Householder Transform in Least Square Parameter Estimation

最小二乘辨识中的Householder变换法*

姜伟 刘豹

（天津大学）

一 引言

最小二乘法是辨识中一种最基本的方法，也是一种很好的方法。无论在辨识学科本身，或随机控制（如自校正调节器）中，都有着广泛的应用。一般文献介绍的在线算法都是递推法，递推公式是对最小二乘法的方程进行数学加工而导出的。它每递推一步计算量小，方法简单。但是，应当指出，一般文献都是从数学角度介绍最小二乘法，而实际应用中，往往会遇到病态问题。这是数值计算方面的一项困难。病态问题不但随参数个数增加（即阶数增加）而严重，而且与计算机字长有密切关系。所以在目前小型、微型机逐渐普及的情况下，研究病态问题及克服病态的算法是有意义的。

二 问题评述

设有一个系统由差分方程描述：

$$y(k)+\sum_{i=1}^{n}a_i y(k-i)=\sum_{i=1}^{n}b_i u(k-l-i)+e(k) \quad (1)$$

其中$u(k)$,$y(k)$分别为输入、输出序列；l为延时；$e(k)$为噪声序列。令

$$\theta^{(注)}=[a_1,\cdots,a_n,b_1,\cdots,b_n]^T \quad (2)$$

$$\varphi(k)=[-y(k-1),\ \cdots,\ -y(k-n),\ u(k-l-1),\cdots,u(k-l-n)]^T \quad (3)$$

则(1)式可写为：

$$y(k)=\varphi^T(k)\theta+e(k) \quad (4)$$

令 $$\zeta(k)=[y(1),y(2),\cdots,\ y(k)]^T \quad (5)$$

$$\varepsilon(k)=[e(1),e(2),\cdots,e(k)]^T \quad (6)$$

$$\phi(k)=[\varphi(1),\varphi(2),\cdots,\varphi(k)]^T \quad (7)$$

则对应不同k值，由(1)式可得一线性方程组，写成矩阵形式就是：

$$\zeta(k)=\phi(k)\cdot\theta(k)+\varepsilon(k) \quad (8)$$

求最小二乘参数估值就是求这个线性方程组的最小二乘解，其法方程为：

$$\phi^T\phi\,\hat{\theta}=\phi^T\zeta \quad (9)$$

最小二乘的递推法，就是在法方程的基础上导出的。

对（8）求最小二乘解的另一途径是用*Householder*变换（下面简称为H变换）。这种解法不是由法方程解，而是直接对方程组

$$\phi\ \theta=\zeta \quad (10)$$

施以H变换，变换成三角阵后，通过回代过程求出最小二乘解[1]。

从数学上来说，这两种解法，即解法方程法与H变换法是完全等价的。但由于计算机的字长有限，因而对病态情况的反应亦不同。这里顺便说一下，有的文献对病态的解释是$\phi^T\phi$阵接近奇异。这种说法不够严格。文献[2]中对此给出反例作了说明。评价病态应当看其条件数的大小。

按法方程求解时，线性方程组是(9)式，而按H变换法求解时，线性方程组是(10)，两式系数阵的条件数不同，一般(9)式条件数远大于(10)式[1]。这样，原生问题也许本身病态并不严重，经$\phi^T\phi$相乘后变得严重了，这就增加了解题时的困难。据文献介绍，一般来说，用H变换求解，是目前最有效的方法。

三 H变换法

我们这里介绍的是一种在线算法。设ϕ经

---

* 收稿时间1981年7月10日．

〔注〕 本文中 θ,φ,ζ,ε,η 为黑体。

变换为上三角阵$\boldsymbol{R}$，ξ变换为r。每取得一组新数据，重新构造$\boldsymbol{R}$的最后一行，这样递推k步后变为：

$$\begin{bmatrix}\boldsymbol{R}(k)_{2n\times 2n}\\ \boldsymbol{O}_{1\times 2n}\end{bmatrix}\hat{\theta}(k)=\begin{bmatrix}\eta(k)_{2n\times 1}\\ *\end{bmatrix} \tag{11}$$

解此三角方程得出$\hat{\theta}(k)$后，采取新数据，重新构造最后一行：

$$\begin{bmatrix}\boldsymbol{R}(k)_{2n\times 2n}\\ \varphi^T(k+1)_{1\times 2n}\end{bmatrix}\hat{\theta}(k+1)=\begin{bmatrix}\eta(k)_{2n\times 1}\\ y(k+1)\end{bmatrix} \tag{12}$$

再施以H变换，变换为：

$$\begin{bmatrix}\boldsymbol{R}(k+1)_{2n\times 2n}\\ \boldsymbol{O}_{1\times 2n}\end{bmatrix}\hat{\theta}(k+1)=\begin{bmatrix}\eta(k+1)_{2n\times 1}\\ *\end{bmatrix} \tag{13}$$

解此三角方程，便可求得$\hat{\theta}(k+1)$。就这样可进行在线计算。

每递推一步，要进行2n次变换。其中第k步第i次变换阵为

$$\boldsymbol{H}(k,i)=\begin{pmatrix}\boldsymbol{I}_{i-1} & & & \\ & c_i & & s_i\\ & & \boldsymbol{I}_{2n-i-1} & \\ & -s_i & & c_i\end{pmatrix} \tag{14}$$

其中未注明的元素全为0，I为单位阵。这第k步是指象（4）式的方程得到k个；第i次变换完成后，则第i行以上为上三角阵。因为每次采取的新数据都在最后一行，故$-s_i$是在H阵的最后一行。

为编写程序方便，列向量（η向量）是放到Φ阵（R阵）中作为最后一列，递推公式为：

$$\boldsymbol{R}(k)_{2n+1,j}=\varphi(k)_j \tag{15}$$

$$\lambda(k)_i=\sqrt{\boldsymbol{R}^2(k,i-1)_{i,i}+\boldsymbol{R}^2(k,i-1)_{2n+1,i}} \tag{16}$$

$$c(k)_i=\frac{\boldsymbol{R}(k,i-1)_{i,i}}{\lambda(k)_i} \tag{17}$$

$$s(k)_i=\frac{\boldsymbol{R}(k,i-1)_{2n+1,i}}{\lambda(k)_i} \tag{18}$$

$$\boldsymbol{R}(k,i)_{i,j}=c(k)_i\boldsymbol{R}(k,i-1)_{i,j}+S(k)_i\boldsymbol{R}(k,i-1)_{2n+1,j} \tag{19}$$

$$\boldsymbol{R}(k,i)_{2n+1,j}=-s(k)_i\boldsymbol{R}(k,i-1)_{i,j}+c(k)_i\boldsymbol{R}(k,i-1)_{2n+1,j} \tag{20}$$

$$\hat{\theta}(k+1)_{2n}=\frac{\boldsymbol{R}(k,2n)_{2n,2n+1}}{\boldsymbol{R}(k,2n)_{2n,2n}} \tag{21}$$

$$\hat{\theta}(k+1)_i=\frac{\boldsymbol{R}(k,2n)_{i,2n+1}-\sum\limits_{m=i+1}^{2n}\hat{\theta}(k+1)_m\boldsymbol{R}(k,2n)_{i,m}}{\boldsymbol{R}(k,2n)_{i,i}} \tag{22}$$

为把问题表达清楚，这里每个变量都用了较多的下标，其中$\boldsymbol{R}(k,i)$表示第k步经第i次变换后的$\boldsymbol{R}$阵，显然$\boldsymbol{R}(k+1,0)=\boldsymbol{R}(k,2n)$。而在编写程序时，应当少用下标以节省内存并使运算速度加快。

在递推公式（15）—（22）中，$\boldsymbol{R}$阵是$2n+1$行。若将$\boldsymbol{R}$阵变为$2n+2$行，每次采取新数据后，构造$\boldsymbol{R}$阵的第$2n+2$行，则计算终了时，$\boldsymbol{R}$阵元素$\boldsymbol{R}_{2n,2n}$就是损失函数（残差平方和）。此时只需将公式（15）—（22）中的$2n+1$改为$2n+2$即可。另外，亦可对$\boldsymbol{R}$阵进行简单计算求出方差阵，即$(\Phi^T\Phi)^{-1}$。

以上递推公式时，递推时初值可选$\boldsymbol{R}(0)=\alpha I$，其中$I$为单位阵，α是个很小的数，如$\alpha=10^{-7}$。选$\boldsymbol{R}(0)$为0阵是不行的，这样做会使分母为0而无法运算。

H变换法的主要缺点是计算量较大。每递推一步要开方$2n$次。但在目前计算机技术飞速发展情况下，此缺点就不那么突出了。

参 考 文 献

〔1〕 冯康，数值计算方法，国防出版社，1978．

〔2〕 武汉大学、山东大学，计算方法，人民教育出版社。

〔3〕 Strejc, V., Least Squares Parameter Estimation, Automatica, 1980.