第4卷 第4期 控制理论与应用 Vol.4, No.4
1987年12月 CONTROL THEORY AND APPLICATIONS Dec., 1987

求多输入多输出有理传递函数的最小实现的一种算法

金亿创 孙增圻
（清华大学，北京）

摘 要

本文叙述了多输入多输出有理传递函数到状态方程的最小实现的方法.在所述的算法中用SVD分解求得最小实现的维数，再求解一系列相当于系数矩阵为上三角阵的线性方程组，得出最小实现的状态方程系数矩阵 A、B、C。

一、原 理

设 $G(s)$是输出为 r 维输入为 m 维的多输入多输出有理传递函数矩阵，并设 $G(s)$是严格正则的（否则可先解出最小实现的矩阵 D）.因为 $G(s)$是严格正则的，它能分解成下列形式：

$$G(s)=J_0s^{-1}+J_1s^{-2}+J_2s^{-3}+\cdots \tag{1}$$

记 α_i 为 $G(s)$的第 i 行分母最小公倍式的阶，β_j 为 $G(s)$的第 j 列分母最小公倍式的阶。将 $G(s)=C(sI-A)^{-1}B$ 的级数展开式与（1）式比较有

$$(J_l)_{\mu\nu}=c^{\mu}A^{(l)}b_{\nu}.$$

（用上标 i 表示第 i 行，下标 j 表示第 j 列，上标加括号表示幂，即 c^{μ} 表示 C 的第 μ 行，b_{ν} 表示 B 的第 ν 列，$A^{(l)}$表示 A 的 l 次幂，下同）。

令

$$H_{\mu\nu}(i,\ j)=\begin{pmatrix}(J_0)_{\mu\nu} & (J_1)_{\mu\nu} & \cdots & (J_{j-1})_{\mu\nu}\\ (J_1)_{\mu\nu} & (J_2)_{\mu\nu} & \cdots & (J_j)_{\mu\nu}\\ \vdots & \vdots & & \vdots\\ (J_{i-1})_{\mu\nu} & (J_i)_{\mu\nu} & & (J_{i+j-2})_{\mu\nu}\end{pmatrix}.$$

由 $H_{\mu\nu}(i,\ j)$组成矩阵 $\widetilde{S}$

$$\widetilde{S}=\begin{pmatrix}H_{11}(\alpha_1,\beta_1) & H_{12}(\alpha_1,\beta_2) & \cdots & H_{1m}(\alpha_1,\beta_m)\\ H_{21}(\alpha_2,\beta_1) & H_{22}(\alpha_2,\beta_2) & \cdots & H_{2m}(\alpha_2,\beta_m)\\ \vdots & \vdots & & \vdots\\ H_{r1}(\alpha_r,\beta_1) & H_{r2}(\alpha_r,\beta_2) & & H_{rm}(\alpha_r,\beta_m)\end{pmatrix}.$$

本文于1985年11月2日收到.1986年5月19日收到修改稿。

再将$\tilde{S}$扩展成S

$$S=\begin{pmatrix} & (J\beta_1)_{11} & \cdots & (J\beta_m)_{1m} \\ & \vdots & & \vdots \\ & (J_{\alpha_1+\beta_1-1})_{11} & \cdots & (J_{\alpha_1+\beta_m-1})_{1m} \\ \tilde{S} & (J\beta_1)_{21} & \cdots & (J\beta_m)_{2m} \\ & \vdots & & \vdots \\ & (J_{\alpha_2+\beta_1-1})_{21} & \cdots & (J_{\alpha_2+\beta_m-1})_{2m} \\ & \vdots & & \vdots \\ & (J_{\alpha_r+\beta_1-1})_{r1} & \cdots & (J_{\alpha_r+\beta_m-1})_{rm} \end{pmatrix}.$$

根据参考文献[1]的说明，S的秩就是最小实现的维数（记为N），并且对所有可能的分解

$$S=H_{\alpha\times N}P_{N\times\beta+m}\qquad\left(\alpha=\sum_{i=1}^{r}\alpha_i,\quad \beta=\sum_{j=1}^{m}\beta_j\right)$$

都有最小实现的A、B、C使

$$H=\begin{pmatrix} c^1 \\ c^1A \\ \vdots \\ c^1A^{(\alpha_1-1)} \\ c^2 \\ \vdots \\ c^2A^{(\alpha_2-1)} \\ \vdots \\ c^rA^{(\alpha_r-1)} \end{pmatrix}.$$

$$P=[b_1Ab_1\cdots A^{(\beta_1-1)}\ b_1\cdots b_m\cdots A^{(\beta_m-1)}b_m\ A^{(\beta_1)}b_1\ A^{(\beta_2)}\ b_2\cdots A^{(\beta_m)}b_m].\quad(2)$$

二、算　法

1．采用SVD算法将S分解成H和P的积（只要求保存P），且同时获得最小实现的维数。

一般来说S矩阵是比较大的，占用较多的内存，在对S阵进行SVD分解并求出矩阵P的过程中不应该再另开与S大小相近的数组。所以SVD分解要采用一定的技巧，分解过程是：

1° TR数组存贮矩阵S。用一系列文献[2]所述的Householder变换把S矩阵分解成上双对角矩阵。数组TS保留上双对角各元素，并且用TR数组保留所有的对S矩阵进

行 Householder 变换的矩阵。

任何一个 Householder 变换都能被一个向量所确定，因此保留了这一向量就保留了 Householder 变换阵。在对 S 矩阵（TR 数组）的第 i 行进行 Householder 变换后，这行正好能保存进行变换的 Householder 变换阵。设这些矩阵分别为 R_1，R_2，…，R_t。这样就有：$T\cdot S\cdot R_1R_2\cdots R_t$ = 上双对角阵。其中 T 是所有对 S 矩阵列进行 Householder 变换的矩阵之积。

2° 用数组 TR 来存贮 R_1，R_2，…，R_t 之积。

现在 TR 数组的第 i 行决定着 R_i 矩阵，而 R_i 是对角块矩阵，从第一行到第 $i-1$ 行、第一列到第 $i-1$ 列的对角块是单位阵，因此用 TR 存贮 R_i，R_{i+1}，…，R_t 之积可以不动原来 TR 数组第一行到第 $i-1$ 行的内容。这样，即使不另开与 TR 数组大小相同的数组也能用 TR 来恢复并存贮 R_1，R_2，…，R_t 之积，其恢复次序是：

$$R_t,\quad R_tR_{t-1},\quad \cdots,\quad R_tR_{t-1}\cdots R_1。$$

3° 用 Givens 转动和 QR 迭代把上双对角阵对角化，并且把所有对上双对角阵进行的列变换对应的行变换（即行变换矩阵是列变换矩阵之逆）作用到数组 TR。那么 S 分解成

$$S=U^T\cdot A\cdot V。$$

A 是对角阵，对角元素就是 S 的奇异值。V 阵是变换后数组 TR 的内容。

4° 最后非零奇异值个数即为最小实现的维数（记为 N）。非零奇异值乘矩阵 V 中相应的行组成的矩阵即为 P。

2。为了求解 A、B、C 的方便，用 Householder 变换将 P 阵变换成新的 P。

1° $j\leftarrow 1$，$i\leftarrow 1$；

2° 第 j 列中第 i 行到末行元素均为零时转3°，否则转4°；

3° $j\leftarrow j+1$，转 2°；

4° $r_i\leftarrow j$，进行 Householder 变换，使 $p_{i,j}\neq 0$，$p_{k,j}=0$ $k\in[i+1,\ N]$；

5° $j\leftarrow j+1$，$i\leftarrow i+1$；

6° i 大于 N 时结束，否则转 2°。

原来的 P 是 S 的一种分解，即有 $H_{\alpha\times N}$ 使

$$S=H_{\alpha\times N}\cdot P。$$

设这里的所有 Householder 变换矩阵之积为 T。因为 T 是正交对称阵，所以有 $T\cdot T=I$。

$$S=H_{\alpha\times N}\cdot P=(H_{\alpha\times N}\cdot T)\cdot(T\cdot P),$$

所以新的 P 阵也是 S 的一种分解。

3。从 P 中解出 A、B、C。

1° 求 B。

$$b_i=p_{k_i};\ k_i=\sum_{j=1}^{i-1}\beta_j+1\quad i=1,\ 2,\ \cdots,\ m;$$

2° 求 A。

从 P 的表示式（2）看出：若存在 k 使 $r_i=\sum_{j=1}^{k}\beta_j$，那么 $A\cdot p_{r_i}=p_{\beta+k}$，否则有 $Ap_{r_i}=p_{r_{i+1}}$。

令
$$\tau_i=\begin{cases}\beta+k & \text{若有 } k \text{ 使 } r_i=\sum_{j=1}^{k}\beta_j,\\ r_i+1 & \text{其它}\end{cases}$$

$$A\cdot[p_{r_1}\ p_{r_2}\cdots p_{r_N}]=[p_{\tau_1}\ p_{\tau_2}\cdots p_{\tau_N}]. \qquad (3)$$

由于（3）式中$[p_{r_1}\cdots p_{r_N}]$是上三角阵，且对角线元素非零（从第 2 步p_{r_i}的构成求得此结论），所以从（3）式很容易解出矩阵 A；

3° 求 C。

要求得 C 只有 P 是不行的，需要在对 S 变换前保留其中的 r 行，设用 CS 数组保留，有

$$cs^i=s^{k_i},\quad k_i=\sum_{j=1}^{i-1}\alpha_i+1,\ i=1,\ 2,\ \cdots,\ r.$$

有了 CS 可以建立如下方程：

$$C\cdot[p_{r_1}\ p_{r_2}\cdots p_{r_N}]=[cs_{r_1}\ cs_{r_2}\cdots cs_{r_N}],$$

从而很容易解出矩阵 C。

三、举　例

$$G(s)=\begin{pmatrix}\dfrac{s}{(s+1)^2} & \dfrac{1}{s^2}\\[2ex] \dfrac{1}{s+1} & \dfrac{1}{s+2}\end{pmatrix}.$$

$$g_{11}(s)=\frac{s}{(s+1)^2}=s^{-1}-2s^{-2}+3s^{-3}-4s^{-4}+5s^{-5}-6s^{-6}+\cdots,$$

$$g_{12}(s)=\frac{1}{s^2}=s^{-2},$$

$$g_{21}(s)=\frac{1}{s+1}=s^{-1}-s^{-2}+s^{-3}-s^{-4}+\cdots,$$

$$g_{22}(s)=\frac{1}{s+2}=s^{-1}-2s^{-2}+4s^{-3}-8s^{-4}+16s^{-5}-\cdots,$$

此例：$\alpha_1=4$，　$\alpha_2=2$，

$\beta_1=2$，　$\beta_2=3$。

组成的 S 矩阵：

$$\begin{pmatrix} 1 & -2 & 0 & 1 & 0 & 3 & 0 \\ -2 & 3 & 1 & 0 & 0 & -4 & 0 \\ 3 & -4 & 0 & 0 & 0 & 5 & 0 \\ -4 & 5 & 0 & 0 & 0 & -6 & 0 \\ 1 & -1 & 1 & -2 & 4 & 1 & -8 \\ -1 & 1 & -2 & 4 & -8 & -1 & 16 \end{pmatrix}$$

SVD 分解后产生的 P：

-1.906973E+00 2.098671E+00 -2.184443E+00 4.428025E+00 -8.895504E+00
-2.290367E+00 1.779101E=01
5.279229E+00 -7.182046E+00 -6.447889E-01 7.483106E-01 -9.268888E
-01 9.084860E+00 1.853777E+00
-6.808615E-01 1.126307E-01 -1.838439E-01 8.321735E-01 1.029227E-01
4.556005E-01 -2.058455E-01
1.365084E-01 -3.029731E-02 8.799038E-01 3.584783E-01 -1.089950E
-02 -7.591413 E-02 2.179878 E-02
-1.047211E-01 1.320924E-02 6.651963E-02 -1.076615E-01 -1.290440E
-02 7.830248E-02 2.580879E-02

经 Householder 变换后产生的新 P：

5.656852E+00 -7.424621E+00 1.767781E-01 -8.838849E-01 2.121325E+00
9.192383E+00 -4.242647E+00
0.000000E+00 9.354146E-01 1.403121E+00 -2.739429E+00 4.008922E+00
-1.870829E+00 -8.017841E+00
0.000000E+00 0.000000E+00 -1.999999E+00 2.999996E+00 -6.999996E+00
8.853516E-07 1.399999E+01
0.000000E+00 0.000000E+00 0.000000E+00 -1.927246E+00 3.187371E+00
-1.341108E -07 -6.374740E+00
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 5.188743E-01
1.743272E-07 -1.037748E+00

最后得出最小实现的系数矩阵 A、B、C：

矩阵 A（状态矩阵）：

-1.312500E+00 -5.905766E-01 -8.839345E-02 2.031070E-01 -6.879579E-01
1.653596E-01 -6.874996E-01 9.020080E-01 2.253457E-01 -3.218190E-02
0.000000E+00 9.464804E-07 -1.499998E+00 1.297190E+00 -1.223071E+00
0.000000E+00 -1.433705E-07 9.636232E-01 -1.538486E-01 1.659346E+00
0.000000E+00 1.863635E-07 1.307453E -07 -2.692311E-01 -3.461511E-01

矩阵 B（控制矩阵）：

```
5.656852E+00  1.767781E-01
0.000000E+00  1.403121E+00
0.000000E+00 -1.999999E+00
0.000000E+00  0.000000E+00
0.000000E+00  0.000000E+00
```

矩阵 C（输出矩阵）：

```
1.767768E-01 -7.349679E-01 -4.999995E-01 -3.335615E-01 2.594409E-01
1.767768E-01  3.340769E-01 -2.499998E-01  9.265628E-02 4.632788E-01
```

四、结　束　语

本文所述的算法采用了SVD分解，因此比文献〔1〕介绍的方法的稳定性和可靠性要好.算法中的所有变换全是正交变换，结果有较好的精度.

参 考 文 献

〔1〕 Pal Rozsa and Naresh K. Sinha, Efficient Algorithm for Irreducible Realization of a Rational Matrix Int. J. Control, **20**, 5, (1974), 739-751.

〔2〕 黄琳，系统与控制理论中的线性代数，科学出版社，北京，（1984）.

〔3〕 George E. Forsythe, Michael A. Malcolm and Cleve B. Moler, Computer Methods for Mathematical Computations, PRENTICE-HALL, INC., (1977), Chapter 3.9.

AN ALGORITHM FOR TRANSFORMATION OF A RATIONAL TRANSFER MATRIX TO MINIMAL REALIZATION OF STATE EQUATIONS

Jin Yichuang, Sun Zengqi

(Tsinghua University, Beijing)

Abstract

In this paper an algorithm for minimal realization of a multi-input and multi-output transfer function matrix is presented. The algorithm uses SVD(Singular Value Decomposition) to compute the dimension of minimal realization. For the computation of matrices A, B, C, a set of linear equations with coefficient matrix of upper triangle is only needed to solve. An example is included to show the procedure of the algorithm.