

RGIA 算法的 QR 分解实现

孟晓风 王行仁

(北京航空航天大学自动控制系 北京 100083)

摘要 提出递推辨识算法一般形式(简称 RGIA)的 QR 分解实现方法,并推导出相应的基于 HOUSEHOLDER 变换的快速递推算法. 该算法可用于各种递推辨识算法中,以减小增益矩阵计算的误差积累和传递,提高辨识精度,同时减少了计算所需的运算量. 数值仿真结果表明该算法是正确的.

关键词 系统辨识, 参数估计, QR 分解, 多变量系统, HOUSEHOLDER 变换, 快速递推算法.

1 引言

递推辨识算法的一般形式描述为^[1,2]

$$\bar{z}(k) = z(k) - \hat{z}(k), \quad (1a)$$

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)\bar{z}(k), \quad (1b)$$

$$K(k) = \rho(k)R^{-1}(k)\Psi(k)\hat{\Sigma}^{-1}(k), \quad (1c)$$

$$\hat{\Sigma}(k) = \hat{\Sigma}(k-1) + \rho(k)[\bar{z}(k)\bar{z}^T(k) - \hat{\Sigma}(k-1)], \quad (1d)$$

$$R(k) = R(k-1) + \rho(k)[\Psi(k)\hat{\Sigma}^{-1}(k)\Psi^T(k) - R(k-1)], \quad (1e)$$

$$x(k+1) = A(\hat{\theta}(k))x(k) + B(\hat{\theta}(k))\bar{u}(k), \quad (1f)$$

$$\begin{bmatrix} \hat{z}(k+1) \\ \text{col}\Psi^T(k+1) \end{bmatrix} = C[\hat{\theta}(k)]x(k+1), \quad (1g)$$

$$\bar{u}(k) = [z^T(k), u^T(k)]^T. \quad (1h)$$

其中 $z(k) \in R^{m \times 1}$ 和 $u(k) \in R^{r \times 1}$ 为过程的输入输出数据; $\hat{\theta}(k) \in R^{N \times 1}$ 表示 k 时刻的参数估计值; $\hat{z}(k)$, $\bar{z}(k)$, $\Psi(k)$ 和 $x(k)$ 分别代表 k 时刻的输出预报值、新息、输出预报值关于参数 θ 的导数和预报模型的状态变量; $\rho(k)$ 是收敛因子; $R(k)$ 是 Hessian 矩阵的近似式; $K(k)$ 为增益矩阵; $\hat{\Sigma}(k)$ 是输出残差 $\epsilon(k, \theta_0)$ 协方差的估计值. RGIA 算法把解决辨识问题的思想统一起来, 归结成输出预报值及其关于参数 θ 梯度的推导问题, 许多递推算法都可以看作它的一种特例. RGIA 算法不仅在理论上具有重要意义, 而且是算法收敛性分析的有力工具^[1,2].

RGIA 算法包括四部分内容: 1) 计算输出预报误差 $\hat{z}(k)$; 2) 计算增益矩阵 $K(k)$; 3) 修正参数估计值; 4) 计算输出预报值 $\hat{z}(k)$ 及其关于 θ 的梯度. 这里 1)–3) 的计算所涉及的前 5 个式子具有固定的结构形式, 与具体所用的模型类无关, 因此可以看作一般的数值计算问题进行研究. 其中增益矩阵 $K(k)$ 的计算是 RGIA 算法的关键, 它涉及到矩阵 $\hat{\Sigma}(k)$ 和

$R(k)$ 的求逆运算,这给算法的实现带来困难.为了解决这一难题,就要设法给出不含矩阵逆运算的等价形式.通常采用的方法是利用矩阵反演公式,将(1c)–(1e)式化成另一种不含矩阵逆运算的等价形式(简称 RCKE)^[1].正如一般辨识递推算法那样,RCKE 算法中 $P(k)$ 的计算误差会出现累积、传递现象,尤其是参数个数较多时更是如此.

本文将讨论 RGIA 算法的 QR 分解实现方法,并推导出相应的基于 HOUSEHOLDER 变换(简称 H -变换)的计算公式.

2 RGIA 算法的 QR 分解实现原理

定义

$$P(k) = \rho(k)R^{-1}(k), \quad \Pi(k) = \rho(k)\hat{\Sigma}^{-1}(k), \quad (2)$$

则(1c),(1d)和(1e)式可分别改写成

$$K(k) = P(k)\Psi(k)\Pi(k)/\rho(k), \quad (3a)$$

$$\Pi(k) = [\mu(k)\Pi^{-1}(k-1) + \bar{z}(k)\bar{z}^r(k)]^{-1} \quad (3b)$$

$$P(k) = [\mu(k)P^{-1}(k-1) + \Psi(k)\Pi(k)\Psi^r(k)/\rho(k)]^{-1}, \quad (3c)$$

其中 $\mu(k) = \rho(k-1) \cdot (1-\rho(k))/\rho(k)$.

对于 $k-1$ 时刻的正定矩阵 $\Pi(k-1)$,存在上三角矩阵 $R_{\Pi}(k-1)$,使

$$\Pi^{-1}(k-1) = R_{\Pi}^r(k-1) \cdot R_{\Pi}(k-1). \quad (4)$$

利用 $k-1$ 时刻的上三角矩阵 $R_{\Pi}(k-1)$ 和 k 时刻的新息向量 $\bar{z}(k)$,构造矩阵

$$F_{\Pi}(k) = [\lambda(k)R_{\Pi}^r(k-1) | \bar{z}(k)]^r. \quad (5)$$

式中 $\lambda^2(k) = \mu(k)$.对于 $F_{\Pi}(k)$,存在正交变换阵 $T_{\Pi}(k)$,使其上三角化,即

$$T_{\Pi}(k) \cdot F_{\Pi}(k) = [R_{\Pi}^r(k) | O]^r. \quad (6)$$

于是, k 时刻的矩阵 $\Pi(k)$ 可以表示为

$$\begin{aligned} \Pi(k) &= [\mu(k)R_{\Pi}^r(k-1)R_{\Pi}(k-1) + \bar{z}(k)\bar{z}^r(k)]^{-1} \\ &= \left[\begin{array}{c|c} \lambda(k)R_{\Pi}^r(k-1) & \lambda(k)R_{\Pi}^r(k-1) \\ \bar{z}^r(k) & \bar{z}^r(k) \end{array} \right]^{-1} \\ &= (F_{\Pi}^r(k)[T_{\Pi}^r(k)T_{\Pi}(k)]F_{\Pi}(k))^{-1} \\ &= R_{\Pi}^{-1}(k)R_{\Pi}^{-r}(k). \end{aligned} \quad (7)$$

因此,(3b)式的计算可由(5)–(7)式代替.

同理,对 $P(k-1)$ 有上三角矩阵 $R_p(k-1)$,使

$$P^{-1}(k-1) = R_p^r(k-1)R_p(k-1). \quad (8)$$

利用(6)式的计算结果 $R_{\Pi}(k)$ 和 K 时刻的输出预报值关于参数 θ 的导数 $\Psi(k)$,构造矩阵

$$\Phi(k) = \Psi(k)R_{\Pi}^{-1}(k)/\delta(k). \quad (9)$$

再利用 $\Phi(k)$ 和 $k-1$ 时刻的上三角矩阵 $R_p(k-1)$,构造矩阵

$$F_p(k) = [\lambda(k)R_p^r(k-1), \Phi(k)]^r, \quad (10)$$

其中 $\delta^2(k) = \rho(k)$.则有正交变换阵 $T_p(k)$,使其上三角化,即

$$T_p(k)F_p(k) = [R_p^r(k) | O]^r. \quad (11)$$

将(7)和(8)式代入(3c)式,得

$$P(k) = [\mu(k)R_p^r(k-1)R_p(k-1) + \Psi(k)R_{\bar{H}}^{-1}(k)R_{\bar{H}}^{-r}(k)\Psi^r(k)/\rho(k)]^{-1}. \quad (12)$$

由(9)–(12)式得

$$\begin{aligned} P(k) &= \left[\begin{array}{c|c} \lambda(k)R_p(k-1) & \lambda(k)R_p(k-1) \\ \hline \Phi^r(k) & \Phi^r(k) \end{array} \right]^{-1} \\ &= [F_P^r(k)[T_P^r(k)T_P(k)]F_P(k)]^{-1} \\ &= R_P^{-1}(k)R_P^{-r}(k). \end{aligned} \quad (13)$$

因此,(3c)式的计算可由(9)–(13)式代替. 将(7)和(13)式代入(3a)式,并考虑到(9)式,得

$$\begin{aligned} K(k) &= R_P^{-1}(k)R_P^{-r}(k)\Psi(k)R_{\bar{H}}^{-1}(k)R_{\bar{H}}^{-r}(k)/\rho(k) \\ &= R_P^{-1}(k)R_P^{-r}(k)\Phi(k)R_{\bar{H}}^{-r}(k)/\delta(k). \end{aligned} \quad (14)$$

将(14)式代入(1b)式,得

$$\theta(k) = \theta(k-1) + R_P^{-1}(k)R_P^{-r}(k)\Phi(k)R_{\bar{H}}^{-r}(k)\bar{z}(k)/\delta(k). \quad (15)$$

综上所述,RGIA 算法中(1b)–(1e)式的计算可由(5),(6),(9)–(11)和(15)式代替.

定理 1. 在矩阵方程(11)中,若将正交矩阵 $T_p(k)$ 分块为

$$T_P(k) = \begin{bmatrix} G_{11}(k) & G_{12}(k) \\ G_{21}(k) & G_{22}(k) \end{bmatrix}, \quad (16)$$

其中 $G_{11}(k) \in R^{N \times N}$, $G_{12}(k) \in R^{N \times 1}$, $G_{21}(k) \in R^{m \times N}$, $G_{22}(k) \in R^{m \times m}$, 则各子矩阵块满足

$$R_P(k) = G_{11}^{-r}(k)\lambda(k)R(k-1), \quad (17a)$$

$$R_P(k) = G_{11}(k)\lambda(k)R_p(k-1) + G_{12}(k)\Phi^r(k), \quad (17b)$$

$$R_P^{-r}(k)\Phi(k) = G_{12}(k). \quad (17c)$$

证明. 根据分块矩阵求逆公式^[1]以及矩阵 $T_P(k)$ 的正交性,有

$$\begin{aligned} \begin{bmatrix} G_{11}(k) & G_{12}(k) \\ G_{21}(k) & G_{22}(k) \end{bmatrix}^r &= \begin{bmatrix} G_{11}(k) & G_{12}(k) \\ G_{21}(k) & G_{22}(k) \end{bmatrix}^{-1} \\ &= \begin{bmatrix} A^{-1} & -A^{-1}G_{12}(k)G_{22}^{-1}(k) \\ -G_{22}^{-1}(k)G_{21}(k)A^{-1} & G_{22}^{-1}(k) + G_{22}^{-1}(k)G_{21}(k)A^{-1}G_{12}(k)G_{22}^{-1}(k) \end{bmatrix}, \end{aligned} \quad (18)$$

式中

$$A = G_{11}(k) - G_{12}(k)G_{22}^{-1}(k)G_{21}(k). \quad (19)$$

由(18)式得

$$G_{11}^{-r}(k) = G_{11}(k) - G_{12}(k)G_{22}^{-1}(k)G_{21}(k). \quad (20)$$

由 $T_P^r(k)T_P(k) = I$ 得

$$G_{11}^r(k)G_{12}(k) + G_{21}^r(k)G_{22}(k) = 0. \quad (21)$$

将(16)式分块表达代入(11)式,比较等式两端可得(17b)式,并且

$$G_{21}(k)\lambda(k)R_p(k-1) + G_{22}(k)\Phi^r(k) = 0. \quad (22)$$

由(22)式得

$$\Phi^r(k) = -G_{22}^{-1}(k)G_{21}(k)\lambda(k)R_p(k-1). \quad (23)$$

将(23)式代入(17b)式,并考虑到(20)式,得

$$\begin{aligned} R_P(k) &= (G_{11}(k) - G_{12}(k)G_{22}^{-1}(k)G_{21}(k))\lambda(k)R_p(k-1) \\ &= G_{11}^{-r}(k)\lambda(k)R_p(k-1), \end{aligned} \quad (24)$$

(17a)式得证. 根据(23), (17a)和(21)式, 有

$$\begin{aligned} R_P^{-r}(k)\Phi(k) &= G_{11}(k)R_P^{-r}(k-1)[-G_{22}^{-1}(k)G_{21}(k)R_P(k-1)]^r \\ &= -G_{11}(k)G_{21}^r(k)G_{22}^{-r}(k) = G_{12}(k). \end{aligned} \quad (25)$$

证毕.

记

$$e(k-1) = R_P(k-1) \cdot \hat{\theta}(k-1), \quad (26)$$

$$v(k) = R_{\bar{H}}^{-r}(k)(\bar{z}(k) + \Psi^r(k)\hat{\theta}(k-1))/\delta(k), \quad (27)$$

$$(G_{11}(k) | G_{12}(k)(\lambda(k)e^r(k-1) | v^r(k)) = e(k), \quad (28)$$

其中 $v(k) \in R^{m \times 1}$, $e(k-1) \& e(k) \in R^{N \times 1}$.

定理 2. RGIA 算法 k 时刻数据的参数估计值为

$$\hat{\theta}(k) = R_P^{-1}(k)e(k). \quad (29)$$

证明. 将(17c)式代入(15)式, 得

$$\hat{\theta}(k) = \hat{\theta}(k-1) + R_P^{-1}(k)G^{12}(k)R_{\bar{H}}^{-r}(k)\bar{z}(k)/\delta(k). \quad (30)$$

将(17b)式代入(30)式, 得

$$\begin{aligned} \hat{\theta}(k) &= R_P^{-1}(k)(R_P(k)\hat{\theta}(k-1) + G_{12}(k)R_{\bar{H}}^{-r}(k)\bar{z}(k)/\delta(k)) \\ &= R_P^{-1}(k)([G_{11}(k)\lambda(k)R_P(k-1) + G_{12}(k)\Phi^r(k)]\hat{\theta}(k-1) \\ &\quad + G_{12}(k)R_{\bar{H}}^{-r}(k)\bar{z}(k)/\delta(k)). \end{aligned} \quad (31)$$

将(9)式代入(31)式, 并考虑到(26)–(28)式, 得

$$\begin{aligned} \hat{\theta}(k) &= R_P^{-1}(k)(G_{11}(k) : G_{12}(k)) \cdot \begin{bmatrix} \lambda(k)R_P(k-1)\hat{\theta}(k-1) \\ R_{\bar{H}}^{-r}(k)(\bar{z}(k) + \Psi^r(k)\hat{\theta}(k-1))/\delta(k) \end{bmatrix} \\ &= R_P^{-1}(k)(G_{11}(k) | G_{12}(k))(\lambda(k)e^r(k-1) | v^r(k))^r \\ &= R_P^{-1}(k)e(k). \end{aligned} \quad (32)$$

证毕.

综上所述, RGIA 的正交变换递推算法可表示为

(1a) 式,

$$T_{\bar{H}}(k) \begin{bmatrix} \lambda(k)R_{\bar{H}}(k-1) \\ \bar{z}^r(k) \end{bmatrix} = \begin{bmatrix} R_{\bar{H}}(k) \\ 0 \end{bmatrix}, \quad (33a)$$

$$\Phi(k) = \Psi(k)R_{\bar{H}}^{-1}(k)/\delta(k), \quad (33b)$$

$$v(k) = [R_{\bar{H}}^{-1}(k)/\delta(k)][\bar{z}(k) + \Psi^r(k)\hat{\theta}(k-1)], \quad (33c)$$

$$T_P(k) \begin{bmatrix} \lambda(k)R_P(k-1) & \lambda(k)e(k-1) \\ \Phi^r(k) & v(k) \end{bmatrix} = \begin{bmatrix} R_P(k) & e(k) \\ 0 & \varepsilon(k) \end{bmatrix}, \quad (33d)$$

$$\hat{\theta}(k) = R_P^{-1}(k)e(k), \quad (33e)$$

(1f) – (1h) 式.

初始条件

$$\begin{cases} R_P^r(0)R_P(0) = P^{-1}(0), \\ R_{\bar{H}}^r(0)R_{\bar{H}}(0) = \hat{\Sigma}^{-1}(0)/\rho(0), \\ e(0) = R_P(0)\theta(0). \end{cases} \quad (34)$$

若算法无需加权,取 $\Sigma^{-1}(k) = I_m$,则有 $R_p^{-1}(k)/\delta(k) = I$,此时(33*)式可简化为

$$\begin{cases} \text{(1a) 式,} \\ v(k) = \bar{z}(k) + \Psi^r(k)\hat{\theta}(k-1), & (35a) \\ T_p(k) \begin{bmatrix} \lambda(k)R_p(k-1) & \lambda(k)e(k-1) \\ \Phi^r(k) & v(k) \end{bmatrix} = \begin{bmatrix} R_p(k) & e(k) \\ 0 & \varepsilon(k) \end{bmatrix}, & (35b) \\ \hat{\theta}(k) = R_p^{-1}(k)e(k), & (35c) \\ \text{(1f) - (1h) 式.} \end{cases}$$

3 RGIA 算法的 H-变换快速递推算法

对于 SISO 系统,(33d)式中的 $\Phi(k)$ 和(33a)式中的 $\bar{z}(k)$ 都是向量,直接利用文献[4]提出的 H-变换快速递推公式即可实现(33a)和(33d)式所需的正交变换.

对于 MIMO 过程,为了利用 H-变换快速递推公式,向信息阵($\lambda(k)R_p(k-1) | \lambda(k)e(k-1)$)添加新信息($\Phi^r(k) | v(k)$)时,可采用逐行添加方式,具体推导如下所述.

定理 3. 用 $N \times N$ 维分块矩阵

$$T = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \quad (36)$$

构造 $(N+m) \times (N+m)$ 维分块矩阵

$$T' = \begin{bmatrix} T_{11} & 0 & T_{12} & 0 \\ 0 & I_{i-1} & 0 & 0 \\ T_{21} & 0 & T_{22} & 0 \\ 0 & 0 & 0 & I_{m-i+1} \end{bmatrix}, \quad (37)$$

其中 I_n 表示 n 阶单位阵.若 T 是正交矩阵,则 T' 也是正交矩阵.

证明.显然,由 $T^r T = I_N$,可推导出 $T'^r T' = I_{m+N}$.

证毕.

记

$$\begin{cases} \Phi(k) = [\varphi_1(k), \varphi_2(k), \dots, \varphi_m(k)], \\ v(k) = [v_1(k), v_2(k), \dots, v_m(k)]^r, \end{cases} \quad (38)$$

$$R_0(k) = \lambda(k)R_p(k-1); e_0(k) = \lambda(k)e(k-1). \quad (39)$$

定理 4. 递推算式(33d)中的正交变换 $T_p(k)$ 可用如下的 m 步正交变换实现,构造 $(N+1)$ 阶信息阵 $F_i(k)$ 及相应的正交变换阵 $T_i(k)$ 为

$$\begin{cases} F_i(k) = \begin{bmatrix} R_{i-1}(k) & e_{i-1}(k) \\ \varphi_i^r(k) & v_i(k) \end{bmatrix}, \\ T_i(k)F_i(k) = \begin{bmatrix} R_i(k) & e_i(k) \\ 0 & \varepsilon_i(k) \end{bmatrix}, \\ i = 1, 2, \dots, m. \end{cases} \quad (40)$$

其中 $R_i(k)$ 为 N 阶上三角阵, $e_i(k)$ 为 N 维向量, $\varepsilon_i(k)$ 为标量.

证明.用(40b)式所定义的正交矩阵 $T_i(k)$,按(37)式构造矩阵 $T_{i,p}(k)$,并取

$$T_p(k) = T_{m,p}(k)T_{m-1,p}(k)\dots T_{1,p}(k). \quad (41)$$

根据定理 3 及正交矩阵性质,知 $T_p(k)$ 为正交矩阵. 并且,用归纳法可证明

$$T_p(k) \begin{bmatrix} R_0(k) & e_0(k) \\ \Psi^r(k) & v(k) \end{bmatrix} = \begin{bmatrix} R_m(k) & e_m(k) \\ 0 & \varepsilon(k) \end{bmatrix}, \quad (42)$$

其中

$$\varepsilon(k) = [\varepsilon_1(k), \varepsilon_2(k), \dots, \varepsilon_m(k)]^r. \quad (43)$$

证毕.

(40)式中所需的正交变换可直接利用文献[4]提出的 H-变换快速递推公式实现.

综上所述,MIMO 过程的 RGIA 的正交变换递推算法表示为

$$\left\{ \begin{array}{l} \text{(1a) 式,} \\ T_{II}(k) \begin{bmatrix} \lambda(k)R_{II}(k-1) \\ \bar{z}^r(k) \end{bmatrix} = \begin{bmatrix} R_{II}(k) \\ 0 \end{bmatrix}, \quad (44a) \\ \Phi(k) = \Psi(k)R_{II}^{-1}(k)/\delta(k), \quad (44b) \\ v(k) = [R_{II}^{-1}(k)/\delta(k)][\bar{z}(k) + \Psi^r(k)\hat{\theta}(k-1)], \quad (44c) \\ R_0(k) = \lambda(k)R_m(k-1), e_0(k) = \lambda(k)e_m(k-1), \quad (44d) \\ \text{(40) 式} \\ \hat{\theta}(k) = R_m^{-1}(k)e_m(k), \quad (44e) \\ \text{(1f) - (1h) 式.} \end{array} \right.$$

初始条件同(33*).

4 性能比较

当加权阵取 $\Sigma(k)=1$,且过程为单变量时,几种计算方法的计算复杂性比较,如表 1 所示.

表 1 几种计算增益矩阵或上三角信息阵方法的计算复杂性比较

算 法	运 算 量		
	加、减法	乘、除法	开方
RCKE 算法	$1.5N^2 + 3.5N$	$1.5N^2 + 4.5N$	0
U-D 分解	$N^3/6 + 1.5N^2 + 8N/6$	$N^3/6 + 1.5N^2 + 44N/6$	0
本文方法	$N^2 + 3N$	$N^2 + 9N$	0

考虑 SISO 模型

$$A(z^{-1}) \cdot y(k) = B(z^{-1}) \cdot u(k) + B(z^{-1}) \cdot w(k), \quad (45)$$

其中 $\{w(k)\}$ 是均值为零的随机不相关噪声序列,且

$$\begin{aligned} A(z^{-1}) &= 1.0 + a_1z^{-1} + \dots + a_{na}z^{-na}, \\ B(z^{-1}) &= b_1z^{-1} + b_2z^{-2} + \dots + b_{nb}z^{-nb}, \\ D(z^{-1}) &= 1.0 + d_1z^{-1} + \dots + d_{nd}z^{-nd}. \end{aligned}$$

根据(45)式,可以推导出输出预报值的表达式为

$$\hat{y}(k|\theta) = h^r(k, \theta) \cdot \theta, \quad (46)$$

其中

$$\mathbf{h}(k, \theta) = (-y(k-1), \dots, -y(k-na), u(k-1), \dots, u(k-bn), \varepsilon(k-1, \theta), \dots, \varepsilon(k-nd, \theta))^T, \quad (47)$$

$$\theta = (a_1, \dots, a_{na}, b_1, \dots, b_{bn}, d_1, \dots, d_{nd})^T. \quad (48)$$

输出预报值关于参数 θ 的梯度可写成

$$\boldsymbol{\psi}(k) = (-z_f(k-1), \dots, -z_f(k-na), u_f(k-1), \dots, u_f(k-bn), \varepsilon_f(k-1, \theta), \dots, \varepsilon_f(k-nd, \theta))^T, \quad (49)$$

其中

$$\begin{cases} z_f(k) = y(k) - \hat{d}_1(k)z_f(k-1) - \dots - \hat{d}_{nd}(k)z_f(k-nd), \\ u_f(k) = u(k) - \hat{d}_1(k)u_f(k-1) - \dots - \hat{d}_{nd}(k)u_f(k-nd), \\ \varepsilon_f(k) = \varepsilon(k) - \hat{d}_1(k)\varepsilon_f(k-1) - \dots - \hat{d}_{nd}(k)\varepsilon_f(k-nd). \end{cases} \quad (50)$$

输出残差可写成

$$\varepsilon(k, \theta) = y(k) - \hat{y}(k|\theta) = \hat{e}(k). \quad (51)$$

当加权阵取为单位阵时, RGIA 算法退化为递推的极大似然法(RML)

$$\begin{cases} \tilde{y}(k) = y(k) - \mathbf{h}^T(k) \cdot \hat{\boldsymbol{\theta}}(k-1), \\ \hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \mathbf{K}(k) \cdot \tilde{y}(k), \\ \mathbf{k}(k) = \mathbf{P}(k) \cdot \boldsymbol{\psi}(k) / S(k), \\ \mathbf{P}(k) = (\mathbf{P}(k-1) - \mathbf{K}(k) \cdot S(k) \cdot \mathbf{k}^T(k)) / \mu(k), \\ S(k) = \boldsymbol{\psi}^T(k) \cdot \mathbf{P}(k-1) \cdot \boldsymbol{\psi}(k) + \mu(k), \\ (47), (51), (49) \text{ 和 } (50) \text{ 式.} \end{cases} \quad (52)$$

相应的正交变换算法表达为(称为 RHT-ML)

$$\begin{cases} \tilde{y}(k) = y(k) - \mathbf{h}^T(k) \cdot \hat{\boldsymbol{\theta}}(k-1), \\ v(k) = \tilde{y}(k) + \boldsymbol{\psi}^T(k) \cdot \hat{\boldsymbol{\theta}}(k-1), \\ T_p(k) \begin{bmatrix} \lambda(k)R_p(k-1) & \lambda(k)\mathbf{e}(k-1) \\ \boldsymbol{\Psi}^T(k) & v(k) \end{bmatrix} = \begin{bmatrix} R_p(k) & \mathbf{e}(k) \\ 0 & \sigma(k) \end{bmatrix}, \\ \hat{\boldsymbol{\theta}} = R_p^{-1}(k) \cdot \mathbf{e}(k), \\ (51), (47), (50) \text{ 和 } (49) \text{ 式.} \end{cases} \quad (53)$$

仿真例. 在(45)式中, 输入量 $u(k)$ 采用幅度为 1.0 的伪随机二进制序列, $w(k)$ 的方差为 1.0. 在 RML 算法(52)式和 RHT-ML 算法(53)式中, 取递推初值 $R_p(0) = 10^{-6}I$, $P(0) = 10^6I$, $\hat{\boldsymbol{\theta}}(0) = 0$, $\boldsymbol{\psi}(0) = 0$, $\mathbf{h}(0) = 0$, 遗忘因子取($\lambda(k) = \mu(k)$)

$$\mu(k) = 0.99 * \mu(k-1) + 0.01, \mu(0) = 0.95. \quad (54)$$

表 2 和 3 分别给出了采用单精度和双精度运算时两种算法的辨识结果比较. 结果表明, RML 算法对运算的舍入误差十分敏感, 用单精度运算(4 字节的浮点运算)根本辨识不了参数, 而必须采用双精度(8 字节的浮点)运算; 对于 RHT-ML 算法, 无论是采用单精度或双精度浮点运算, 都可获得很好的辨识结果. 可见, 与 RHT 算法相比, RHT-ML 算法的数值稳定性好、辨识精度高.

表 2 采用单精度运算时,RML 和 RHT-ML 两种算法的辨识结果比较

参数(真值)	RML				RHT-ML			
	L=100	L=500	L=1000	L=2000	L=100	L=500	L=1000	L=2000
a1(-1.6)	-1.3584	-1.0060	-1.7439	5.3229	-1.2629	-1.5423	-1.6525	-1.5946
a2(1.61)	1.4109	1.5757	1.8040	-22.228	1.3247	1.5659	1.6505	1.6033
a3(-0.776)	-0.5542	-0.5759	-0.9019	2.2523	-0.4620	-0.7228	-0.8271	-0.7681
b1(1.2)	1.0983	2.3263	1.0705	-6.1893	1.0626	1.2049	1.2397	1.1949
b2(-0.95)	-0.5459	-5.2059	-0.3766	88.404	-0.4947	-0.8665	-1.0346	-0.9889
b3(0.2)	0.2312	2.4412	1.8009	24.646	0.0807	0.2669	0.1636	0.1713
d1(0.1)	0.3658	-0.2385	0.2984	-0.3495	0.0379	0.0777	0.1041	0.0939
d2(0.25)	0.0691	0.4781	-0.0026	-6.2295	0.1573	0.2249	0.1978	0.2175
d3(0.875)	0.3277	0.9521	0.0043	-2.9693	0.2574	0.7346	0.7362	0.8200

表 3 采用双精度运算时,RML 和 RHT-ML 两种算法的辨识结果比较

参数(真值)	RML				RHT-ML			
	L=100	L=500	L=1000	L=2000	L=100	L=500	L=1000	L=2000
a1(-1.6)	-1.3638	-1.5684	-1.5982	-1.5998	-1.4087	-1.5696	-1.5989	-1.6002
a2(1.61)	1.4137	1.5833	1.6081	1.6104	1.4504	1.5842	1.6085	1.6106
a3(-0.776)	-0.5593	-0.7452	-0.7749	-0.7767	-0.5998	-0.7463	-0.7755	-0.7770
b1(1.2)	1.1002	1.1882	1.2297	1.2095	1.0716	1.1875	1.2293	1.2095
b2(-0.95)	-0.5522	-0.9098	-0.9538	0.9514	-0.6352	-0.9125	-0.9551	-0.9518
b3(0.2)	0.2256	0.2179	0.2138	0.2138	0.2183	0.2164	0.2135	0.2138
d1(0.1)	0.3621	0.1100	0.1121	0.0831	0.3424	0.1108	0.1124	0.0833
d2(0.25)	0.0590	0.2026	0.2335	0.2325	0.0768	0.2015	0.2334	0.2325
d3(0.875)	0.3264	0.7217	0.8158	0.8501	0.3578	0.7246	0.8168	0.8502

参 考 文 献

- [1] 方崇智, 萧德云. 过程辨识. 北京: 清华大学出版社, 1989, 424—430.
- [2] Ljung L, Soderstrom T. Theory and practice of recursive identification, Cambridge Mass: MIT Press, 1983.
- [3] Ljung S, Ljung L. Error propagation properties of recursive least squares adaptation algorithms. *Automatica*, 1985, **21**: 157—167.
- [4] 孟晓风, 王行仁, 黄俊钦. 最小二乘估计的 Householder 变换快速递推算法. 自动化学报, 1994, **20**(1): 20—27.
- [5] Bierman G J. Factorization. Methods for Discrete Sequential Estimation, New York: Academic Press, 1977.

IMPLEMENTATION OF RECURSIVE GENERAL IDENTIFICATION ALGORITHM VIA QR DECOMPOSITION

MENG XIAOFENG WANG XINGREN

(Dept. of Automatic Control, Beijing University of Aero. and Astro. Beijing 100083)

Abstract An implementation method with QR decomposition for recursive general identification algorithm (RGIA) is proposed and a corresponding fast recursive algorithm based on HOUSEHOLDER transformation is developed in this paper. This algorithm can be used in all sorts of identification methods to get better numerical stability and less numerical complexity. Finally, simulation study on this algorithm is made, which shows that this implementation method is correct.

Key words System identification, parameter estimation, QR decomposition, multivariable system, HOUSEHOLDER transformation, fast recursive algorithm.

孟晓风 1995 年在北京航空航天大学获博士学位, 现任该校自控系副教授. 目前的研究方向是参数估计、容错技术、飞行控制与飞行仿真.

王行仁 1955 年北京航空学院研究生毕业. 现任北京航空航天大学自控系教授、博士生导师、国防科工委科技委兼职委员和军用仿真技术专业组副组长. 1985 年获国家科技进步一等奖. 目前的研究方向是飞行控制与飞行仿真、容错技术.