y}(t)=\boldsymbol{\Phi}(t)\boldsymbol{\theta}+\boldsymbol{w}(t)=\\&\boldsymbol{\Phi}(t)\boldsymbol{\theta}+\boldsymbol{\theta}_2^{\mathrm{T}}\boldsymbol{\psi}(t)+\boldsymbol{v}(t)=\\&\boldsymbol{\Phi}(t)\boldsymbol{\theta}+[\boldsymbol{\psi}^{\mathrm{T}}(t)\otimes\boldsymbol{I}_m]\mathrm{col}[\boldsymbol{\theta}_2]+\boldsymbol{v}(t)=\\&[\boldsymbol{\Phi}(t),\boldsymbol{\psi}^{\mathrm{T}}(t)\otimes\boldsymbol{I}_m]\begin{bmatrix}\boldsymbol{\theta}\\ \mathrm{col}[\boldsymbol{\theta}_2]\end{bmatrix}+\boldsymbol{v}(t)=\\&\boldsymbol{\Gamma}(t)\boldsymbol{\vartheta}+\boldsymbol{v}(t),\end{aligned} \tag{73}$$

式(73)中

$$\boldsymbol{\Gamma}(t):=[\boldsymbol{\Phi}(t),\boldsymbol{\Psi}(t)]\in\mathbf{R}^{m\times n},$$

$$n:=n_1+m^2(n_c+n_d),$$

$$\boldsymbol{\Psi}(t):=\boldsymbol{\psi}^{\mathrm{T}}(t)\otimes\boldsymbol{I}_m\in\mathbf{R}^{m\times m^2(n_c+n_d)},$$

$$\boldsymbol{\vartheta}:=\begin{bmatrix}\boldsymbol{\theta}\\ \mathrm{col}[\boldsymbol{\theta}_2]\end{bmatrix}\in\mathbf{R}^{m^2(n_c+n_d)}.$$

1)最小二乘迭代算法

令 $\hat{\boldsymbol{\vartheta}}_k:=\begin{bmatrix}\boldsymbol{\theta}_k\\ \mathrm{col}[\hat{\boldsymbol{\theta}}_{2,k}]\end{bmatrix}$ 是 $\boldsymbol{\vartheta}=\begin{bmatrix}\boldsymbol{\theta}\\ \mathrm{col}[\boldsymbol{\theta}_2]\end{bmatrix}$ 在第 k 次迭代的估计. 式(73)与式(52)具有类似的形式,因此不难得到估计多元线性自回归滑动平均系统(70)的最小二乘迭代辨识算法:

$$\hat{\boldsymbol{\vartheta}}_k=[\hat{\boldsymbol{H}}_k^{\mathrm{T}}\hat{\boldsymbol{H}}_k]^{-1}\hat{\boldsymbol{H}}_k^{\mathrm{T}}\boldsymbol{Y},\quad k=1,2,3,\cdots, \tag{74}$$

$$\boldsymbol{Y}=[\boldsymbol{y}^{\mathrm{T}}(1),\boldsymbol{y}^{\mathrm{T}}(2),\cdots,\boldsymbol{y}^{\mathrm{T}}(L)]^{\mathrm{T}}, \tag{75}$$

$$\hat{\boldsymbol{H}}_k=\begin{bmatrix}\boldsymbol{\Phi}(1)&\hat{\boldsymbol{\Psi}}_k(1)\\ \boldsymbol{\Phi}(2)&\hat{\boldsymbol{\Psi}}_k(2)\\ \vdots&\vdots\\ \boldsymbol{\Phi}(L)&\hat{\boldsymbol{\Psi}}_k(L)\end{bmatrix}, \tag{76}$$

$$\hat{\boldsymbol{\Gamma}}_k(t)=[\boldsymbol{\Phi}(t),\hat{\boldsymbol{\Psi}}_k(t)], \tag{77}$$

$$\hat{\boldsymbol{\Psi}}_k(t)=\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t)\otimes\boldsymbol{I}_m, \tag{78}$$

$$\begin{aligned}\hat{\boldsymbol{\psi}}_k(t)=[&-\hat{\boldsymbol{w}}_{k-1}^{\mathrm{T}}(t-1),-\hat{\boldsymbol{w}}_{k-1}^{\mathrm{T}}(t-2),\cdots,-\hat{\boldsymbol{w}}_{k-1}^{\mathrm{T}}(t-n_c),\\&\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-1),\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-2),\cdots,\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-n_d)]^{\mathrm{T}},\end{aligned} \tag{79}$$

$$\hat{\boldsymbol{w}}_k(t)=\boldsymbol{y}(t)-\boldsymbol{\Phi}(t)\hat{\boldsymbol{\theta}}_k, \tag{80}$$

$$\hat{\boldsymbol{v}}_k(t)=\boldsymbol{y}(t)-\hat{\boldsymbol{\Gamma}}_k(t)\hat{\boldsymbol{\vartheta}}_k,\quad t=1,2,\cdots,L. \tag{81}$$

2)基于块矩阵求逆的最小二乘迭代算法

多元线性自回归滑动平均系统(70)的基于块矩阵求逆的最小二乘迭代算法如下:

$$\hat{\boldsymbol{\vartheta}}_k=\begin{bmatrix}\boldsymbol{\alpha}+\boldsymbol{R}\hat{\boldsymbol{\Psi}}_k\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{Y})\\ -\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{Y})\end{bmatrix},\quad k=1,2,3,\cdots, \tag{82}$$

$$\hat{\boldsymbol{\Psi}}_k=[\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(1),\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(2),\cdots,\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(L)]^{\mathrm{T}}, \tag{83}$$

$$\boldsymbol{Q}_k=\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{M}\hat{\boldsymbol{\Psi}}_k, \tag{84}$$

$$\hat{\boldsymbol{\Gamma}}_k(t)=[\boldsymbol{\Phi}(t),\hat{\boldsymbol{\Psi}}_k(t)],\quad t=1,2,\cdots,L, \tag{85}$$

$$\hat{\boldsymbol{\Psi}}_k(t)=\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t)\otimes\boldsymbol{I}_m, \tag{86}$$

$$\begin{aligned}\hat{\boldsymbol{\psi}}_k(t)=[&-\hat{\boldsymbol{w}}_{k-1}^{\mathrm{T}}(t-1),-\hat{\boldsymbol{w}}_{k-1}^{\mathrm{T}}(t-2),\cdots,-\hat{\boldsymbol{w}}_{k-1}^{\mathrm{T}}(t-n_c),\\&\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-1),\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-2),\cdots,\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-n_d)]T,\end{aligned} \tag{87}$$

$$\hat{\boldsymbol{w}}_k(t)=\boldsymbol{y}(t)-\boldsymbol{\Phi}(t)\hat{\boldsymbol{\theta}}_k, \tag{88}$$

$$\hat{\boldsymbol{v}}_k(t)=\boldsymbol{y}(t)-\hat{\boldsymbol{\Gamma}}_k(t)\hat{\boldsymbol{\vartheta}}_k, \tag{89}$$

$$\boldsymbol{Y}=[\boldsymbol{y}^{\mathrm{T}}(1),\boldsymbol{y}^{\mathrm{T}}(2),\cdots,\boldsymbol{y}^{\mathrm{T}}(L)]^{\mathrm{T}}, \tag{90}$$

$$\boldsymbol{\Phi}=[\boldsymbol{\Phi}^{\mathrm{T}}(1),\boldsymbol{\Phi}^{\mathrm{T}}(2),\cdots,\boldsymbol{\Phi}^{\mathrm{T}}(L)]^{\mathrm{T}}, \tag{91}$$

$$\boldsymbol{S}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi},\quad \boldsymbol{R}=\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}},\quad \boldsymbol{M}=\boldsymbol{I}-\boldsymbol{\Phi}\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}},$$
$$\boldsymbol{\alpha}=\boldsymbol{R}\boldsymbol{Y},\quad \boldsymbol{\beta}=\boldsymbol{\Phi}\boldsymbol{\alpha}. \tag{92}$$

读者可以结合上述多元伪线性滑动平均系统和多元伪线性自回归滑动平均系统的最小二乘迭代辨识方法,写出下列有色噪声干扰伪线性自回归滑动平均系统的最小二乘迭代辨识方法:

$$\boldsymbol{y}(t)=\boldsymbol{\Phi}(t)\boldsymbol{\theta}+\frac{d(z)}{c(z)}\boldsymbol{v}(t), \tag{93}$$

$$\boldsymbol{y}(t)=\boldsymbol{\Phi}(t)\boldsymbol{\theta}+\frac{\boldsymbol{D}(z)}{c(z)}\boldsymbol{v}(t), \tag{94}$$

$$\boldsymbol{y}(t)=\boldsymbol{\Phi}(t)\boldsymbol{\theta}+\boldsymbol{C}^{-1}(z)d(z)\boldsymbol{v}(t), \tag{95}$$

或

$$\boldsymbol{A}(z)\boldsymbol{y}(t)=\boldsymbol{\Phi}(t)\boldsymbol{\theta}+\boldsymbol{C}^{-1}(z)d(z)\boldsymbol{v}(t), \tag{96}$$

$$\boldsymbol{y}(t)=\frac{\boldsymbol{\Phi}(t)\boldsymbol{\theta}}{a(z)}+\boldsymbol{C}^{-1}(z)d(z)\boldsymbol{v}(t), \tag{97}$$

$$\boldsymbol{y}(t)=\boldsymbol{A}^{-1}(z)\boldsymbol{\Phi}(t)\boldsymbol{\theta}+\boldsymbol{C}^{-1}(z)d(z)\boldsymbol{v}(t). \tag{98}$$

4 多变量伪线性回归系统

多输入多输出伪线性系统,简称为多变量伪线性系统(multivariable pseudo-linear system). 多变量伪线性系统的特征是其对应辨识模型的信息向量中包含不可测未知变量(或噪声项).

4.1 多变量伪线性滑动平均系统

考虑下列多变量伪线性滑动平均系统(multivariable pseudo-linear moving average system):

$$\boldsymbol{A}(z)\boldsymbol{y}(t)=\boldsymbol{B}(z)\boldsymbol{u}(t)+d(z)\boldsymbol{v}(t), \tag{99}$$

其中 $\boldsymbol{u}(t)\in\mathbf{R}^r$ 为系统输入向量, $\boldsymbol{y}(t)\in\mathbf{R}^m$ 为系统输出向量, $\boldsymbol{v}(t)\in\mathbf{R}^m$ 为零均值随机噪声向量.

定义参数矩阵 $\boldsymbol{\theta}$、参数向量 $\boldsymbol{d}$ 和信息向量 $\boldsymbol{\varphi}(t)$ 如下:

$$\boldsymbol{\theta}^{\mathrm{T}}:=[\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_{n_a},\boldsymbol{B}_1,\boldsymbol{B}_2,\cdots,\boldsymbol{B}_{n_b}]\in\mathbf{R}^{m\times n_1},$$
$$n_1:=mn_a+rn_b,$$
$$\boldsymbol{d}:=[d_1,d_2,\cdots,d_{n_d}]^{\mathrm{T}}\in\mathbf{R}^{n_d},$$
$$\boldsymbol{\varphi}(t):=[-\boldsymbol{y}^{\mathrm{T}}(t-1),-\boldsymbol{y}^{\mathrm{T}}(t-2),\cdots,-\boldsymbol{y}^{\mathrm{T}}(t-n_a),$$
$$\boldsymbol{u}^{\mathrm{T}}(t-1),\boldsymbol{u}^{\mathrm{T}}(t-2),\cdots,\boldsymbol{u}^{\mathrm{T}}(t-n_b)]^{\mathrm{T}}\in\mathbf{R}^{n},$$
$$\boldsymbol{\Psi}(t):=[\boldsymbol{v}(t-1),\boldsymbol{v}(t-2),\cdots,\boldsymbol{v}(t-n_d)]\in\mathbf{R}^{m\times n_d},$$

则式(99)可等价写为

$$\boldsymbol{y}(t)=\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\varphi}(t)+\boldsymbol{\Psi}(t)\boldsymbol{d}+\boldsymbol{v}(t)=$$
$$[\boldsymbol{\varphi}^{\mathrm{T}}(t)\otimes\boldsymbol{I}_{n_1}]\mathrm{col}[\boldsymbol{\theta}]+\boldsymbol{\Psi}(t)\boldsymbol{d}+\boldsymbol{v}(t)=$$
$$[\boldsymbol{\varphi}^{\mathrm{T}}(t)\otimes\boldsymbol{I}_{n_1},\boldsymbol{\Psi}(t)]\begin{bmatrix}\mathrm{col}[\boldsymbol{\theta}]\\ \boldsymbol{d}\end{bmatrix}+\boldsymbol{v}(t)=$$
$$\boldsymbol{\Gamma}(t)\boldsymbol{\vartheta}+\boldsymbol{v}(t), \tag{100}$$

其中

$$\boldsymbol{\Gamma}(t):=[\boldsymbol{\Phi}(t),\boldsymbol{\Psi}(t)]\in\mathbf{R}^{m\times n},$$
$$n:=m^2n_a+mrn_b+n_d,$$
$$\boldsymbol{\Phi}(t):=\boldsymbol{\varphi}^{\mathrm{T}}(t)\otimes\boldsymbol{I}_{n_1}\in\mathbf{R}^{m\times(m^2n_a+mrn_b)},$$
$$\boldsymbol{\vartheta}:=\begin{bmatrix}\mathrm{col}[\boldsymbol{\theta}]\\ \boldsymbol{d}\end{bmatrix}\in\mathbf{R}^{m^2n_a+mrn_b+n_d}.$$

式(100)称为多变量伪线性滑动平均系统的辨识模型, $\boldsymbol{\vartheta}$ 是要从输入输出数据 $\{\boldsymbol{u}(t),\boldsymbol{y}(t)\}$ 辨识的参数向量.

1) 最小二乘迭代算法

估计式(100)中参数向量 $\boldsymbol{\vartheta}$ 的最小二乘迭代辨识算法如下:

$$\hat{\boldsymbol{\vartheta}}_k=[\hat{\boldsymbol{H}}_k^{\mathrm{T}}\hat{\boldsymbol{H}}_k]^{-1}\hat{\boldsymbol{H}}_k^{\mathrm{T}}Y,\quad k=1,2,3,\cdots, \tag{101}$$

$$\boldsymbol{Y}=[\boldsymbol{y}^{\mathrm{T}}(1),\boldsymbol{y}^{\mathrm{T}}(2),\cdots,\boldsymbol{y}^{\mathrm{T}}(L)]^{\mathrm{T}}, \tag{102}$$

$$\hat{\boldsymbol{H}}_k=\begin{bmatrix}\boldsymbol{\Phi}(1) & \hat{\boldsymbol{\Psi}}_k(1)\\ \boldsymbol{\Phi}(2) & \hat{\boldsymbol{\Psi}}_k(2)\\ \vdots & \vdots\\ \boldsymbol{\Phi}(L) & \hat{\boldsymbol{\Psi}}_k(L)\end{bmatrix}, \tag{103}$$

$$\hat{\boldsymbol{\Gamma}}_k(t)=[\boldsymbol{\Phi}(t),\hat{\boldsymbol{\Psi}}_k(t)], \tag{104}$$

$$\boldsymbol{\Phi}(t)=\boldsymbol{\phi}^{\mathrm{T}}(t)\otimes\boldsymbol{I}_{mn_a+rn_b}, \tag{105}$$

$$\hat{\boldsymbol{\Psi}}_k(t)=[\hat{\boldsymbol{v}}_{k-1}(t-1),\hat{\boldsymbol{v}}_{k-1}(t-2),\cdots,$$
$$\hat{\boldsymbol{v}}_{k-1}(t-n_d)], \tag{106}$$

$$\hat{\boldsymbol{v}}_k(t)=\boldsymbol{y}(t)-\hat{\boldsymbol{\Gamma}}_k(t)\hat{\boldsymbol{\vartheta}}_k,\quad t=1,2,\cdots,L. \tag{107}$$

2) 基于块矩阵求逆的最小二乘迭代算法

估计式(100)中参数向量 $\boldsymbol{\vartheta}$ 的基于块矩阵求逆的最小二乘迭代算法如下:

$$\hat{\boldsymbol{\vartheta}}_k=\begin{bmatrix}\boldsymbol{\alpha}+\boldsymbol{R}\hat{\boldsymbol{\Psi}}_k\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{Y})\\ -\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{Y})\end{bmatrix},\quad k=1,2,3,\cdots, \tag{108}$$

$$\hat{\boldsymbol{\Psi}}_k=[\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(1),\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(2),\cdots,\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}(L)]^{\mathrm{T}}, \tag{109}$$

$$\boldsymbol{Q}_k=\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{M}\hat{\boldsymbol{\Psi}}_k, \tag{110}$$

$$\hat{\boldsymbol{\Gamma}}_k(t)=[\boldsymbol{\Phi}(t),\hat{\boldsymbol{\Psi}}_k(t)],\quad t=1,2,\cdots,L, \tag{111}$$

$$\boldsymbol{\Phi}(t)=\boldsymbol{\varphi}^{\mathrm{T}}(t)\otimes\boldsymbol{I}_{mn_a+rn_b}, \tag{112}$$

$$\hat{\boldsymbol{\Psi}}_k(t)=[\hat{\boldsymbol{v}}_{k-1}(t-1),\hat{\boldsymbol{v}}_{k-1}(t-2),\cdots,$$
$$\hat{\boldsymbol{v}}_{k-1}(t-n_d)], \tag{113}$$

$$\hat{\boldsymbol{v}}_k(t)=\boldsymbol{y}(t)-\hat{\boldsymbol{\Gamma}}_k(t)\hat{\boldsymbol{\vartheta}}_k, \tag{114}$$

$$\boldsymbol{Y}=[\boldsymbol{y}^{\mathrm{T}}(1),\boldsymbol{y}^{\mathrm{T}}(2),\cdots,\boldsymbol{y}^{\mathrm{T}}(L)]^{\mathrm{T}}, \tag{115}$$

$$\boldsymbol{\Phi}=[\boldsymbol{\Phi}^{\mathrm{T}}(1),\boldsymbol{\Phi}^{\mathrm{T}}(2),\cdots,\boldsymbol{\Phi}^{\mathrm{T}}(L)]^{\mathrm{T}}, \tag{116}$$

$$\boldsymbol{S}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi},\quad \boldsymbol{R}=\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}},\quad \boldsymbol{M}=\boldsymbol{I}-\boldsymbol{\Phi}\boldsymbol{S}^{-1}\boldsymbol{\Phi}^{\mathrm{T}},$$
$$\boldsymbol{\alpha}=\boldsymbol{R}\boldsymbol{Y},\quad \boldsymbol{\beta}=\boldsymbol{\Phi}\boldsymbol{\alpha}. \tag{117}$$

4.2 交互干扰噪声多变量伪线性滑动平均系统

考虑下列交互干扰噪声多变量伪线性滑动平

均系统(multivariable pseudo-linear moving average systems with interactive noise),简称多变量受控滑动平均系统(multivariable CARMA system)[32-33]:

$$\boldsymbol{A}(z)\boldsymbol{y}(t)=\boldsymbol{B}(z)\boldsymbol{u}(t)+\boldsymbol{D}(z)\boldsymbol{v}(t).$$

定义参数矩阵 $\boldsymbol{\theta}$ 和信息向量 $\boldsymbol{\varphi}(t)$ 如下:

$$\boldsymbol{\theta}^{\mathrm{T}}:=[\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_{n_a},\boldsymbol{B}_1,\boldsymbol{B}_2,\cdots,\boldsymbol{B}_{n_b},\boldsymbol{D}_1,\boldsymbol{D}_2,\cdots,\boldsymbol{D}_{n_d}]\in\mathbf{R}^{m\times n},n:=mn_a+rn_b+mn_d,$$

$$\boldsymbol{\varphi}(t):=\begin{bmatrix}\boldsymbol{\phi}(t)\\\boldsymbol{\psi}(t)\end{bmatrix}\in\mathbf{R}^n,$$

$$\boldsymbol{\phi}(t):=[-\boldsymbol{y}^{\mathrm{T}}(t-1),-\boldsymbol{y}^{\mathrm{T}}(t-2),\cdots,-\boldsymbol{y}^{\mathrm{T}}(t-n_a),\boldsymbol{u}^{\mathrm{T}}(t-1),\boldsymbol{u}^{\mathrm{T}}(t-2),\cdots,\boldsymbol{u}^{\mathrm{T}}(t-n_b)]^{\mathrm{T}}\in\mathbf{R}^{mn_a+rn_b},$$

$$\boldsymbol{\psi}(t):=[\boldsymbol{v}^{\mathrm{T}}(t-1),\boldsymbol{v}^{\mathrm{T}}(t-2),\cdots,\boldsymbol{v}^{\mathrm{T}}(t-n_d)]^{\mathrm{T}}\in\mathbf{R}^{mn_d}.$$

则多变量受控滑动平均系统(118)对应的辨识模型为

$$\boldsymbol{y}(t)=\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\varphi}(t)+\boldsymbol{v}(t).\tag{119}$$

1) 最小二乘迭代算法

考虑从 $t=1$ 到 $t=L$ 的数据($L\gg n$ 为数据长度),定义堆积输出矩阵(stacked output matrix)$\boldsymbol{Y}$,堆积信息矩阵(stacked information matrix)$\boldsymbol{H}$ 和堆积白噪声矩阵 $\boldsymbol{V}$ 如下:

$$\boldsymbol{Y}:=[\boldsymbol{y}(1),\boldsymbol{y}(2),\cdots,\boldsymbol{y}(L)]\in\mathbf{R}^{m\times L},$$

$$\boldsymbol{H}:=[\boldsymbol{\varphi}(1),\boldsymbol{\varphi}(2),\cdots,\boldsymbol{\varphi}(L)]\in\mathbf{R}^{n\times L},$$

$$\boldsymbol{V}:=[\boldsymbol{v}(1),\boldsymbol{v}(2),\cdots,\boldsymbol{v}(L)]\in\mathbf{R}^{m\times L}.$$

$\boldsymbol{Y}$ 和 $\boldsymbol{H}$ 包含了所有量测数据$\{u(t),y(t):t=1,2,\cdots,L\}$. 从式(119)可得

$$\boldsymbol{Y}=\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{H}+\boldsymbol{V}.\tag{120}$$

定义二次准则函数(quadratic criterion function):

$$J_3(\boldsymbol{\theta}):=\|\boldsymbol{Y}-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{H}\|^2.$$

因为 $\boldsymbol{V}$ 是一个零均值白噪声矩阵. 令 $J_3(\boldsymbol{\theta})$ 对 $\boldsymbol{\theta}$ 的偏导数为零,得到

$$\frac{\partial J_3(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}=-2[\boldsymbol{Y}-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{H}]\boldsymbol{H}^{\mathrm{T}}=\mathbf{0}.$$

假设信息向量 $\boldsymbol{\varphi}(t)$ 是持续激励的,即$[\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}]$是可逆矩阵,从上式求得 $\boldsymbol{\theta}$ 的最小二乘估计(Least Squares Estimate,LSE):

$$\hat{\boldsymbol{\theta}}=[\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}]^{-1}\boldsymbol{H}\boldsymbol{Y}^{\mathrm{T}}\tag{121}$$

$$=\Big[\sum_{t=1}^{L}\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\Big]^{-1}\sum_{t=1}^{L}\boldsymbol{\varphi}(t)\boldsymbol{y}^{\mathrm{T}}(t).\tag{122}$$

用式 $\hat{\boldsymbol{H}}_k$ 代替式(121)中 $\boldsymbol{H}$,用 $\hat{\boldsymbol{\theta}}_k$ 表示 $\boldsymbol{\theta}$ 的第 k 次迭代估计,可以得到式(120)参数矩阵 $\boldsymbol{\theta}$ 的最小二乘迭代算法(least squares based iterative algorithm)[34]:

$$\hat{\boldsymbol{\theta}}_k=[\hat{\boldsymbol{H}}_k\hat{\boldsymbol{H}}_k^{\mathrm{T}}]^{-1}\hat{\boldsymbol{H}}_k\boldsymbol{Y}^{\mathrm{T}},\quad k=1,2,3,\cdots,\tag{123}$$

$$\boldsymbol{Y}=[\boldsymbol{y}(1),\boldsymbol{y}(2),\cdots,\boldsymbol{y}(L)],\tag{124}$$

$$\hat{\boldsymbol{H}}_k=[\hat{\boldsymbol{\varphi}}_k(1),\hat{\boldsymbol{\varphi}}_k(2),\cdots,\hat{\boldsymbol{\varphi}}_k(L)],\tag{125}$$

$$\hat{\boldsymbol{\varphi}}_k(t)=[\boldsymbol{\phi}^{\mathrm{T}}(t),\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t)]^{\mathrm{T}},t=1,2,\cdots,L,]\tag{126}$$

$$\boldsymbol{\phi}(t)=[-\boldsymbol{y}^{\mathrm{T}}(t-1),-\boldsymbol{y}^{\mathrm{T}}(t-2),\cdots,-\boldsymbol{y}^{\mathrm{T}}(t-n_a),\boldsymbol{u}^{\mathrm{T}}(t-1),\boldsymbol{u}^{\mathrm{T}}(t-2),\cdots,\boldsymbol{u}^{\mathrm{T}}(t-n_b)]^{\mathrm{T}},\tag{127}$$

$$\hat{\boldsymbol{\psi}}_k(t)=[\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-1),\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-2),\cdots,\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-n_d)]^{\mathrm{T}},\tag{128}$$

$$\hat{\boldsymbol{v}}_k(t)=\boldsymbol{y}(t)-\hat{\boldsymbol{\theta}}_k^{\mathrm{T}}\hat{\boldsymbol{\phi}}_k(t).\tag{129}$$

2) 基于块矩阵求逆的最小二乘迭代算法

估计式(120)参数矩阵 $\boldsymbol{\theta}$ 的基于块矩阵求逆的最小二乘迭代算法如下:

$$\hat{\boldsymbol{\theta}}_k=\begin{bmatrix}\boldsymbol{\alpha}+\boldsymbol{R}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k(\boldsymbol{\beta}-\boldsymbol{Y}^{\mathrm{T}})\\-\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k(\boldsymbol{\beta}-\boldsymbol{Y}^{\mathrm{T}})\end{bmatrix},\quad k=1,2,3,\cdots,\tag{130}$$

$$\hat{\boldsymbol{\Psi}}_k=[\hat{\boldsymbol{\psi}}_k(1),\hat{\boldsymbol{\psi}}_k(2),\cdots,\hat{\boldsymbol{\psi}}_k(L)],\tag{131}$$

$$\boldsymbol{Q}_k=\hat{\boldsymbol{\Psi}}_k\boldsymbol{M}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}},\tag{132}$$

$$\hat{\boldsymbol{\varphi}}_k(t)=[\boldsymbol{\phi}^{\mathrm{T}}(t),\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t)]^{\mathrm{T}},t=1,2,\cdots,L,\tag{133}$$

$$\hat{\boldsymbol{\psi}}_k(t)=[\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-1),\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-2),\cdots,\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-n_d)]^{\mathrm{T}},\tag{134}$$

$$\hat{\boldsymbol{v}}_k(t)=\boldsymbol{y}(t)-\hat{\boldsymbol{\theta}}_k^{\mathrm{T}}\hat{\boldsymbol{\phi}}_k(t),\tag{135}$$

$$\boldsymbol{\phi}(t)=[-\boldsymbol{y}^{\mathrm{T}}(t-1),-\boldsymbol{y}^{\mathrm{T}}(t-2),\cdots,-\boldsymbol{y}^{\mathrm{T}}(t-n_a),\boldsymbol{u}^{\mathrm{T}}(t-1),\boldsymbol{u}^{\mathrm{T}}(t-2),\cdots,\boldsymbol{u}^{\mathrm{T}}(t-n_b)]^{\mathrm{T}},\tag{136}$$

$$\boldsymbol{Y}=[\boldsymbol{y}(1),\boldsymbol{y}(2),\cdots,\boldsymbol{y}(L)],\tag{137}$$

$$\boldsymbol{\Phi}=[\boldsymbol{\phi}(1),\boldsymbol{\phi}(2),\cdots,\boldsymbol{\phi}(L)],\tag{138}$$

$$\boldsymbol{S}=\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}},\quad\boldsymbol{R}=\boldsymbol{S}^{-1}\boldsymbol{\Phi},\quad\boldsymbol{M}=\boldsymbol{I}-\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{S}^{-1}\boldsymbol{\Phi},\quad\boldsymbol{\alpha}=\boldsymbol{R}\boldsymbol{Y}^{\mathrm{T}},\quad\boldsymbol{\beta}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\alpha}.\tag{139}$$

4.3 多变量伪线性自回归滑动平均系统

考虑下列多变量受控自回归 ARMA 系统(controlled autoregressive ARMA system,CARARMA)[35-36]:

$$\boldsymbol{A}(z)\boldsymbol{y}(t)=\boldsymbol{B}(z)\boldsymbol{u}(t)+\boldsymbol{C}^{-1}(z)\boldsymbol{D}(z)\boldsymbol{v}(t).\tag{140}$$

令

$$\boldsymbol{w}(t):=\boldsymbol{C}^{-1}(z)\boldsymbol{D}(z)\boldsymbol{v}(t)\in\mathbf{R}^m.\tag{141}$$

定义参数矩阵 $\boldsymbol{\vartheta}$ 和信息向量 $\boldsymbol{\varphi}(t)$ 如下:

$$\boldsymbol{\vartheta}^{\mathrm{T}}:=[\boldsymbol{\theta}^{\mathrm{T}},\boldsymbol{\theta}_2^{\mathrm{T}}]\in\mathbf{R}^{m\times n},n:=mn_a+rn_b+mn_c+mn_d,$$

$$\boldsymbol{\theta}^{\mathrm{T}}:=[\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_{n_a},\boldsymbol{B}_1,\boldsymbol{B}_2,\cdots,\boldsymbol{B}_{n_b}]\in\mathbf{R}^{m\times(mn_a+rn_b)},$$

$$\boldsymbol{\theta}_2^{\mathrm{T}}:=[\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_{n_c},\boldsymbol{D}_1,\boldsymbol{D}_2,\cdots,\boldsymbol{D}_{n_d}]\in\mathbf{R}^{m\times(mn_c+mn_d)},$$

$$\boldsymbol{\varphi}(t):=\begin{bmatrix}\boldsymbol{\phi}(t)\\\boldsymbol{\psi}(t)\end{bmatrix}\in\mathbf{R}^n,$$

$$\boldsymbol{\phi}(t):=[-\boldsymbol{y}^{\mathrm{T}}(t-1),-\boldsymbol{y}^{\mathrm{T}}(t-2),\cdots,-\boldsymbol{y}^{\mathrm{T}}(t-n_a),\boldsymbol{u}^{\mathrm{T}}(t-1),\boldsymbol{u}^{\mathrm{T}}(t-2),\cdots,\boldsymbol{u}^{\mathrm{T}}(t-n_b)]^{\mathrm{T}}\in\mathbf{R}^{mn_a+rn_b},$$

$$\boldsymbol{\psi}(t):=[-\boldsymbol{w}^{\mathrm{T}}(t-1),-\boldsymbol{w}^{\mathrm{T}}(t-2),\cdots,-\boldsymbol{w}^{\mathrm{T}}(t-n_d),$$

$\boldsymbol{v}^{\mathrm{T}}(t-1),\boldsymbol{v}^{\mathrm{T}}(t-2),\cdots,\boldsymbol{v}^{\mathrm{T}}(t-n_d)]^{\mathrm{T}}\in\mathbf{R}^{mn_c+mn_d}$.

式(141)和(140)可以写成下列形式:

$$\boldsymbol{w}(t)=[\boldsymbol{I}-\boldsymbol{C}(z)]\boldsymbol{w}(t)+\boldsymbol{D}(z)\boldsymbol{v}(t)=\boldsymbol{\theta}_2^{\mathrm{T}}\boldsymbol{\psi}(t)+\boldsymbol{v}(t),\tag{142}$$

$$\boldsymbol{y}(t)=[\boldsymbol{I}-\boldsymbol{A}(z)]\boldsymbol{y}(t)+\boldsymbol{B}(z)\boldsymbol{u}(t)+\boldsymbol{w}(t)=\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}(t)+\boldsymbol{w}(t)\tag{143}$$

$$=\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}(t)+\boldsymbol{\theta}_2^{\mathrm{T}}\boldsymbol{\psi}(t)+\boldsymbol{v}(t)=\boldsymbol{\vartheta}^{\mathrm{T}}\boldsymbol{\varphi}(t)+\boldsymbol{v}(t).\tag{144}$$

1)最小二乘迭代算法

定义二次准则函数(quadratic criterion function):

$$J_4(\boldsymbol{\vartheta}):=\sum_{t=1}^{L}[\boldsymbol{y}(t)-\boldsymbol{\vartheta}^{\mathrm{T}}\boldsymbol{\varphi}(t)]^{\mathrm{T}}[\boldsymbol{y}(t)-\boldsymbol{\vartheta}^{\mathrm{T}}\boldsymbol{\varphi}(t)].$$

极小化$J_4(\boldsymbol{\vartheta})$,信息向量中有关不可测变量用其估计值代替,可以得到估计式(144)参数向量$\boldsymbol{\vartheta}$的最小二乘迭代辨识方法[36]:

$$\hat{\boldsymbol{\vartheta}}_k=[\hat{\boldsymbol{H}}_k\hat{\boldsymbol{H}}_k^{\mathrm{T}}]^{-1}\hat{\boldsymbol{H}}_k\boldsymbol{Y}^{\mathrm{T}}=\tag{145}$$

$$[\sum_{t=1}^{L}\hat{\boldsymbol{\varphi}}_k(t)\hat{\boldsymbol{\phi}}_k^{\mathrm{T}}(t)]^{-1}\sum_{t=1}^{L}\hat{\boldsymbol{\varphi}}_k(t)\boldsymbol{y}^{\mathrm{T}}(t),\quad k=1,2,3,\cdots,$$

$$\boldsymbol{Y}=[\boldsymbol{y}(1),\boldsymbol{y}(2),\cdots,\boldsymbol{y}(L)],\tag{146}$$

$$\hat{\boldsymbol{H}}_k=[\hat{\boldsymbol{\varphi}}_k(1),\hat{\boldsymbol{\varphi}}_k(2),\cdots,\hat{\boldsymbol{\varphi}}_k(L)],\tag{147}$$

$$\hat{\boldsymbol{\varphi}}_k(t)=[\boldsymbol{\phi}^{\mathrm{T}}(t),\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t)]^{\mathrm{T}},\quad t=1,2,\cdots,L,\tag{148}$$

$$\boldsymbol{\phi}(t)=[-\boldsymbol{y}^{\mathrm{T}}(t-1),-\boldsymbol{y}^{\mathrm{T}}(t-2),\cdots,-\boldsymbol{y}^{\mathrm{T}}(t-n_a),\boldsymbol{u}^{\mathrm{T}}(t-1),\boldsymbol{u}^{\mathrm{T}}(t-2),\cdots,\boldsymbol{u}^{\mathrm{T}}(t-n_b)]^{\mathrm{T}},\tag{149}$$

$$\hat{\boldsymbol{\psi}}_k(t)=[-\hat{\boldsymbol{w}}_{k-1}^{\mathrm{T}}(t-1),-\hat{\boldsymbol{w}}_{k-1}^{\mathrm{T}}(t-2),\cdots,-\hat{\boldsymbol{w}}_{k-1}^{\mathrm{T}}(t-n_c),\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-1),\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-2),\cdots,\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-n_d)]T,\tag{150}$$

$$\hat{\boldsymbol{\vartheta}}_k^{\mathrm{T}}=[\hat{\boldsymbol{\theta}}_k^{\mathrm{T}},\hat{\boldsymbol{\theta}}_{2,k}^{\mathrm{T}}],\tag{151}$$

$$\hat{\boldsymbol{w}}_k(t)=\boldsymbol{y}(t)-\hat{\boldsymbol{\theta}}_k^{\mathrm{T}}\boldsymbol{\phi}(t),\tag{152}$$

$$\hat{\boldsymbol{v}}_k(t)=\boldsymbol{y}(t)-\hat{\boldsymbol{\vartheta}}_k^{\mathrm{T}}\hat{\boldsymbol{\varphi}}_k(t).\tag{153}$$

2)基于块矩阵求逆的最小二乘迭代算法

估计式(144)参数矩阵$\boldsymbol{\theta}$的基于块矩阵求逆的最小二乘迭代算法如下:

$$\hat{\boldsymbol{\vartheta}}_k=\begin{bmatrix}\boldsymbol{\alpha}+\boldsymbol{R}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k(\boldsymbol{\beta}-\boldsymbol{Y}^{\mathrm{T}})\\-\boldsymbol{Q}_k^{-1}\hat{\boldsymbol{\Psi}}_k(\boldsymbol{\beta}-\boldsymbol{Y}^{\mathrm{T}})\end{bmatrix},\quad k=1,2,3,\cdots,\tag{154}$$

$$\hat{\boldsymbol{\Psi}}_k=[\hat{\boldsymbol{\psi}}_k(1),\hat{\boldsymbol{\psi}}_k(2),\cdots,\hat{\boldsymbol{\psi}}_k(L)],\tag{155}$$

$$\boldsymbol{Q}_k=\hat{\boldsymbol{\Psi}}_k\boldsymbol{M}\hat{\boldsymbol{\Psi}}_k^{\mathrm{T}},\tag{156}$$

$$\hat{\boldsymbol{\varphi}}_k(t)=[\boldsymbol{\phi}^{\mathrm{T}}(t),\hat{\boldsymbol{\psi}}_k^{\mathrm{T}}(t)]^{\mathrm{T}},\quad t=1,2,\cdots,L,\tag{157}$$

$$\hat{\boldsymbol{\psi}}_k(t)=[-\hat{\boldsymbol{w}}_{k-1}^{\mathrm{T}}(t-1),-\hat{\boldsymbol{w}}_{k-1}^{\mathrm{T}}(t-2),\cdots,-\hat{\boldsymbol{w}}_{k-1}^{\mathrm{T}}(t-n_c),\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-1),\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-2),\cdots,\hat{\boldsymbol{v}}_{k-1}^{\mathrm{T}}(t-n_d)]^{\mathrm{T}},\tag{158}$$

$$\hat{\boldsymbol{\vartheta}}_k^{\mathrm{T}}=[\hat{\boldsymbol{\theta}}_k^{\mathrm{T}},\hat{\boldsymbol{\theta}}_{2,k}^{\mathrm{T}}],\tag{159}$$

$$\hat{\boldsymbol{w}}_k(t)=\boldsymbol{y}(t)-\hat{\boldsymbol{\theta}}_k^{\mathrm{T}}\boldsymbol{\phi}(t),\tag{160}$$

$$\hat{\boldsymbol{v}}_k(t)=\boldsymbol{y}(t)-\hat{\boldsymbol{\vartheta}}_k^{\mathrm{T}}\hat{\boldsymbol{\varphi}}_k(t),\tag{161}$$

$$\boldsymbol{\phi}(t)=[-\boldsymbol{y}^{\mathrm{T}}(t-1),-\boldsymbol{y}^{\mathrm{T}}(t-2),\cdots,-\boldsymbol{y}^{\mathrm{T}}(t-n_a),\boldsymbol{u}^{\mathrm{T}}(t-1),\boldsymbol{u}^{\mathrm{T}}(t-2),\cdots,\boldsymbol{u}^{\mathrm{T}}(t-n_b)]^{\mathrm{T}},\tag{162}$$

$$\boldsymbol{Y}=[\boldsymbol{y}(1),\boldsymbol{y}(2),\cdots,\boldsymbol{y}(L)],\tag{163}$$

$$\boldsymbol{\Phi}=[\boldsymbol{\phi}(1),\boldsymbol{\phi}(2),\cdots,\boldsymbol{\phi}(L)],\tag{164}$$

$$\boldsymbol{S}=\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}},\quad\boldsymbol{R}=\boldsymbol{S}^{-1}\boldsymbol{\Phi},\quad\boldsymbol{M}=\boldsymbol{I}-\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{S}^{-1}\boldsymbol{\Phi},\quad\boldsymbol{\alpha}=\boldsymbol{R}\boldsymbol{Y}^{\mathrm{T}},\quad\boldsymbol{\beta}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\alpha}.\tag{165}$$

基于块矩阵求逆的最小二乘迭代算法(154)—(165)计算参数估计向量$\hat{\boldsymbol{\vartheta}}_k$的步骤如下.

1)采集输入输出数据$\{u(t),y(t):t=1,2,3,\cdots,L\}$.

2)初始化:令$k=1$,$\hat{\boldsymbol{w}}_0(t)$是随机向量,$\hat{\boldsymbol{v}}_0(t)$是随机向量,给定小正数$\varepsilon$.

3)用式(163)构成$\boldsymbol{Y}$,用式(162)和(164)构成$\boldsymbol{\phi}(t)$和$\boldsymbol{\Phi}$.

4)用式(165)计算$\boldsymbol{S},\boldsymbol{R},\boldsymbol{M},\boldsymbol{\alpha}$和$\boldsymbol{\beta}$.

5)用式(158)构成$\hat{\boldsymbol{\psi}}_k(t)$,用式(157)构成$\hat{\boldsymbol{\varphi}}_k(t)$,用式(155)构成$\hat{\boldsymbol{\Psi}}_k$,用式(156)计算$\boldsymbol{Q}_k$.

6)用式(154)刷新参数估计向量$\hat{\boldsymbol{\vartheta}}_k$,用式(160)计算$\hat{\boldsymbol{w}}_k(t)$,用式(161)计算$\hat{\boldsymbol{v}}_k(t)$.

7)如果$\|\hat{\boldsymbol{\vartheta}}_k-\hat{\boldsymbol{\vartheta}}_{k-1}\|\leqslant\varepsilon$,那么中断迭代循环,获得迭代次数$k$和参数估计向量$\hat{\boldsymbol{\vartheta}}_k$;否则$k$增1,返回到第5步.

基于块矩阵求逆的最小二乘迭代算法(154)—(165)计算参数估计向量$\hat{\boldsymbol{\vartheta}}_k$的流程如图2所示.

读者可以研究下列有色噪声干扰的多变量方程误差类系统和多变量输出误差类系统的最小二乘迭代辨识方法:

$$\boldsymbol{A}(z)\boldsymbol{y}(t)=\boldsymbol{B}(z)\boldsymbol{u}(t)+\frac{1}{c(z)}\boldsymbol{v}(t),\tag{166}$$

$$\boldsymbol{A}(z)\boldsymbol{y}(t)=\boldsymbol{B}(z)\boldsymbol{u}(t)+\boldsymbol{C}^{-1}(z)\boldsymbol{v}(t),\tag{167}$$

$$\boldsymbol{A}(z)\boldsymbol{y}(t)=\boldsymbol{B}(z)\boldsymbol{u}(t)+\boldsymbol{C}^{-1}(z)d(z)\boldsymbol{v}(t),\tag{168}$$

$$\boldsymbol{A}(z)\boldsymbol{y}(t)=\boldsymbol{B}(z)\boldsymbol{u}(t)+\frac{\boldsymbol{D}(z)}{c(z)}\boldsymbol{v}(t),\tag{169}$$

$$\boldsymbol{A}(z)\boldsymbol{y}(t)=\boldsymbol{B}(z)\boldsymbol{u}(t)+\boldsymbol{C}^{-1}(z)d(z)\boldsymbol{v}(t),\tag{170}$$

或

$$\boldsymbol{y}(t)=\frac{\boldsymbol{B}(z)}{a(z)}\boldsymbol{u}(t)+d(z)\boldsymbol{v}(t),\tag{171}$$

$$\boldsymbol{y}(t)=\frac{\boldsymbol{B}(z)}{a(z)}\boldsymbol{u}(t)+\boldsymbol{D}(z)\boldsymbol{v}(t),\tag{172}$$

$$\boldsymbol{y}(t)=\boldsymbol{A}^{-1}(z)\boldsymbol{B}(z)\boldsymbol{u}(t)+d(z)\boldsymbol{v}(t),\tag{173}$$

$$\boldsymbol{y}(t)=\boldsymbol{A}^{-1}(z)\boldsymbol{B}(z)\boldsymbol{u}(t)+\boldsymbol{D}(z)\boldsymbol{v}(t),\tag{174}$$

$$\boldsymbol{y}(t)=\frac{\boldsymbol{B}(z)}{a(z)}\boldsymbol{u}(t)+\frac{1}{c(z)}\boldsymbol{v}(t),\tag{175}$$

$$\boldsymbol{y}(t)=\boldsymbol{A}^{-1}(z)\boldsymbol{B}(z)\boldsymbol{u}(t)+\frac{1}{c(z)}\boldsymbol{v}(t),\tag{176}$$

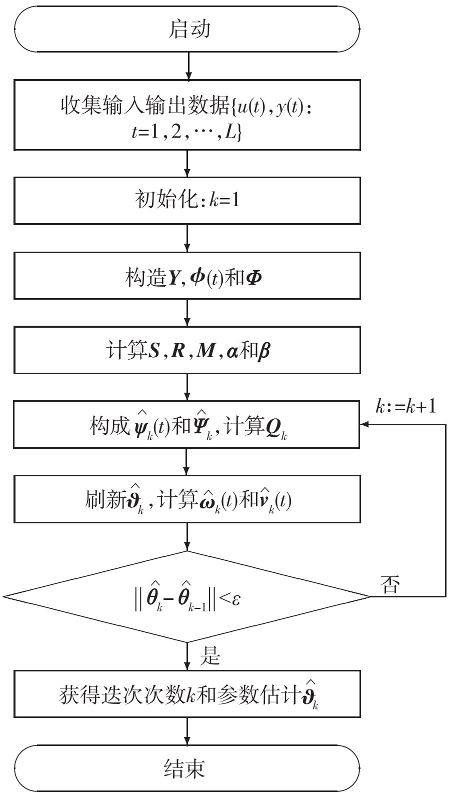

图 2　计算参数估计向量 $\hat{\boldsymbol{\vartheta}}_k$ 的流程

Fig. 2　The flowchart of computing the parameter estimation vector $\hat{\boldsymbol{\vartheta}}_k$

$$\boldsymbol{y}(t)=\boldsymbol{A}^{-1}(z)\boldsymbol{B}(z)\boldsymbol{u}(t)+\boldsymbol{C}^{-1}(z)\boldsymbol{v}(t), \tag{177}$$

$$\boldsymbol{y}(t)=\boldsymbol{A}^{-1}(z)\boldsymbol{B}(z)\boldsymbol{u}(t)+\boldsymbol{C}^{-1}(z)d(z)\boldsymbol{v}(t), \tag{178}$$

$$\boldsymbol{y}(t)=\boldsymbol{A}^{-1}(z)\boldsymbol{B}(z)\boldsymbol{u}(t)+\frac{\boldsymbol{D}(z)}{c(z)}\boldsymbol{v}(t), \tag{179}$$

$$\boldsymbol{F}(z)\boldsymbol{y}(t)=\boldsymbol{A}^{-1}(z)\boldsymbol{B}(z)\boldsymbol{u}(t)+\boldsymbol{C}^{-1}(z)d(z)\boldsymbol{v}(t). \tag{180}$$

5　结语

迭代算法可以解决信息向量中包含不可测变量的系统辨识问题. 本文主要从辨识算法的实现方式上,研究了伪线性回归系统、多元伪线性回归系统、多变量伪线性回归系统的最小二乘迭代辨识算法、基于块矩阵求逆的最小二乘迭代辨识算法. 基于块矩阵求逆的最小二乘迭代算法只是最小二乘迭代算法的一种实现方式,不是一种新算法,二者产生的参数估计是相同的,但前者的计算量小. 虽然基于辨识模型分解的递阶辨识方法[19-24,37]、两阶段辨识方法[38-40]等都是计算量小的辨识方法,但是本文方法不同之处在于从算法的实现角度来减小计算量.

参考文献
References

[1] 丁锋. 系统辨识新论[M]. 北京:科学出版社,2012
DING Feng. System Identification: New Theory and Methods[M]. Beijing:Science Press,2012

[2] 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

[3] 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

[4] 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

[5] 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

[6] 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

[7] 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

[8] 丁锋. 系统辨识(7):递阶辨识原理与方法[J]. 南京信息工程大学学报:自然科学版,2012,4(2):97-124
DING Feng. System identification. Part G:Hierarchical identification principle and methods[J]. Journal of Nanjing University of Information Science & Technology:Natural Science Edition,2012,4(2):97-124

[9] 丁锋. 系统辨识(8):耦合辨识概念与方法[J]. 南京信息工程大学学报:自然科学版,2012,4(3):193-212
DING F. System identification. Part H:Coupling identification concept and methods[J]. Journal of Nanjing University of Information Science and Technology:Natural Science Edition,2011,4(3):193-212

[10] Ding F,Chen T. Identification of dual-rate systems based on finite impulse response models [J]. International Journal of Adaptive Control and Signal Processing,2004,18(7):589-598

[11] Ding F,Chen T. Combined parameter and output estimation of dual-rate systems using an auxiliary model[J]. Automatica,2004,40(10):1739-1748

[12] Ding F,Chen T. Parameter estimation of dual-rate stochastic systems by using an output error method [J]. IEEE Transactions on Automatic Control,2005,50(9):1436-1441

[13] Ding F,Ding J. Least squares parameter estimation with irregularly missing data[J]. International Journal of Adaptive Control and Signal Processing,2010,24(7):540-553

[14] Ding F,Liu G,Liu X P. Parameter estimation with scarce measurements[J]. Automatica,2011,47(8):1646-1655

[15] Ding F,Chen T. Performance analysis of multi-innovation gradient type identification methods [J]. Automatica,2007,43(1):1-14

[16] Ding F,Liu X P,Liu G. Multi-innovation least squares identification for system modeling[J]. IEEE Transactions on Systems,Man,and Cybernetics,Part B:Cybernetics,2010,40(3):767-778

[17] Ding F,Liu X P,Liu G. Auxiliary model based multi-innovation extended stochastic gradient parameter estimation with colored measurement noises [J]. Signal Processing,2009,89(10):1883-1890

[18] Zhang J B,Ding F,Shi Y. Self-tuning control based on multi-innovation stochastic gradient parameter estimation [J]. Systems & Control Letters,2009,58(1):69-75

[19] Ding F,Chen T. Hierarchical gradient-based identification of multivariable discrete-time systems[J]. Automatica,2005,41(2):315-325

[20] Ding F,Chen T. Hierarchical least squares identification methods for multivariable systems [J]. IEEE Transactions on Automatic Control,2005,50(3):397-402

[21] 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

[22] 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

[23] Zhang Z N,Ding F,Liu X G. Hierarchical gradient based iterative parameter estimation algorithm for multivariable output error moving average systems[J]. Computers and Mathematics with Applications,2011,61(3):672-682

[24] 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

[25] Ding F,Liu G,Liu X P. Partially coupled stochastic gradient identification methods for non-uniformly sampled systems[J]. IEEE Transactions on Automatic Control,2010,55(8):1976-1981

[26] Ding F. Coupled least squares identification for multivariable systems [J]. IET Control Theory and Applications,2013,Accepted

[27] 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

[28] 丁锋.辨识方法的计算效率(1):递推算法[J].南京信息工程大学学报:自然科学版,2012,4(4):289-300
DING Feng. Computational efficiency of the identification methods. Part A:Recursive algorithms [J]. Journal of Nanjing University of Information Science and Technology:Natural Science Edition,2012,4(4):289-300

[29] Golub G H,Van Loan C F. Matrix Computations[M],3rd Ed. Baltimore,MD:Johns Hopkins University Press,1996

[30] Hu H Y,Ding F. An iterative least squares estimation algorithm for controlled moving average systems based on matrix decomposition[J]. Applied Mathematics Letters,2012,25(12):2332-2338

[31] Ding F,Liu X P,Liu G. Gradient based and least-squares based iterative identification methods for OE and OEMA systems[J]. Digital Signal Processing,2010,20(3):664-677

[32] 丁锋,杨慧中,刘飞.弱条件下随机梯度算法性能分析[J].中国科学:E辑,2008,38(12):2173-2184

[33] Ding F,Yang H Z,Liu F. Performance analysis of stochastic gradient algorithms under weak conditions[J]. Science in China Series F:Information Sciences,2008,51(9):1269-128

[34] Bao B,Xu Y Q,Sheng J,et al. Least squares based iterative parameter estimation algorithm for multivariable controlled ARMA systems modelling with finite measurement data[J]. Mathematical and Computer Modelling,2011,53(9/10):1664-1669

[35] Wang D Q,Ding F. Input-output data filtering based recursive least squares parameter estimation for CARARMA systems [J]. Digital Signal Processing,2010,20(4):991-999

[36] Ding F,Liu Y J,Bao B. Gradient based and least squares based iterative estimation algorithms for multi-input multi-output systems[J]. Journal of Systems and Control Engineering,2012,226(1):43-55

[37] 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

[38] 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

[39] Yao G Y,Ding R F. Two-stage least squares based iterative identification algorithm for controlled autoregressive moving average (CARMA) systems[J]. Computers and Mathematics with Applications,2012,63(5):975-984

[40] Xiao Y S,Zhang Y,Ding J,et al. The residual based interactive least squares algorithms and simulation studies [J]. Computers and Mathematics with Applications,2009,58(6):1190-1197

Computational efficiency of the identification methods. Part B: Iterative algorithms

DING Feng[1,2,3]

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 This paper focuses on the computational efficiency of the least squares based iterative algorithms. The computational burdens of the least squares based iterative (LSI) algorithms are heavy due to computing large-size matrix inversion. In order to reduce the computational burdens, the block matrix inversion based LSI algorithms are presented. The proposed methods can reduce the computational cost through simplifying the implementation of the least squares based iterative algorithms, thus the estimation accuracies remain unchanged. The least squares based iterative algorithms and the block matrix inversion based LSI methods are studied for pseudo-linear regression systems, multivariate pseudo-linear regression systems and multivariable pseudo-linear systems.

Key words recursive identification; iterative 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