计算高阶线性自动调节系统稳定性和暂态过程的通用程序

郝柏林　張淑譽

（中国科学院物理研究所）

提　　要

本文介绍一个通用程序。只要按照高阶线性自动调节系统的框图填写原始数据，并写出给定函数的形状，它就自动形成描述整个系统的方程组，消去其中代数部份，求出特征方程的根，判断系统的稳定性，然后求积微分方程部份，印出各个环节输出量随时间的变化。从数学性质上看，这是"矩阵偶"理论的一种具体应用。用 FORTRAN 语言或 ALGOL 语言写出的程序全文，可向有关单位联系。

一、引　　言

虽然自动控制系统的理论和实践已经有了很大的发展(可参看[1])，古典的用传递函数描述的自动调节系统目前仍在工业控制中普遍应用。为了分析各种参数对调节特性的影响，常常要讨论系统的稳定性，以及在单位阶跃(或其它给定函数)作用下，输出量随时间的变化。传统的作法是对描述调节系统的常微分方程作拉氏变换，然后察看总传递函数的零点，并用运算微积的方法求得输出量。当系统的阶较高时，这套手续相当麻烦。因此，只有二、三阶调节系统的性质讨论得比较深透。

生产实践中经常遇到高阶调节系统，特别是愈要使所用的模型接近实际，系统的阶就愈高。另一方面，用电子计算机处理上百阶的常微分方程组其实是不困难的。因此，不一定要想方设法把高阶系统归併成二、三阶系统来近似处理，而可以直接去计算调节系统的稳定性和时间响应。问题在于为每一个具体的调节系统写出全部方程和编制程序，反而是效率不高的手工劳动。

本文介绍一个计算高阶线性自动调节系统稳定性和暂态过程的通用程序*。只要根据调节系统的框图填写原始数据单，它就自动形成方程组并求解。本程序可成为设计调节系统的同志们手中的一个常备工具，每次在经验基础上初步选定各环节参数后，利用这个程序进行分析整定，能省去不少模型试验。将来实现计算机辅助设计的程序系统时，本程序可改造为其中一个模块。

为了便于阅读，我们力求使本文自成一体，从基本公式的推导讲起，一直介绍到程序的框图结构。用 FORTRAN 算法语言写出的程序全文，可向科学院物理所函索。ALGOL 语言的程序，请与冶金部北京钢铁设计院电力科联系。

二、传递函数

一个自动调节系统由很多个"环节"组成。我们只讨论单输入、单输出的环节。第 i 个环节有一个输入量 Z_i 和一个输出量 Y_i。环节的作用就是把输入量变换成输出量：

* 这个程序的一种不完备的方案收入了"FORTRAN 程序设计讲义"第十一章，见[2]。

$$Y_i=K_iZ_i$$

其中 K_i 就是传递函数。假定每个环节的传递函数都可以写成线性分式

$$K_i=\frac{C_i+D_ip}{A_i+B_ip},$$

其中 A_i, B_i, C_i 和 D_i 是给定常数，P 代表微分运算 $\frac{d}{dt}$。每个环节对应一个一阶常微分方程：

$$(A_i+B_ip)Y_i=(C_i+D_ip)Z_i$$

当 $B_i=D_i=0$ 时，这只是一个代数方程。

线性分式包括了许多实践中常见的环节。例如

比例(P)环节　　$Y=KZ,$

积分(I)环节　　$Y=\frac{1}{Tp}Z,$

微分(D)环节　　$Y=TpZ,$

惯性(T)环节　　$Y=\frac{1}{1+Tp}Z,$

以及 PI、PD 等等。式中 K、T 都是常数。K 称为增益或放大系数，T 叫作时间常数。

振荡环节的传递函数是二阶有理分式：

$$Y=\frac{D+Ep}{A+Bp+Cp^2}Z,$$

只要按图 1 所示，将它换成带反馈的等效框图，就化为三个用线性分式描述的环节。

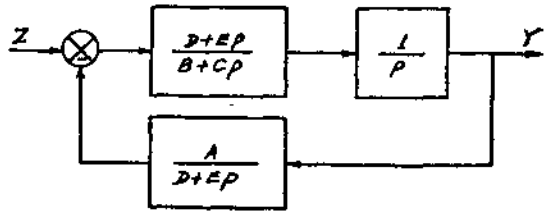

图 1　振荡环节的等效框图

延迟环节

$$Y(t)=Z(t-T)$$

形式上可对 T 展开成泰勒级数并求和

$$Y(t)=\sum_{n=0}^{\infty}\frac{(-T)^n}{n!}\frac{d^n}{dt^n}Z(t)=e^{-Tp}Z(t)$$

可见它的传递函数是无穷阶的。由于

$$e^{-Tp}\cong 1-Tp\cong\frac{1}{1+Tp}$$

时间常数很小的延迟环节，可以作为小惯性环节对待。这是一种常用的近似。其实用线性分式还能更好一些地逼近延迟环节：

$$e^{-Tp}\cong 1-Tp+\frac{(Tp)^2}{2!}\cong\frac{2-Tp}{2+Tp}.$$

在化工流程控制等情形下，会遇到时间常数很大的延迟环节。这时可用高阶有理分式来逼近传递函数 e^{-p}。例如取

$$f_2(p)=\frac{12-6p+p^2}{12+6p+p^2}$$

$$f_3(p)=\frac{120-60p+12p^2-p^3}{120+60p+12p^2+p^3}$$

$$f_4(p)=\frac{1680-840p+180p^2-20p^3+p^4}{1680+840p+180p^2+20p^3+p^4}$$

$$f_5(p)=\frac{30240-15120p+3360p^2-420p^3+30p^4-p^5}{30240+15120p+3360p^2+420p^3+30p^4+p^5}$$

$$f_6(p)=\frac{665280-332640p+75600p^2-10080p^3+840p^4-420p^5+p^6}{665280+332640p+75600p^2+10080p^3+840p^4+420p^5+p^6}$$

等等。它们可以截取 e^{-p} 的连分式表示求得，或对 e^{-p} 的泰勒级数前若干项作 Pade′变换[3]算出。本文不再详述这些变换，仅仅在此指出：无论作为 p 的函数，还是回到时间 t 的表示，这些有理分式都能相当好地逼近原来的传递函数。例如 e^{-1} 与 $f_5(1)$的差别小于 10^{-10}，e^{-5}与 $f_5(5)$的差别也还小于 10^{-4}。每个有理分式可以仿照前面处理振荡环节的办法，化成有限个线性分式构成的等价框图，而每个复杂的传递函数 $K(p)$都可用高阶有理分式逼近。因此本文介绍的算法原则上适用于任意高阶线性自动调节系统。这是用传递函数方法对自动调节系

统作数值分析时值得注意的一条途径。

三、连接矩阵和给定函数

许多环节连接成网络，才能组成调节系统。我们保持每个环节输出量 Y_i 的记号不变，让输入量 Z_i 反映网络的连接情况。每个 Z_i 可能是其它若干个 Y_j 的线性组合：

$$Z_i=\sum_{j=1}^{N}W_{ij}Y_j$$

这时可能遇到几种情形：

（一）输出量 Y_j 直接连到环节 i 的入口上，$W_{ij}=1$。

（二）环节 j 的输出量反馈（或前馈）到环节 i 的入口，$W_{ij}=\beta$。常数 $\beta>0$ 是正反馈，$\beta<0$ 是负反馈。最常见的是 $\beta=\pm1$。

（三）环节 j 的输出量没有连到环节 i 的入口上，$W_{ij}=0$。

W_{ij} 组成一个 N 行 N 列的表，我们称之为连接矩阵。它的零元素很多，因此规定只输入非零元素的角标和数值，但要把非零元素的数目放在前面。

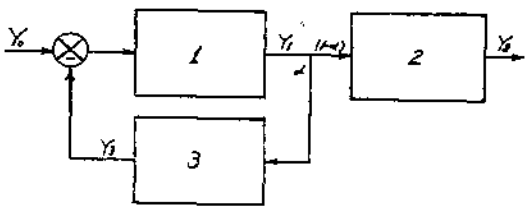

图 2　电流型反馈

通常反馈线的输入端是电压型的量，而输出端是电流型的量，即取出“电压”，馈入“电流”。实际上这只是一些控制信息，不一定对应真正的电流或电压。连接矩阵描述法对于电流型反馈也是适用的，如图 2 所示的反馈环节，相应的连接矩阵元为 $W_{21}=1-\alpha$, $W_{31}=\alpha$, $W_{13}=-1$。

输入量 Z_i 中还允许有一个或多个给定函数，记作 $Y_{0j}(j=1,2,\cdots,M)$。Y_{0j} 可以是阶跃、脉冲、正弦或随机函数等等。因此 Z_i 的一般形式是

$$Z_i=\sum_{j=1}^{N}W_{ij}Y_j+\sum_{j=1}^{M}U_{ij}Y_{0j},$$

$N*M$ 的矩阵 U 乃是反映给定函数 Y_{0j} 分布情况的另一个连接矩阵。这种描述方法允许把同一个给定函数接到许多不同环节上去。最常见的简单情形是 $M=1$，Y_{01} 为线性渐增函数

$$Y_{01}(t)=\begin{cases}1 & (t\geqslant A)\\ t/A & (A>t\geqslant 0)\\ 0 & (t<0)\end{cases}$$

$A=0$ 时这就是阶跃函数。

一般情形下 M 个给定函数可能各不相同，不宜作为程序的通用部份。我们要求单独编写一段程序来处理它们。具体作法与所用的算法语言有关：ALGOL 语言中是一个过程，FORTRAN 语言中是一个独立的子程序块。这一段程序为每个时间值 t，算出各个给定函数及其导数。自动调节系统依赖给定函数的几阶导数，应有明确限制。本文介绍的程序，允许系统依赖于给定函数本身及其一、二阶导数。放宽这一限制的办法将在第五节中指出。

四、形成基本方程组

分别引入具有 N 个和 M 个分量的列矢

$$Y=\begin{bmatrix}Y_1\\ Y_2\\ \vdots\\ Y_N\end{bmatrix},\quad Z=\begin{bmatrix}Z_1\\ Z_2\\ \vdots\\ Z_N\end{bmatrix},\quad Y_0=\begin{bmatrix}Y_{01}\\ \vdots\\ Y_{0M}\end{bmatrix}.$$

于是全部“连接”和“传递”关系表示为

$$Z=WY+UY_0,$$
$$Y=KZ.$$

其中除了两种连接矩阵 W 和 U 外，还有由各个环节传递函数组成的对角矩阵

$$K=(A+Bp)^{-1}(C+Dp)$$

这里 A、B、C、D 是各个环节参数形成的对角矩阵。例如

$$A=\begin{bmatrix}A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_N\end{bmatrix}.$$

把这些关系代入前面两式，得到

$$(A+Bp)Y=(C+Dp)(WY+UY_0).$$

分出微分项即列矢 pY，为此引入两个 $N*N$ 的方阵

$$S=B-DW,\quad R=CW-A,$$

和两个 $N*M$ 的矩阵

$$V_1=CU,\quad V_2=DU.$$

写成分量形式是

$$S_{ij}=B_i\delta_{ij}-D_iW_{ij},\quad (i,j=1,2,\cdots,N)$$
$$R_{ij}=C_iW_{ij}-A_i\delta_{ij},$$
$$(V_1)_{ij}=C_iU_{ij},\qquad \begin{pmatrix} i=1,2,\cdots,N \\ j=1,2,\cdots,M \end{pmatrix}$$
$$(V_2)_{ij}=D_iU_{ij},$$

其中 δ_{ij} 是单位矩阵的分量。于是得出决定 Y 的基本方程组

$$S(pY)=RY+(V_1+V_2p)Y_0$$

这是一阶常微分方程和线性代数方程的混合方程组。许多数值解法都要求把微分方程组化成标准形式，即明显解出一阶导数

$$\frac{dY_i}{dt}=f_i(t,Y_1,Y_2,\cdots Y_N)(i=1,2,\cdots N)$$

从形式上看，似乎只要矩阵 S 可以求逆，则方程组就可以化成标准形式，直接求积。

实际上问题要深刻得多，上述混合方程组的解是由 R 和 S 这一对"矩阵偶"的性质决定的。这是由于方程组中有代数部份，而消去代数部份的过程中又可能出现对给定函数高阶导数的依赖关系。现在就来仔细讨论这个问题。

五、分离代数部份

基本方程组

$$S(pY)=RY+(V_1+V_2p)Y_0$$

的解，由矩阵偶 $R-pS$ 的性质完全决定[4]。矩阵偶要变化成标准形式，才能暴露其本质，这在数学文献中已有详尽的讨论(可参看[5])。下面先略述一般理论所提供的背景，再结合自动调节系统的具体实践，给出程序中的算法。

对于高阶线性自动调节系统所导致的方程组，$R-pS$ 只能是所谓正则(非奇异)矩阵偶，即行列式

$$det|R-pS|\not\equiv 0,$$

因为只有满足这一条件，才能存在从各个给定端到各个输出端的传递函数矩阵。

$$Y=-(R-pS)^{-1}(V_1+V_2p)Y_0,$$

详见本文下一节开头的讨论。

正则矩阵偶可以通过一系列初等变换化成标准形式：对角线上有若干矩阵块，其余元素都等于零的准对角矩阵。其中有一些形状如下的方块：

$$\begin{bmatrix} 1 & -p & 0 & \cdots & 0 & 0 \\ 0 & 1 & -p & \cdots & 0 & 0 \\ \cdots & & & & \cdots & \\ 0 & 0 & 0 & \cdots & 1 & -p \\ 0 & 0 & 0 & & 0 & 1 \end{bmatrix}$$

它有 μ 行 μ 列，整数 μ 是矩阵偶的"无限初等因子"的幂次。每个这类块对应的解，不须经过积分，即由给定函数及其导数的组合决定，表达式中可能含有给定函数的 $\mu-1$ 次导数。只有方阵 S 不能求逆，即 $detS=0$ 时，才有这类块出现。

此外还可能有一个普通 Jordan 正则形式的方阵 $W-pI$，它才对应完全消去了代数部份的微分方程组。

虽然原则上存在两个非奇异方阵 P 和 Q，使得 $P(R-pS)Q$ 具有标准形式，但目前尚未编制出矩阵偶化标准形，同时给出 P 和 Q 的程序。加之使用数值方法计算给定函数的高阶导数，会使精度每况愈下，不能与微分方程部份求积的高阶逼近精度相匹配。因此，我们具体限制了自动调节系统只能依赖于给定函数的

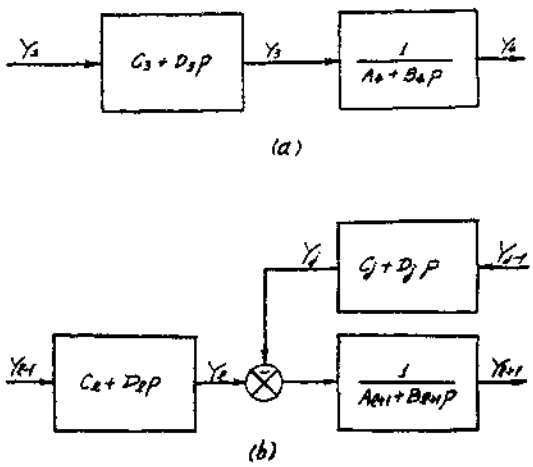

图 3

二阶以内导数。下面介绍的算法（主要是行消去部分）可以推广到任意有限阶导数依赖性，但是所涉及的给定函数的各阶导数，都必须由使用者在一个单独的程序块中提供。

现在来具体分析一个任意的高阶线性自动调节系统中可能遇到的各种特殊情形。这同时也是为正则矩阵偶的一般理论，提供一种形象化的说明。

（一）某个 Y_j 只出现在方程组右端，而左端没有 pY_j。这时 S 矩阵的第 j 列全是零。例如图 3(a)对应的方程

$$Y_3=(C_3+D_3p)Y_2,$$
$$(A_4+B_4p)Y_4=Y_3,$$

就没有 Y_3 的导数，而图 3(b) 的方程中没有 Y_1 和 Y_j 的导数。

当然可以用传递函数的简单变换把没有输出量导数的环节（以及下面将提到的“多余”环节）从调节系统的框图中取消。图 3 的两个例子可以分别变换为图 4(a)和(b)那样。如果要求在准备原始数据之前，对调节系统的框图完成这类变换，对于复杂的系统，这又是一种很容易出错的手工劳动。我们走另一条途径：让

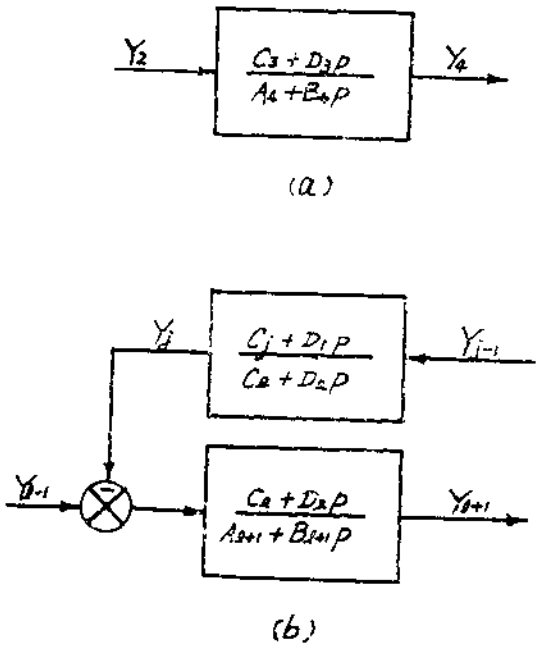

图 4

程序自己在解微分方程组之前，先挑选和消去这些环节，但是输出计算结果时，仍然按原来编号顺序印出一切输出量的值。

下面讨论如何处理 S 矩阵中的零元素列。我们将看到，这时只要按 R 矩阵的相应列，对整个方程组完成一次列主元素消去。

回到基本方程组

$$S(pY)=RY+(V_1+V_2p)Y_0$$

如果某个 pY_j 根本不出现在左端，矩阵 S 的第 j 列就全是零；对于这个固定的 j 有

$$S_{ij}=0 \qquad (i=1,2,\cdots N)$$

矩阵 R 的第 j 列决不会全是零，否则 Y_j 也不出现在方程组中，环节 j 就没有输出量也未与系统发生关系。从 R 的第 j 列中挑选一个绝对值最大的元素 R_{lj}，称为列主元素。对于这个固定的行号 l，写出基本方程组中相应的那一个方程：

$$\sum_{m=1}^{N}{}'S_{lm}(pY_m)=\sum_{m=1}^{N}{}'R_{lm}Y_m+R_{lj}Y_j+$$
$$+\sum_{m=1}^{M}(V_1+V_2p)_{lm}Y_{0m}$$

前两个求和符号上的一撇，表示其中不包含 $m=j$ 的项。主元素 $R_{lj}\neq 0$，可将 Y_j 明显解出来并代入其余的 $N-1$ 个方程。稍加归併得：

$$\sum_{m=1}^{N}{}'\left(S_{lm}-\frac{R_{lj}S_{lm}}{R_{lj}}\right)(pY_m)=$$
$$=\sum_{m=1}^{N}{}'\left(R_{lm}-\frac{R_{lj}R_{lm}}{R_{lj}}\right)Y_m+$$
$$+\sum_{m=1}^{M}{}'\left(V_{1lm}-\frac{R_{lj}V_{1lm}}{R_{lj}}\right)Y_{0m}+$$
$$+\left(V_{2lm}-\frac{R_{lj}V_{2lm}}{R_{lj}}\right)pY_{0m}\Big],$$

求和符号上可以不画撇，因为相应系数自动为零。这样就从方程组中消去了没有导数项的第 j 环节。如果新得到的 S 矩阵中仍有某个未用过的列为零，则重复以上作法。每一次消去，都是套用如下的公式：

$$\tilde{S}_{im}=S_{im}-\frac{R_{ij}S_{lm}}{R_{lj}},$$

$$\tilde{R}_{im}=R_{im}-\frac{R_{ij}R_{lm}}{R_{lj}},$$

$$\tilde{V}_{im}=V_{im}-\frac{R_{ij}V_{lm}}{R_{lj}}.$$

这些式子中的记号 S、$\tilde{S}$ 等，只代表消去前后的矩阵。每次套用时，它们都是不同的。

（二）第 i 个方程中根本没有任何 Y_j 的微分项。例如，比例环节

$$Y_j=CY_{j-1}$$

就不是微分方程。又如最靠近某个给定函数端的一个环节不含有积分，则

$$Y_j=(C+Dp)Y_0,$$

它也没有 pY_j 项。这些情形下 S 矩阵的第 i 行全是零。它导致的“行消去”比前面叙述的“列消去”要复杂一些。首先在基本方程组的给定函数中补写一项：

$$S(pY)=RY+(V_1+V_2p+V_3p^2)Y_0。$$

这里增加了一个 $N*M$ 的矩阵 V_3，它的元素一开始全是零，但在行消去的过程中可能出现非零元素。

设 S 矩阵的第 i 行全是零，于是方程组中第 i 个方程是纯代数方程，其中系数 R_{ij} 不可能全是零。取其绝对值最大者 R_{il}，解出相应 Y_l

$$Y_l=-\sum_{j\neq l}\frac{R_{ij}}{R_{jl}}Y_j-\sum_{j\neq l}\frac{1}{R_{il}}\times$$
$$(V_1+V_2p+V_3p^2)_{ij}Y_{0j},$$

并代入其它各个方程，归併后得行消去的公式：

$$\tilde{S}_{kj}=S_{kj}-S_{kl}\frac{R_{ij}}{R_{il}},$$

$$\tilde{R}_{kj}=R_{kj}-R_{kl}\frac{R_{ij}}{R_{il}},$$

$$(\tilde{V}_1)_{kj}=(V_1)_{kj}-R_{kl}\frac{(V_1)_{ij}}{R_{il}},$$

$$(\tilde{V}_2)_{kj}=(V_2)_{kj}-R_{kl}\frac{(V_2)_{ij}}{R_{il}}+S_{kl}\frac{(V_1)_{ij}}{R_{il}},$$

$$(\tilde{V}_3)_{kj}=(V_3)_{kj}-R_{kl}\frac{(V_3)_{ij}}{R_{il}}+S_{kl}\frac{(V_2)_{ij}}{R_{il}}.$$

这些式子比列消去的公式更清楚地说明，S 和 R 两个矩阵必须同时处理，而且行消去可能带来对给定函数高阶导数的依赖性：原来全是零的 V_3 矩阵中会出现非零元素。

我们看到，没有作消去之前的基本方程组，只含有 pY_0 即给定函数的一阶导数。列消去对应 S 中的全零列，不增加新的微分运算。行消去时 $S(pY)$ 中某个 Y_l 要代以右端表达式，就可能出现 p^2Y_0 项，即给定函数的二阶导数。如果对 S 矩阵实现一遍行消去的过程中，在已经消过的地方又出现新的零行（原则上是可能遇到这种情形的，后面(四)中给一个简例），下一遍行消去就会带来高阶导数 p^3Y，等等。为了正确计入这种情形，必须再引入矩阵 $V_4, V_5\cdots$，事先冲零，在行消去中按下式变换：

$$(\tilde{V}_m)_{kj}=(V_m)_{kj}-R_{kl}\frac{(V_m)_{ij}}{R_{il}}+S_{kl}\frac{(V_{m-1})_{ij}}{R_{il}},$$

$$m=4,5\cdots$$

并在描写给定函数的程序块中提供高阶导数的表达式。由于这一套作法使程序中定义的矩阵数目改变，而在实践中又极少遇到这类蜕化情况，我们没有采取普遍的解决办法，而限于在程序中完成一遍行消去和若干遍列消去。因此程序中只说明了数组 $V_1\sim V_3$，给定函数也只要求提供到二阶导数。

（三）调节系统参数的一定特殊组合，使得 S 矩阵的某些行或列发生线性相关。最简单的例子如图 5 所示的“多余”环节：第 j 和 $j+1$ 两个环节实际上相消，其方程：

$$Y_{j+1}=pY_j$$
$$pY_j=Y_{j-1}$$

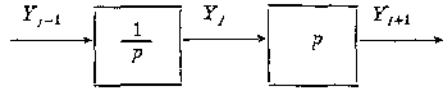

图 5 “多余”环节

使得 S 矩阵中第 j 和第 $j+1$ 两行元素完全相

同。

这类线性相关常常掩盖了 S 中的一些零行或零列。因此，我们在作行消去和列消去之前，先对 S 完成一次消去，使得每行每列中最多只剩下一个非零元素。这是对 S 进行的列主元素消去，如果第 j 列第 i 行的主元素是 S_{ij}，则消去公式是：

$$\tilde{S}_{kl}=S_{kl}-S_{kj}\frac{S_{il}}{S_{ij}},$$

$$\tilde{R}_{kl}=R_{kl}-S_{kj}\frac{R_{il}}{S_{ij}},$$

$$\tilde{V}_{kl}=V_{kl}-S_{kj}\frac{V_{il}}{S_{ij}}.$$

（四）再考察一个导致对给定函数高阶导数依赖关系的特例（图 6），它对应的矩阵偶 $R-pS$ 很简单：

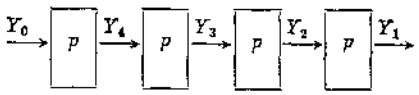

图 6　由纯微分环节组成的调节系统

$$\begin{bmatrix}1 & -p & 0 & 0\\ 0 & 1 & -p & 0\\ 0 & 0 & 1 & -p\\ 0 & 0 & 0 & 1\end{bmatrix}.$$

这就是前面提到的"无限初等因子"的最单纯的表现形式（$\mu=4$），它根本不含有 Jordan 正则形式的方块。如果对相应的 S 矩阵最后一个零行消去，则消完一遍之后，又会出现新的零行，一直到给出

$$Y_1=p^4Y_0$$

为止（这只算是 p^3 依赖关系，因为一个 pY_0 原来就在给定函数之中）。

实际上无论处理行或列，每次消去后都把主元素所在的行调到下面去。模主元素时不再涉及这些行，但对一切行都要作消去变换。这样就在分离微分方程时，完全解出了代数方程。

完成行消去或列消去之后，都要对 S、R 矩阵的各列作一次重排，把 S 中的零元素列全部交换到最右面去。这样，S 的左上角就剩下一个 $L*L$ 的方阵 S_D，其中 $L=N-K$，而 K 是被消去的环节数目。线性方程组的矩阵记法中，行是可以随便交换的，而列的交换相当于环节重新编号。这个新旧编号的对应关系要记录下来，以便按原来的序号输出计算结果。为此引入一个矢量 IC_i（我们用两个大写字母 IC 代表一个矢量，以便与程序中的数组名字对应），换列之前令其第 i 个分量的数值就等于 i：

$$IC_i=i,\ (i=1,2,\cdots N)。$$

每次两列交换时，IC_i 的相应元素也交换一次。这样，IC_i 的序号 i 始终对应新的列号，而其数值本身则指明此列的原始编号。

列交换时各环节输出函数也作了一次换位：

$$\tilde{Y}=TY,$$

变换矩阵 T 的元素可用矢量 IC 的分量定义：

$$T_{ij}=\delta_{IC_i,j}\qquad(i,j=1,2,\cdots N)$$

这里 δ 就是前面引入的 δ_{ij}，只不过把角标 i 换成了 IC_i。T 显然是正交矩阵，于是

$$Y=T^{-1}\tilde{Y}=T^T\tilde{Y},$$

写成分量形式是

$$Y_i=\sum_j\delta_{IC_j,i}\tilde{Y}_j$$

实际上这就是 $Y_{IC_j}=\tilde{Y}_j$。数值结果就这样按原来环节编号印出。

程序中还引入了另一个矢量 ID 来保存 S 矩阵中全零列的列号。对于列消去，如果原来 S 中第 2、4、7 三列为零，则完成列消去的同时将形成 $ID_1=2$，$ID_2=4$，$ID_3=7$。对于行消去，行主元素的列号指明 S 中完成消去后出现的新零列，也要用 ID 保存。列交换根据保存在 ID 中的信息，按一定顺序进行，以保证 R 矩阵右下角成为单位矩阵。

完成消去和交换后的基本方程组可以写成

$$\begin{pmatrix}S_D & 0\\ S_A & 0\end{pmatrix}\begin{pmatrix}p\tilde{Y}_D\\ p\tilde{Y}_A\end{pmatrix}=\begin{pmatrix}R_D & 0\\ R_A & I\end{pmatrix}\begin{pmatrix}\tilde{Y}_D\\ \tilde{Y}_A\end{pmatrix}+$$
$$+\begin{pmatrix}V_{1D}+V_{2D}p+V_{3D}p^2\\ V_{1A}+V_{2A}p+V_{3A}p^2\end{pmatrix}Y_0$$

其中 S_D 和 R_D 是 $L*L$ 的方阵，角标 D 和 A 分别指明"微分"和"代数"部份。常微分方程组

$$S_D p\tilde{Y}_D = R_D\tilde{Y}_D + (V_{1D}+V_{2D}p+V_{3D}p^2)Y_0$$

中，如果 S_D 可以求逆，则可化为标准形式，用任何一种数值积分方法求解。如果 S_D 仍不能求逆(调节系统违背对给定函数高阶导数依赖性的限制时，就会出现这种情形)，则程序就不予处理，但印出有关信息，以供分析。

每次输出数值结果之前，将 $\tilde{Y}_D$ 代入其余的代数方程，求得。

$$\tilde{Y}_A = S_A(p\tilde{Y}_D) - R_A\tilde{Y}_D - (V_{1A}+V_{2A}p+ +V_{3A}p^2)Y_0,$$

再恢复成原来 Y 的顺序印出。

在 $L=0$ 的特殊情形下，$R=I$，整个系统根本不由微分方程描述，而是通过代数和微分关系由给定函数直接决定。程序中对此作了考虑。

六、总传递函数和系统的稳定性

分析自动调节系统时，往往要把框图归併，求出一定输出端对某个给定函数输入端的总传递函数。一切这类总传递函数都已包括在前面的推导之中。

回到消去了代数部份的基本方程组：

$$pY_D = WY_D + U(p)Y_0$$

其中引入了符号

$$W = S_D^{-1}R_D,$$

$$U(p) = S_D^{-1}(V_{1D}+V_{2D}p+V_{3D}p^2),$$

并省去了各个字母上的波纹线。明显解出 Y_D 来：

$$Y_D = K(p)Y_0,$$

这里

$$K(p) \equiv (Ip-W)^{-1}U(p)$$

就是总传递函数矩阵。它的每个元素代表从一个输入端到某个输出端的传递函数，具体算出来乃是对于变量 p 的一个有理分式。程序中没有去计算这些有理分式的系数，只是就矩阵 W 的特征值判断系统的稳定性。

不带给定函数的齐次微分方程组

$$(Ip-W)Y_D = 0$$

决定调节系统的自由振动。如果矩阵 W 的特征值是 $\lambda_1, \lambda_2, \cdots$等，则总可以找到非奇异的方阵 P 作相似变换 $P(Ip-W)P^{-1}$，使它成为Jordan标准形式：对角线上有一些方块，每个方块对应一个特征值。假定特征值 λ_i 是三重根，则相应方块是

$$\begin{pmatrix} p-\lambda_i & -1 & 0 \\ 0 & p-\lambda_i & -1 \\ 0 & 0 & p-\lambda_i \end{pmatrix}$$

如果对 Y_D 同时作线性变换 $\tilde{Y}_D = PY_D$，则上述方块对应 $\tilde{Y}_D$ 中的三个分量的通解形式是

$$\alpha e^{\lambda_i t} + \beta t e^{\lambda_i t} + \gamma t^2 e^{\lambda_i t},$$

$$\beta e^{\lambda_i t} + \gamma t e^{\lambda_i t},$$

$$\gamma e^{\lambda_i t},$$

其中 α、β、γ 是常数。系统稳定性要求一切自由振动模式随时间衰减，由此得到稳定性条件：矩阵 W 的一切特征值都必须位于左半复平面上：

$$Re(\lambda_i) < 0$$

实部最靠近 0 的那些特征值，虚部也不宜太大，系统的暂态过程中才不会出现过多的摆动。为了分析调节系统的行为，知道 W 的全部特征值是很有益的。程序中为此增加了求一般实矩阵特征值的部份，只有判断了系统的稳定性之后，才转而求积微分方程组。

七、程序结构

用 FORTRAN 语言写出的程序要求具体规定环节和给定函数数目的上限(我们取 $N \leqslant 30$，$M \leqslant 6$)，ALGOL 语言的程序对 N、M 没有限制(只要存贮区够用)，这是算法语言本身的特点决定的。下面以 FORTRAN 语言的方案为例，说明程序的总体结构。

整个程序由以下十三个模块组成：

(一) 实矩阵平衡子程序 BALANG，它同时分离出矩阵的孤立特征值。

(二) 用初等变换把实矩阵化成上 Hessenberg 形式的子程序 ELMHES。

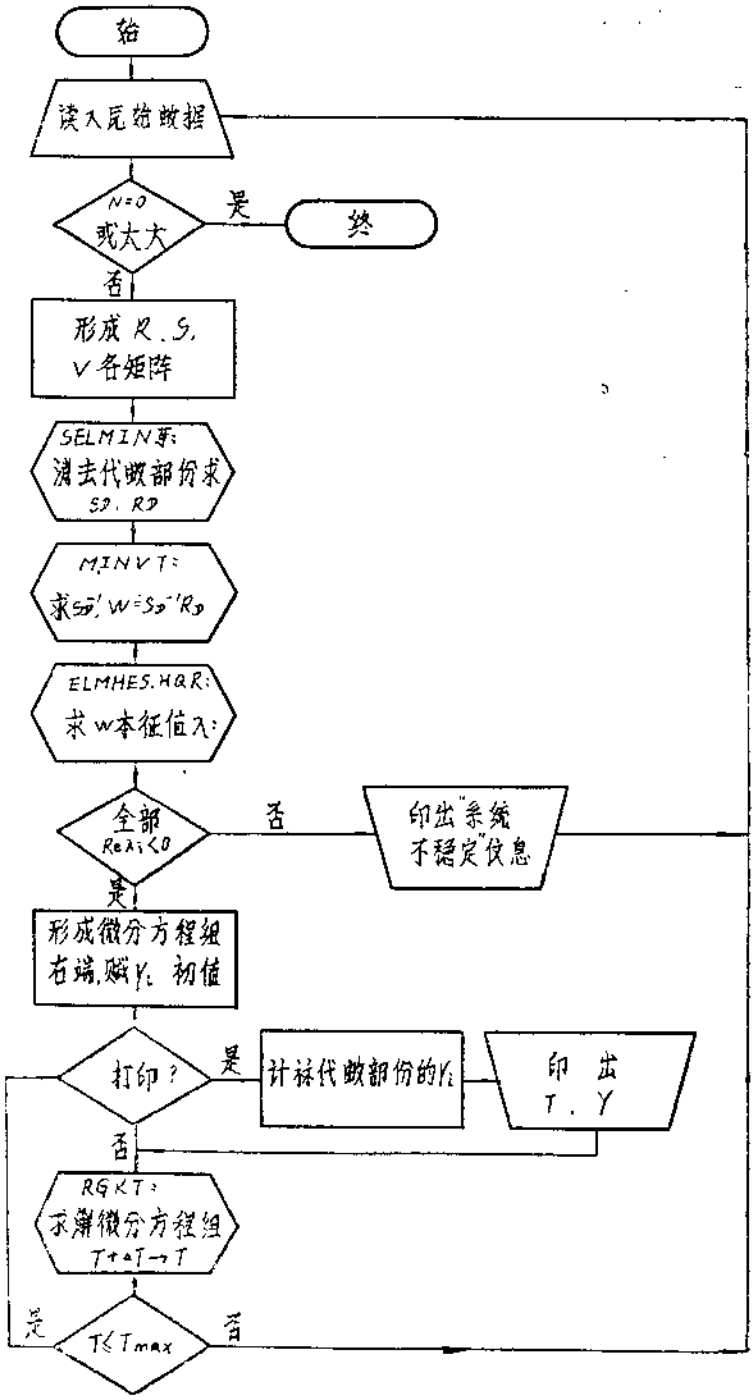

图 7　主程序粗框图

（三）求实 Hessenberg 型矩阵全部特征值的子程序 HQR。

以上三个子程序取自[6]。

（四）四阶龙格-库塔法求积常微分方程组的子程序 RGKT。

（五）两个实变量对换位置的子程序 INTRCH。

（六）实矩阵求逆子程序 MINVT。

以上六块都是一般的通用程序。下面五块是专用的。

（七）实现 S 矩阵中列主元素消去的无参子程序 SELMIN。

（八）针对 S 中的零元素列，按 R 作列主元素消去的子程序 COLUMN。

（九）针对 S 中的零元素行，按 R 作列主元素消去的子程序 ROW。

（十）在 S 和 R 中换列并在 IC 中记录新旧环节编号的子程序 EXCHG。

（十一）为龙格-库塔法子程序提供微分方程组右端函数的子程序 RIGHT。

（十二）由使用者编写的描述各个给定函数及其一、二阶导数的子程序 GIVEN，例如

```
SUBROUTINE GIVEN (T, Y00, Y01, Y02)
DIMENSION Y00(2), Y01(2), Y02(2)
Y00(1)=1.
Y01(1)=0.
Y02(1)=0.0
Y00(2)=.01*SIN(T)
Y01(2)=.01*COS(T)
Y02(2)=-.01*SIN(T)
RETURN
END
```

其形参 T 是时间，Y00 等是各有 M 个分量的一维数组，用以存放给定函数及其一、二阶导数的值。

模块数目虽多，但绝大多数子程序都不发生互相调用关系。因此，在具有覆盖处理功能的计算机上，除 GIVEN 和 INTRCH 两块外都可置入覆盖区，覆盖区的大小由其中最长的 HQR 决定。因此，本文介绍的程序虽然很大却不要求占用很多主存。

（十三）主程序。

图 7 是主程序的粗框图，其中对矩阵偶 S 和 R 作消去和换位的部份在图 8 中给得详细一些。

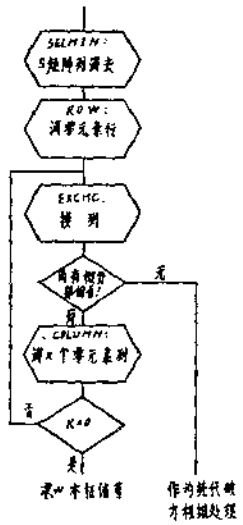

图 8　消去和换位部份细框图

八、计算实例和输出形式

我们以一个可控硅励磁的直流电动机速度

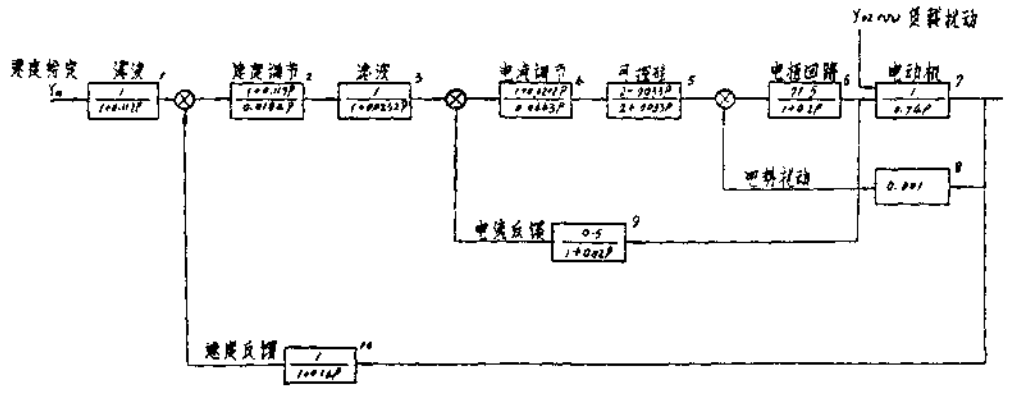

图 9　调节系统实例

调节系统为例，说明一下原始数据的顺序和计算结果的输出形式。调节系统框图如图9所示，其中参数值大致是根据资料[17]选取的。可控硅延迟环节采用了线性分式逼近（见前面第二节）。这个系统环节数 $N=10$，消去后剩下 $L=9$。输出结果的前若干行如下：

```
N=   10   TMAX=      0.1000   STEP   OF   T=       0.0010   KDY=       5
  PARAMETERS IN (C+D*P)/(A+B*P):
      I          A            B            C            D
      1       1.00000      0.117000     1.00000      0.000000
      2       0.000000     0.184000E -1 1.00000      0.117000
      3       1.00000      0.232000E -1 1.00000      0.000000
      4       0.000000     0.443000E -1 1.00000      0.212000E -1
      5       1.00000      0.330000E -2 1.00000      0.000000
      6       1.00000      0.200000     71.5000      0.000000
      7       0.000000     0.740000     1.00000      0.000000
      8       1.00000      0.000000     0.100000E-2  0.000000
      9       1.00000      0.200000E -2 0.500000     0.000000
     10       1.00000      0.140000E -1 1.00000      0.000000
  INCIDENCE MATRIX W:
    0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
    1.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  -1.0
    0.0   1.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
    0.0   0.0   1.0   0.0   0.0   0.0   0.0   0.0  -1.0   0.0
    0.0   0.0   0.0   1.0   0.0   0.0   0.0   0.0   0.0   0.0
    0.0   0.0   0.0   0.0   1.0   0.0   0.0  -1.0   0.0   0.0
    0.0   0.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0   0.0
    0.0   0.0   0.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0
    0.0   0.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0   0.0
    0.0   0.0   0.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0
  INCIDENCE MATRIX U:
    1.0   0.0
    0.0   0.0
    0.0   0.0
    0.0   0.0
    0.0   0.0
    0.0   0.0
    0.0   1.0
    0.0   0.0
    0.0   0.0
    0.0   0.0
  ORDER OF THE SYSTEM=9
  CHARACTERISTIC ROOTS=
  1    -576.249        0.000000
  2    -133.667        0.000000
  3    -63.9243         76.1714
  4    -63.9243        -76.1714
  5    -8.43277         18.4679
  6    -8.43277        -18.4679
  7    -17.4430        0.000000
  8    -50.4898        0.000000
  9    -8.54701        0.000000
    T=  0.005000
        0.418348E -1   0.271738     0.272872E -1   0.138887E -1   0.501947E -2
        0.238715E -2   0.354291E-5  0.354291E -8   0.429919E -3   0.214709E -6
    T=  0.010000
        0.819194E -1   0.543413     0.101964       0.524610E -1   0.293503E -1
        0.295503E -1   0.878093E-4  0.878093E -7   0.818240E -2   0.100906E -4
```

原始数据单如下：

```
1Ø,2 (环节数，给定函数个数)
.1,.ØØ1,5 (时间上限，时间步长，打印间隔)
1,Ø,117, 1,Ø
Ø,Ø.Ø 184,1,Ø.117
1,Ø.Ø 232,1,Ø
Ø,Ø.Ø 443,1,Ø.Ø 212
2,.ØØ 33,2,.ØØ 33
1,Ø.2,71.5,Ø.
Ø,Ø.74,1,Ø
1,Ø,Ø.ØØ 1,Ø
1,Ø.ØØ 2,Ø.5,Ø
1,Ø.Ø 14,1,Ø
```

10 个环节的参数

```
12,2(W 和 U 中的非零元素个数)
2,1,1
2,1Ø,—1
3,2,1
4,3,1
4,9,—1
5,4,1
6,5,1
6,8,—1
7,6,1
8,7,1
9,6,1
1Ø,7,1
```

W 的非零元素下标及数值

```
1,1,1
7,2,1
```

U 的非零元素下标及数值

```
Ø,Ø
Ø,Ø,Ø 结束标志
```

九、讨　论

本文介绍的处理高阶线性自动调节系统的算法和程序，还可以作许多变化和改进。这里仅指出以下几点。

（一）这个程序可以改为对话式，以便在大部份参数不变情形下，迅速整定少量参数。为此要使用一批数组来保存原始数据，增写通过人机对话改变参数的一段。由于对话式读写不是算法语言的标准部份，我们没有提供这样的程序。

（二）本程序提供的各个模块，可以容易地用来组成其它分析自动调节系统用的程序。例如，只要再写一个主程序，不必增加新的模块，就可以构造一个程序，计算和打印调节系统特征方程的根随某些环节参数变化情形。

（三）由于这个程序始终是在时间域 t 中计算，并未转入拉氏变换后的 p 表示或频率域，故非定常问题（即某些环节参数与时间有关），或拟线性问题（即环节参数依赖于 Y_j 的当前值），是不难纳入本文讨论的框架的，但主程序必须改写。

（四）包含非线性环节的自动调节系统很难象线性系统这样作一般讨论。但诸如限幅、开关这类非线性环节，可以借助在时间域中计算这一方便条件，设法加以处理。

（五）本程序可以同时计算几个互相无关的调节系统，只要给全部环节和给定函数统一编号，并在连接矩阵中反映它们各自的连接情况。当然，这样作在存贮容量和计算速度两方面都是不经济的。

（六）最后，我们期望计算数学工作者能早日发展正则矩阵偶化标准形式，同时给出变换矩阵 P 和 Q 的实用程序，以便使本文遇到的这类问题得到更一般的解决。

参 考 资 料

[1] 关肇直，《现代控制理论中的几个问题》，《计算机应用与应用数学》，1974，№2，34～48。

[2] 郝柏林，《FORTRAN程序设计讲义》，《电子计算机参考资料》，1977，№1～2，213～227。

[3] 同上第10章§10.4.3。

[4] Ф. Р. Гантмахер. Теория Матриц. 第十二章§7，1966.（有1954年版汉译本，柯召译，《矩阵论》，高教出版社，1955）。

[5] 许以超，《代数学引论》，第十五章，上海科学技术出版社，1965。

[6] B. T. Smith et al. Matrix Eigensystem Routines EISPACK guide. 1974。

[7] 陈广洲，《最佳调节原理在工程上的应用》，《电气传动》，1974，№1，1～48。