控制系统化简的矩阵方法

陈怀琛，屈胜利，何雅静

西安电子科技大学, 西安 710071
E-mail: hchchen1934@vip.163.com, slqu@mail.xidian.edu.cn, heyajing2008@yahoo.com.cn

摘　要: 在复杂系统结构图需要化简时, 自动控制理论给出的传统方法是将它逐次化为框图的并联, 串联, 反馈…等等基本情况的组合. 此法不仅繁琐和易出错, 而且有时还要移动节点位置, 改变某些状态变量的意义. 本文提供的方法, 可直接对原结构图进行矩阵建模求解, 并藉计算机之助, 快速准确地解决这个问题.

关键词: 信号流图, 梅森公式, 矩阵建模, 线性系统, MATLAB, 系统结构图, 传递函数, 系统函数, 零极增益, 状态空间

Control System Simplification by Matrix Modeling

Chen Huaichen, Qu Shengli, He Yajing

Xidian University, Xi'an, 710071 P. R. China
E-mail:hchchen1934@vip.163.com, slqu@mail.xidian.edu.cn, heyajing2008@yahoo.com.cn

Abstract: Traditionally the reduction of structure diagrams of complex control systems was realized by partitioning it into three basic types: serial, parallel and feedback connections. This method is always difficult and fallible in practice. Very often one have to move the positions of nodes and thus change the meaning of state variables. In this paper, a general matrix model for linear systems is deduced. According to this method, the computing model directly comes from the original structure diagram. With the aid of computing software, this problem can be solved very quickly and precisely.

Keywords: signal flow graph, Mason's formula, matrices modeling, linear systems, MATLAB, signal processing, digital filters, network structure diagram, system structure diagram, transfer function, system function, zero-pole-gain model, state-space model

1 问题的提出(Introduction)

结构图的分析和化简是研究自动控制系统的第一步, 它也是信号流图的另一种形式. 对于比较复杂的信号流图, 最早提出通用解法的是梅森[1][2], 许多教材至今都以此方法为经典. 但梅森公式证明很困难,只能让学生死记硬背, 用起来也很易出错, 因为它用图形拓扑的方法分析,不容易编成程序, 无法实现机算, 所以实际上并未得到应用. 目前在所有的大学控制类教材中实际使用的多是针对简单系统导出的并联, 串联和反馈公式, 对于复杂的系统, 必须把它的方框按三种简单情况, 合二为一地逐次化简, 直到最后成为一个方框为止.

在科学计算软件已高度发展的今天, 有些软件(例如MATLAB)已有了专门针对控制系统的工具箱, 不过它对结构图的化简仍然依靠对这三种简单情况所导出的函数, 每次把两个方框合成一个. 这样逐次处理的方法, 其缺点一是繁琐, N个方框就要作N-1次运算; 二是难以处理多输入多输出(MIMO)系统; 三是并不是所有框图都能化为这三种情况的合成的, 为了凑成这种结果, 往往必须把框图中的某些综合点或分叉点移位, 结果将改变这些点及其邻近点变量的物理意义, 这是系统分析中的大忌. MATLAB中也有一组化简全系统特性的函数, 主要是connect函数[3], 但它的过程仍相当繁, 计算的原理对用户不透明, 特别是合成结果仅限于状态方程模型, 物理意义不清晰, 不受实际工程人员的欢迎.

对分析线性控制系统特性这么一个重要的核心问题, 我们需要给学生和工程技术人员一种概念清楚, 容易记忆, 能同时计入所有环节和全部输入输出, 快捷准确的方法. 显然, 这个方法必须要利用计算机对海量数据能作批处理的能力, 而不是每次只处理两个环节的传统做法, 因此对系统原始方程进行矩阵建模是一个最好的方法.

2 基本原理(Fundamental Principles)

系统结构框图中的每一个箭头代表一个信号, 每一个方框中的函数为信号经过它时所作的数学变换, 即输出信号等于输入信号乘以该函数, 综合节点的输出为输入信号的和, 分叉节点则把同样的信号分送到几个方框, 如果图中共选择了N个系统信号(状态变

此项工作得到教育部“利用信息技术改造课程项目”资助, 项目批准文号:教高厅函[2009]1号

量)x_j(j=1, 2, …, N)和M个输入信号u_i(i=1, 2, …, M), 则其中任何一个信号x_j都可以用包括自身在内的N个系统信号和M个输入信号来表示, 即:

$$x_j = f(x_1, x_2, \cdots, x_N, u_1, \cdots, u_M) \qquad j = 1, \cdots, N \qquad (1)$$

如果这些数学关系都是线性的, 则各方程将是线性方程, 它们构成的方程组可用矩阵表示, 写成:

$$x_j = q_{j1}x_1 + q_{j2}x_2 + \cdots + q_{jN}x_N + p_{j1}u_1 + \cdots + p_{jM}u_M \quad j = 1, \cdots, N$$

或

$$x_j = \sum_{k=1}^{N} q_{jk}x_k + \sum_{i=1}^{M} p_{ji}u_i \qquad j = 1, \cdots, N \qquad (2)$$

写成矩阵形式, 应为:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1N} \\ q_{21} & q_{22} & \cdots & q_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ q_{N1} & q_{N2} & \cdots & q_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1M} \\ p_{21} & p_{22} & \cdots & p_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NM} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_M \end{bmatrix} \qquad (3)$$

用矩阵符号表示为:

$$\mathbf{X} = \mathbf{QX} + \mathbf{PU} \qquad (4)$$

其中: $X = [x_1, x_2, \cdots, x_N]^T$ 为N维状态列向量, $U = [u_1, u_2, \cdots, u_M]^T$ 为M维输入列向量, Q为N×N维的联接矩阵, P为N×M维的输入矩阵, Q和P的元素q_{ji}和p_{jk}是各信号节点之间的传递系数. 移项后, 上式可写为:

$$\mathbf{(I - Q)X = PU}$$

如果矩阵**(I-Q)**可逆(凡是有解的系统都能满足此条件), 即可得到各输入u_i对各信号x_j的传递函数公式:

$$\mathbf{W = X/U = (I - Q)^{-1}P} \qquad (5)$$

W是N×M维矩阵, 即包含由M个输入引起N个状态之间的N×M个传递函数. 根据这个简明的公式, 只要写出P和Q, 任何复杂系统的传递函数都可用这个简单的式子求出. 这个矩阵方程并不难导出, 由于它的阶数等于系统中的变量数N, 手工求N阶矩阵的逆就叫人望而生畏. 所以各种求系统传递函数的方法无非都是想在求逆矩阵$(I\text{-}Q)^{-1}$方面找到某些规律, 但高阶矩阵求逆的普遍简明规律是难以找到的, 所以没有人做过这样的尝试.

现代科学计算工具的发展已改变了这种状况, 求几十阶矩阵的逆所用的时间可用秒计. 不仅可以计算数字矩阵, 也可以计算由字符构成的符号矩阵. 用上述公式几乎可以解决所有的结构图变换问题, 以MATLAB为例, 如果矩阵中有含有非数字的符号. 只要在程序的开始, 规定某些字符是符号元素(矩阵), 系统就不会去追究它的值, 而在运算中始终保留这个符号. 定义符号变量使用的命令是syms. 在对系数矩阵P和Q赋值时, 还有两点要说明:

⑴ 由于变量要分成数值变量和符号变量两类, 所以变量组成的矩阵也分成数值矩阵和符号矩阵两类. 符号变量矩阵可以有数值元素, 但数值矩阵就不容许有符号变量. 比如这里的Q矩阵, 必须是符号矩阵. 矩阵的属性是由第一个赋值元素的属性决定的. 所以程序中给Q赋值的第一条语句的右端, 必须含符号变量, 这很重要, 它保证了Q是符号矩阵.

⑵ Q和P阵都有很多个零元素, 可先不用去管它. 只要先给非零的元素赋值, 最后再把矩阵右下角的元素赋值, MATLAB会自动把矩阵中所有左上方的未赋值元素都赋为零值.

3 以实例说明解题编程方法(An Example for Solution and Programming)

以图1所示系统为例. 该系统有四个方框G_1, G_2, G_3, G_4, 两个输入信号u_1, u_2, 图中任意选定了四个状态变量x_1, x_2, x_3, x_4. 如果它是一个随动系统, 则G_3为1, x_1为测量误差, x_3为输出, 不含测量噪声u_2的真正误差为$e= x_3\text{-}u_1$. 现在题目要求出以u_2为输入, x_1, x_2, x_3, x_4, e为输出的传递函数. 系统虽然简单, 但却不可能直接化成三种简单情况的合成. 除非把第二个合成节点移到第一个合成节点之前, 但这将改变变量x_1的原始物理意义. 另外, 想要把真正误差e涵盖进框图中去, 就必须增加图中的线条, 使图变得更加复杂.

本文提出的方法是从原始方程组出发, 建立矩阵模型, 这就不受结构图画法的局限. 这样的方程组保持了各信号节点的原来物理意义, 在这一步中各环节的传递函数可以先用简单的符号来代表. 在本例中, 取x_1, x_2, x_3, x_4, e五个信号节点列写方程, 有:

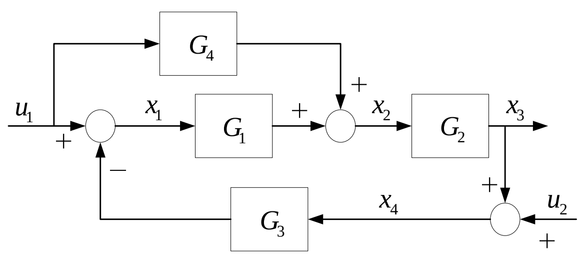

图1 系统结构图实例

$$\left.\begin{aligned} x_1 &= -G_3x_4 + u_1 \\ x_2 &= G_1x_1 + G_4u_1 \\ x_3 &= G_2x_2 \\ x_4 &= x_3 + u_2 \\ e &= x_3 - u_1 \end{aligned}\right\} \Rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ e \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -G_3 & 0 \\ G_1 & 0 & 0 & 0 & 0 \\ 0 & G_2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ e \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ G_4 & 0 \\ 0 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

写成矩阵形式:

$$\mathbf{X = QX + PU}$$

系数矩阵Q和P的内容一旦被赋值, 根据 $\mathbf{W = X/U = (I - Q)^{-1}P}$, 就可以求出以$u_1$, u_2为输入, x_1, x_2, x_3, x_4, e为五个输出的5×2传递函数矩阵, 其中包括十个传递函数. 如果我们还要求输出x_3的表示式, 则本题的MATLAB程序如下:

```
syms G1 G2 G3 G4 u1 u2
Q(1, 4)=-G3; Q(2, 1)=G1;                %给Q赋值
Q(3, 2)=G2; Q(4, 3)=1;Q(5, 3)=1;Q(5, 5)=0;
P(2, 1)=G4;P(1, 1)=1;P(4, 2)=1;         % 给P赋值
P(5, 1)=-1; P(5, 2)=0;
W=inv(eye(5)-Q)*P;     %  信号流图方程解
```

```
pretty(W)   % 美观显示输出误差项
X=W*[u1;u2];
pretty(X(3))      % 计算并美观显示输出
```

程序运行结果为:

$$W=\begin{bmatrix} \dfrac{1-G_2G_3G_4}{1+G_1G_2G_3} & \dfrac{-G_3}{1+G_1G_2G_3} \\ \dfrac{G_1+G_4}{1+G_1G_2G_3} & \dfrac{G_1G_3}{1+G_1G_2G_3} \\ \dfrac{G_1G_2+G_2G_4}{1+G_1G_2G_3} & \dfrac{-G_1G_2G_3}{1+G_1G_2G_3} \\ \dfrac{G_1G_2+G_2G_4}{1+G_1G_2G_3} & \dfrac{1}{1+G_1G_2G_3} \\ \dfrac{G_1G_2+G_2G_4}{1+G_1G_2G_3}-1 & \dfrac{-G_1G_2G_3}{1+G_1G_2G_3} \end{bmatrix}$$

它显示了呈矩阵排列的十个传递函数. 输出为:

$$X(3)=\left(\frac{G_2\left(G_1+G_4\right)}{1+G_1G_2G_3}\right)u_1-\frac{G_1G_2G_3}{1+G_1G_2G_3}u_2$$

用这样的推导也可以研究环节参数敏感度的问题, 只要把环节的传递函数给出一个增量, 例如令G3为G3+dG3就可研究dG3引起的输出.

4 本方法的推广(Extension of This Method)

为了使问题清晰, 前面只把各个环节特性用一个符号G_i来表示, 实际上本方法还可以对系统的详细特性进行推导. 用MATLAB中的两个工具箱都能支持本方法, 一是用符号数学工具箱中的推导方法, 二是用控制系统工具箱中的LTI模型. 例如本题中, 设

$G_1=\dfrac{1}{0.05s+1}$, $G_2=\dfrac{2}{(0.01s^2+0.1s+1)s}$, $G_3=1$,

$G_4=\dfrac{0.5s}{0.001s+1}$, 求以$u_1$为输入, x_1及x_3为输出的传递函数.

第一种方法是直接设G1, G2, G3为拉普拉斯算子s的函数, 符号变量由G1, G2, G3改为s, 把程序的前两句换成为下面5条语句, 得到:

```
clear,  syms s u1 u2
G1=1/(0. 05*s+1)
G2=2/(0. 01*s^2+0. 1*s+1)/s
G3=1;G4=0. 5*s/(0. 001*s+1)
Q(2, 1)=G1;Q(1, 4)=-G3;   %先赋值符号变量
…….. (下同)
```

运行此程序, 将得出一个比较冗长的结果, 因为Symbolic工具箱化简公式的功能还不能进行近似处理, 人工近似处理后的结果为:

$$X(3)=\frac{(4000+100(s+20)s)u_1-4000u_2}{s^4+30s^3+300s^2+2000s+4000}$$

第二种方法是把各部件的特性G用MATLAB控制系统工具箱中适当的LTI模型表示, 也能得出系统特性W的最后LTI模型. 我们知道, MATLAB的LTI模型主要有三种: 即传递函数, 零极增益或状态空间的形式. 诸环节采用哪种模型, 得出的系统传递函数也表示为相应的模型类型. 例如用零极增益模型, 便有:

```
G1=zpk([ ], -20, 20);
G2= zpk(tf(2, [0. 01, 0. 1, 1, 0]))
G3=1;G4=zpk(0, -1000,  500)
Q(2, 1)=G1;   Q(1, 4)=-G3;
Q(3, 2)=G2; Q(4, 3)=1;Q(5, 3)=1;Q(5, 5)=0;
P(2, 1)=G4;P(1, 1)=1;P(4, 2)=1; P(5, 1)=-1;
W=inv(eye(5)-Q)*P;  W(3, :)
```

因为不是用符号运算工具箱, 不需要设定符号变量的命令syms, pretty命令也无效, 程序运行结果为:

Zero/pole/gain from input 1 to output:

$$\frac{10000(s+17.79)(s+2.248)}{(s+19.25)(s+2.968)(s+1000)(s^2+7.779s+70)}$$

Zero/pole/gain from input 2 to output:

$$\frac{4000}{(s+19.25)(s+2.968)(s^2+7.779s+70)}$$

可见, 得到的系统传递函数已化为零极增益形式. 如果环节混合应用了三种LTI模型, 则MATLAB将遵循其运算优先规则确定最后的模型形式.

数字控制系统的化简也可以用这个方法, 只要把环节特性中的z变换的符号设成$q=z^{-1}$即可, 所以这个方法也可推广到数字信号处理, 用来计算数字滤波器的系统函数[5].

5 结语(Conclusions)

复杂线性系统可以方便地用矩阵模型来描述, 当使用软件工具来解矩阵方程时, 它的计算可变得非常简捷而精确. 这种方法理论严密清楚, 又具有极好的实用价值, 它同时适合于实数和复数矩阵, 也同时适合于数值计算和符号推理. 这种方法完全可以取代梅森公式, 而且证明严密简洁, 使用方便明快, 比它高明得多, 它充分利用了计算机软件和矩阵理论发展的新成果, 实现了对系统中的海量数据进行批处理, 一次求出MIMO系统的所有传递函数. 工程设计中常常需要做部件的参数调整, 并研究它对系统全局的影响, 本文提供的方法可以快速有效地解决这个问题.

传统的化简系统方法是依次合二为一, 使方程数(方框数)逐渐减少, 最后变成一个方程(方框), 就算实现了化简, 其实这个过程相当于初等的代入法或消去法, 它通常是极其冗繁的, 而且因为中间手工步骤多, 极易出错且很难查错. 矩阵解法和传统的的思路相反. 它从系统方程组的最原始形态出发, 不管它的阶数有多高,直接写成矩阵形式, 人工所做的工作只是简单的矩阵元素赋值, 然后用数学软件求逆, 一步得出最后结果. 只要元素赋值正确, 结果一定正确, 而赋值的正确性是很容易通过把矩阵Q, P的内容与方程系数直接比对来检验的.

从教育学的角度来看, 采用矩阵方法建模可以把信号与系统中的信号流图, 数字信号处理中的滤波器计算及自动控制课程中的系统化简都统一起来, 用同样的原理和程序解决这几类不同的问题, 它把复杂的高年级课程中的难题直接归结为线性联立方程组的求解, 成为低年级的线性代数命题. 只要学生在线性代数中学过用数学软件解题[6,7], 一看就会, 举重若轻. 既可节约课时, 又提高了教学质量.

参考文献(References)

[1] S. I. Mason. Feedback Theory I. Some Properties of Signal Flow Graphs [J] Proc. IRE, vol. 41(1953). pp1144-1156.

[2] S. I. Mason. Further Properties of Signal Flow Graphs[J]. Proc. IRE, vol. 44(1956). pp. 920-926.

[3] Control System Toolbox 8 User's Guide[M], The Mathworks, Sept. 2007

[4] 陈怀琛, 吴大正, 高西全.《MATLAB及在电子信息课程中的应用》[M]. 北京:电子工业出版社, 2002年1月第一版, 2006年7月第三版.

[5] 陈怀琛.《数字信号处理教程——MATLAB释义与实现》[M]. 北京:电子工业出版社, 2004年12月.

[6] 陈怀琛, 龚杰民.《线性代数实践及MATLAB入门》[M].北京:电子工业出版社, 2008年11月第二版.

[7] 陈怀琛, 高淑萍, 杨威. 《工程线性代数(MATLAB版)》[M]. 北京:电子工业出版社, 2007年7月.