第1卷 第4期　控制理论与应用　Vol. 1, No.4
1984年10月　CONTROL THEORY AND APPLICATIONS　October, 1984

有理分式矩阵求逆的一种新方法

王　世　林
（中国科学院系统科学研究所）

摘　要

本文通过多项式矩阵系数矩阵的简单运算，给出了有理分式矩阵求逆的一种新方法。这种方法可在计算机上实现。

一、前　　言

在多变量控制系统的设计和网络理论中，经常遇到有理分式矩阵求逆的问题。从理论上讲，这是一个不难的问题。利用Gauss消去法化成Smith－MacMillan形即可解决。但在实际工程中，利用上述方法求出有理分式矩阵的逆矩阵，基本上是行不通的。因为用这种方法求出的逆矩阵的每个元素的分子分母中都可能有大量的公根，如果算法不恰当，利用计算机求出公根，往往是很困难的。为了避开这一弱点，A. J. J. Vander Weiden和J. Vandewalle等人分别在1977年和1980年[1,2]给出了有理分式矩阵求逆方法.前者是根据系统理论方法，后者是用Laurent级数展开式方法。他们的方法均可借助计算机求出逆矩阵，但都存在着算法过于复杂，计算量较大的缺点。Vander Weiden方法要利用两次极小实现程序，而J. Vandewalle和J. Van Daele的方法在G(s)的阶数或主项阶数较高时，循环计算逆矩阵的Laurent级数展开式系数所用矩阵的阶数迅速增加，因此，计算量也随着增加。况且这种方法并没有给出所求逆矩阵的具体表达式。故从实际应用的观点来看，这两种方法并不理想。

本文基于两个多项式矩阵系数矩阵之间的简单运算，诸如求两个多项式矩阵右公因，两个多项式矩阵相除，多项式矩阵的正则化、首一化，给出了有理分式矩阵求逆的新方法。这种方法，均是对多项式矩阵的系数矩阵进行“+”、“－”、“×”运算。因此，它在多变量控制系统的设计中，特别是利用Rosenbrock的逆Nyquist基阵法设计多变量系统时是很有用处的，且比其它求逆方法简单，易掌握。

二、几个引理和基本算法

设$P(s)$、$Q(s)$是多项式矩阵，$U(s)$是行初等变换阵的乘积，记

本文于1983年2月25日收到。1983年10月19日收到修改稿。

$$U(s)=\begin{pmatrix} U_{11}(s) & U_{12}(s) \\ U_{21}(s) & U_{22}(s) \end{pmatrix},\quad Z(s)=\begin{pmatrix} P(s) \\ Q(s) \end{pmatrix}.$$

引理 1 若 $U(s)Z(s)=\left(\dfrac{D(s)}{0}\right)$，则$D(s)$为$P(s)$、$Q(s)$的最大右公因。

证 见[3]，若取

$$P(s)=\begin{pmatrix} P_1(s) & & & \\ & P_2(s) & & \\ & & \ddots & \\ & & & P_m(s) \end{pmatrix},\quad Q(s)=(q_1(s)q_2(s)\cdots q_m(s)),$$

由引理 1 知

$$\begin{pmatrix} U_{11}(s) & U_{12}(s) \\ U_{21}(s) & U_{22}(s) \end{pmatrix}\begin{pmatrix} P(s) \\ Q(s) \end{pmatrix}=\begin{pmatrix} D(s) \\ \cdots\cdots \\ 0 \end{pmatrix}.$$

$U_{22}(s)$是$p_1(s),p_2(s),\cdots,p_m(s)$的最小公倍式。

基算本法 1：求两个多项式矩阵的最大右公因和有理分式阵按列通分算法，见[4]P23—24。

引理 2 设 $\det P(s)\neq 0$，则恒可经过列和行的初等变换，将它变成列正则首一矩阵。

证明见[3]。

基本算法 2：满秩阵的正则化、首一化算法，见[4]P20—22。

引理 3 设$G(s)=Q(s)P^{-1}(s)$，$P(s)$为列首一矩阵，$Q(s)$、$P(s)$右互质（即最大右公因为单位么阵），那么，

$$\left\{\begin{aligned} &\begin{pmatrix} \dot{x}_\nu \\ \dot{x}_{\nu-1} \\ \vdots \\ \dot{x}_1 \end{pmatrix}=\begin{pmatrix} 0 & (I_\nu 0) & & \\ & 0 & (I_{\nu-1}0) & \\ & & \ddots & \\ & & & (I_2 0) \\ -A_\nu & -A_{\nu-1} & \cdots & -A_1 \end{pmatrix}\begin{pmatrix} x_\nu \\ x_{\nu-1} \\ \vdots \\ x_1 \end{pmatrix}+\begin{pmatrix} 0 \\ 0 \\ \vdots \\ I \end{pmatrix}u \\ &y=(C_\nu C_{\nu-1}\cdots C_1)\begin{pmatrix} x_\nu \\ x_{\nu-1} \\ \vdots \\ x_1 \end{pmatrix} \end{aligned}\right. \tag{*}$$

是$G(s)$的一个状态空间的最小实现。其中，

$$(A_\nu A_{\nu-1}\cdots A_1 I)、(C_\nu C_{\nu-1}\cdots C_1)$$

是$P(s)$、$Q(s)$按列次降幂排列的系数矩阵。

证　由于$Q(s)$、$P(s)$右互质，而$P(s)$和I左互质，故由$P(s)$、$Q(s)$、I所表示的系统

$$\begin{cases} P(D)z - Q(D)u = 0 \\ z \qquad\qquad = y \end{cases}$$

没有输入解耦零点和输出解耦零点。而(*)又是$G(s)=Q(s)P^{-1}(s)$的一个实现，所以它也是$G(s)$的一个最小实现。

引理 4　设$Q(s)$是$m\times m$列正则阵，$P(s)$为$l\times m$多项式矩阵，则必存在多项式矩阵$N(s)$、$\bar{P}(s)$满足$P(s)\doteq N(s)Q(s)+\bar{P}(s)$，$\partial_{l_i}\bar{P}(s)<\partial_{l_i}Q(s)$。

上述引理为大家所熟知，这里就不证明了。

基本算法3：多项式矩阵除法算法。

1)　设$Q(s)$的列次为$d_1\geq d_2\geq\cdots\geq d_m$，记$\nu=d_1$，$k=\max\limits_{i=1,2,\cdots,m}\partial_{l_i}P(s)$，$l=k-d_1$；

2)　计算

$$n_i=\sum_{j=1}^{m}\operatorname{sign}(d_j+1-i),\quad i=0,1,\cdots,\nu;$$

3)　构造矩阵

$$S_i=\begin{bmatrix} s^{d_1-i} & & & 0 \\ & s^{d_2-i} & & \\ & & \ddots & \\ 0 & & & s^{d n_i-i} \end{bmatrix},\qquad i=0,1,\cdots,\nu;$$

$$S_{-j}=\begin{bmatrix} s^{d_1+j} & & & 0 \\ & s^{d_2+j} & & \\ & & \ddots & \\ 0 & & & s^{d_m+j} \end{bmatrix},\qquad j=1,2,\cdots,l;$$

4)　记$I_i(i=0,1,\cdots,\nu)$为n_i阶单位矩阵，将$P(s)$、$Q(s)$表成

$$Q(s)=Q_0S_0+Q_1S_1(I_1 0)+\cdots+Q_{\nu-1}S_{\nu-1}(I_{\nu-1}0)+Q_\nu(I_\nu 0),$$

$$\begin{aligned} P(s)=&P_{-l}S_{-l}+P_{-l+1}S_{-l+1}+\cdots+P_{-1}S_{-1}+P_0S_0 \\ &+P_1S_1(I_1 0)+\cdots+P_{\nu-1}S_{\nu-1}(I_{\nu-1}0)+P_\nu(I_\nu 0); \end{aligned}$$

5)　计算

$$N_j=\left(P_{-l+j}-\sum_{\alpha=0}^{j-1}N_\alpha Q_{j-\alpha}(I_{j-\alpha}0)\right)Q_0^{-1},\ j=0,1,\cdots,l,$$

注： 当 $j=0$ 时，$\sum\limits_{\alpha=0}^{j-1}N_\alpha Q_{j-\alpha}(I_{j-\alpha}0)=0$，当 $j-\alpha<\nu$ 时，$R_{j-\alpha}=0$，

$$\overline{P}_j=P_j-\sum_{\alpha=0}^{l}N_\alpha Q_{l+j-\alpha}(I_{l+j-\alpha}0),\ j=1,2,\cdots,\nu,$$

注：当 $l+j-\alpha>\nu$ 时，$Q_{l+j-\alpha}=0$；

6） 计算

$$N(s)=N_0s^l+N_1s^{l-1}+\cdots+N_l$$

$$\overline{P}(s)=\sum_{\alpha=1}^{\nu}\overline{P}_\alpha S_\alpha(I_\alpha 0);$$

7）结束。这时 $N(s)$是商，$\overline{P}(s)$是余项，也就是说

$$P(s)=N(s)Q(s)+\overline{P}(s).$$

三、有理分式矩阵求逆的步骤

给定有理分式矩阵

$$G(s)=\begin{pmatrix}\dfrac{q_{11}(s)}{p_{11}(s)} & \dfrac{q_{12}(s)}{p_{12}(s)} & \cdots & \dfrac{q_{1m}(s)}{p_{1m}(s)}\\ \dfrac{q_{21}(s)}{p_{21}(s)} & \dfrac{q_{22}(s)}{p_{22}(s)} & \cdots & \dfrac{q_{2m}(s)}{p_{2m}(s)}\\ \cdots & \cdots & & \\ \dfrac{q_{m1}(s)}{p_{m1}(s)} & \dfrac{q_{m2}(s)}{p_{m2}(s)} & \cdots & \dfrac{q_{mm}(s)}{p_{mm}(s)}\end{pmatrix},$$

其中 $p_{ij}(s)$、$q_{ij}(s)$为多项式。

1. 根据引理1，按照基本算法 1，将有理分式矩阵 $G(s)$按列通分（也可手算通分），

$$\begin{bmatrix} \dfrac{q_{1i}(s)}{p_{1i}(s)} \\ \dfrac{q_{2i}(s)}{p_{2i}(s)} \\ \vdots \\ \dfrac{q_{mi}(s)}{p_{mi}(s)} \end{bmatrix}^{\tau} = (q_{1i}(s)q_{2i}(s)\cdots q_{mi}(s)) \begin{bmatrix} p_{1i}(s) & & & \\ & p_{2i}(s) & & \\ & & \ddots & \\ & & & p_{mi}(s) \end{bmatrix}^{-1}$$

$$= -(U^i_{22})^{-1}(U^i_{21}) \quad i=1,2,\cdots,m.$$

故

$$G(s) = -\begin{bmatrix} U^1_{21}(s) \\ U^2_{21}(s) \\ \vdots \\ U^m_{21}(s) \end{bmatrix}^{\tau} \begin{bmatrix} U^1_{22}(s) & & & \\ & U^2_{22}(s) & & \\ & & \ddots & \\ & & & U^m_{22}(s) \end{bmatrix}^{-1}. \qquad (1)$$

注： 如果$G(s)$的阶数低或$p_{1i}(s)$，$p_{2i}(s)$，$\cdots$，$p_{mi}(s)$的最小公倍式比较好求，可直接将传递矩阵写成（1）的形式。

2． 根据引理4，按照基本算法3，可求出

$$G^{-1}(s) = \begin{bmatrix} U^1_{22}(s) & & & \\ & U^2_{22}(s) & & \\ & & \ddots & \\ & & & U^m_{22}(s) \end{bmatrix} (U^1_{21}(s)^{\tau}\ U^2_{21}(s)^{\tau} \cdots U^m_{21}(s)^{\tau})^{-1}$$

$$= N(s) + \overline{P}(s)(U^1_{21}(s)^{\tau}\ U^2_{21}(s)^{\tau} \cdots U^m_{21}(s)^{\tau})^{-1}.$$

3． 根据引理1，按照基本算法1求出$\overline{P}(s)$与$(U^1_{21}(s)^{\tau}\ U^2_{21}(s)^{\tau} \cdots U^m_{21}(s)^{\tau})$的最大右公因，即

$$\overline{P}(s) = P_1(s)D(s),$$

$$(U^1_{21}(s)^{\tau}\ U^2_{21}(s)^{\tau} \cdots U^m_{21}(s)^{\tau}) = Q_1(s)D(s).$$

4． 根据引理2，按照基本算法2，同时对$P_1(s)$和$Q_1(s)$作相同的列变换，使得$Q_2(s)$为列正则阵，即

$$P_1(s)V(s)=P_2(s),$$
$$Q_1(s)V(s)=Q_2(s).$$

再按基本算法2对$Q_2(s)$进行行变换，即

$$BQ_2(s)=Q_3(s),$$

使得$Q_3(s)$为列首一阵。

这时，

$$G^{-1}(s)=N(s)+P_2(s)Q_3^{-1}(s)B.$$

5. 根据引理3立即可知

$$G^{-1}(s)=N(s)+C(sI-A)^{-1}B,$$

其中

$$C=(C_\nu C_{\nu-1}\cdots C_1),$$

$$A=\begin{bmatrix} 0 & (I_\nu 0) & & \\ & & (I_{\nu-1}0) & \\ & & & \ddots \\ & & & (I_2 0) \\ -A_\nu & -A_{\nu-1} & \cdots & -A_1 \end{bmatrix},$$

(A,B,C)为对应于传递矩阵$W_1(s)=P_2(s)Q_3^{-1}(s)B$的最小实现，$C$为$P_2(s)$按列次降幂排列的系数矩阵，$(A_\nu A_{\nu-1}\cdots A_1 I)$为$Q_3(s)$按列次降幂排列的系数矩阵。

6. 利用Souriau－Frame－Faddeev公式可知

$$G^{-1}(s)=N(s)+C(sI-A)^{-1}B$$
$$=N(s)+\frac{1}{\Delta(s)}(s^{n-1}CB+s^{n-2}CB_{n-2}B+\cdots+sCB_1B+CB_0B),$$

其中

$$\Delta(s)=S^n+a_{n-1}S^{n-1}+\cdots+a_1S+a_0,$$

$$a_{n-1}=-tr(AB_{n-1}),\qquad B_{n-1}=I_n,$$
$$\qquad B_{n-2}=a_{n-1}I_n+AB_{n-1},$$
$$a_{n-2}=-\frac{1}{2}tr(AB_{n-2}),\qquad B_{n-3}=a_{n-2}I_n+AB_{n-2},$$
$$\vdots\qquad\qquad\vdots$$
$$a_1=-\frac{1}{n-1}tr(AB_1),\qquad B_0=a_1I_n+AB_1,$$
$$a_0=-\frac{1}{n}tr(AB_0).$$

7. 再根据引理1可消去 $\Delta(s)$ 和矩阵

$$(s^{n-1}CB+s^{n-2}CB_{n-2}B+\cdots+sCB_1+CB_0B)$$

各元素之间的公因式。

根据上述计算步骤，我们用 Fortran 4 编制了有理分式矩阵求逆的程序并对一些例子进行计算。例如对

$$G(s)=\begin{bmatrix}\dfrac{1}{2s+1} & \dfrac{1}{1+3s}\\[2ex] \dfrac{1}{s+1} & \dfrac{1}{1+2s}\end{bmatrix}$$

的逆矩阵进行了手算和在 Hp－1000 上进行计算，其结果均为

$$G^{-1}(s)=\begin{bmatrix}-6s-11-\dfrac{6}{s}-\dfrac{1}{s^2} & 4s+8+\dfrac{5}{s}+\dfrac{1}{s^2}\\[2ex] 12s+16+\dfrac{7}{s}+\dfrac{1}{s^2} & -6s-11-\dfrac{6}{s}-\dfrac{1}{s^2}\end{bmatrix}$$

四、结　论

上述有理分式求逆的方法，完全是多项式矩阵的系数矩阵初等变换的运算，可在计算机上实现，而且矩阵阶数越高，这种算法的优越性越明显。不仅如此，这种方法中的三个基本算法也都大同小异，没有多大本质的变化。因此，这种求逆方法是与 A. J. J. Vander Weiden 和 J. Vandewalle 等人的方法不同的一种新方法。

参 考 文 献

〔1〕 Van der Weiden, A. J. J., Inversion of Rational Matrices, Int. J. Control, **25**, 3(1977), 393—402.

〔2〕 Vandewalle, J. and Daele, J., On the Computation of the Inverse of a Rational Matrix Via Expansion, Int. J. Control, **31**, 1(1980), 95—107.

〔3〕 Kailath, T., Linear Systems, 1980, Pretce－Hall, Inc. Englewood Cliffs, N. J. 07632, 345—494.

〔4〕 韩京清、许可康，线性定常控制系统中的一些有关问题，华南工学院讲义（1983）。

A NEW METHOD FOR THE INVERSION OF A RATIONAL MATRIX IN THE DESIGN OF CONTROL SYSTREMS

Wang Shilin

(Institute of Systems Science, Academia Sinica, Beijing)

Abstract

This paper presents a new method for the inversion of a rational matrix. It is based on the elementary operations on the coefficient matrices of two polynomial matrices, and is consequently easy to be programmed on a computer.