

LBTR线性系统结构图法数字仿真

清华大学 顾启泰

摘要

本文提出了一种新的系统结构图数字仿真的方法。由于采用分子和分母都是二阶的典型环节,因而通用性强、使用更方便。文中推导了系统结构图和状态方程之间变换的数学表达式。所设计的计算矩阵指数的子程序保证了系统仿真精度不受计算步长的限制,从而大大提高了计算效率。本软件适用于多输入多输出系统,并备有多种类型的输入函数,还可以在任意环节加入初始条件,所以很适合于工程使用。

一、引言

近年来,由于数字计算机的飞速发展,利用数字机对实际系统进行数字仿真已日益为人们所采用。其方法是,首先将实际系统的动力学特性表示为数学模型;其次是将这个数学模型再变换成能在计算机上运算的仿真模型;最后编制出相应的程序,在计算机上对所设计的系统进行各种信号和扰动作用下的瞬态响应计算。采用数字计算机仿真的优点是,它可以对动力学特性截然不同的各种系统进行仿真研究这比作动态模拟试验的准备工作量要小得多,费用也少得多。

在早期的结构图法数字仿真中,选择积分环节 $\frac{1}{s}$ 作为典型环节^[1],当系统的阶次较高时,结构图的分解就显得十分复杂,并且需要一定的技巧。后来有人采用分子和分母都是一阶的环节 $\frac{(C+Ds)}{(A+B_s)}$ 作为典型环节^[4],简化了结构的变换,但在使用中仍

有一定的限制。考虑到在实际系统中一般环节都不超过二阶,所以这里我们选择分子和

分母都是二阶的环节 $\frac{Ks^2+\delta s+\gamma}{s^2+\beta s+\alpha}$ 作为典

型环节,这样就可以直接使用原有系统的结构图,不需要进行结构图变换,因而通用性强,使用更方便。各典型环节的参数及其联结方式分别由参数矩阵和联结矩阵表示。由于输入数据就是各个环节的参数,因而要研究某个参数变化对系统的影响将十分方便。参考输入或扰动输入可以加在任意一个或几个环节上。同样,系统输出也可以在任意一个或几个环节上同时取出,所以本软件适用于多输入多输出系统。

文中推导了系统结构图和状态方程之间变换的数学表达式。关于数值计算方法,以往较多采用四阶龙格-库塔法。为了保证计算的稳定性和精度,需要采用很小的计算步长,因而计算效率较低。这里我们采用连续系统离散化数值计算的方法,并且设计了计算指数矩阵的子程序,保证系统的仿真精度不受计算步长的限制,从而可使计算步长

1986年5月20日收到

和打印间隔一致起来，大大提高了计算效率。

二、典型环节

我们选择下述结构 (图 1)

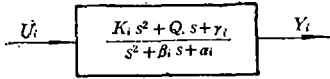


图 1 典型环节

作为典型环节，其中 i 表示序号。它包含 5 个参数： α_i 、 β_i 、 γ_i 、 δ_i 、 K_i ，只要适当选择其中某些参数为零，就很容易得到控制系统中最常见的一些环节，如比例、积分、惯性、一阶领先或滞后以及二阶振荡环节等。

用状态空间来描述上述典型环节，得

$$\begin{bmatrix} \dot{x}_{i1} \\ \dot{x}_{i2} \end{bmatrix} = \begin{bmatrix} -\beta_i & -\alpha_i \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_i$$

$$y_i = [K_{i1} \quad K_{i2}] \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} + K_{i3} u_i \quad (1)$$

简作

$$\dot{x}_i = A_i x_i + B_i u_i \quad (2)$$

$$y_i = C_i x_i + D_i u_i$$

式中， $K_{i1} = \delta_i - K_{i3} \beta_i$ ， $K_{i2} = \gamma_i - K_{i3} \alpha_i$ ， $K_{i2} = K_i$ 。

注意到上述典型环节，当 $\alpha_i = \gamma_i = 0$ 时， $K_{i2} = 0$ ，

$$\frac{Y_i}{U_i} = \frac{K_i s + \delta_i}{s + \beta_i}$$

为一阶环节，这时状态空间也应收缩为 1 维空间。表 1 列出了上述二阶和一阶环节所对应的系统矩阵。在程序设计中我们将分别加以考虑。

如果我们将图 1 中的拉普拉斯算子 s 换作微分算子 D ，即可在此环节上引入初始条件 $y_i(0)$ 和 $\dot{y}_i(0)$ 。与其对应的初始状态，当 $\delta_i = K_i = 0$ 时，

表 1

系统矩阵 环节	A_i	B_i	C_i	D_i
二阶	$\begin{bmatrix} -\beta_i & -\alpha_i \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$[K_{i1} \quad K_{i2}]$	$[K_{i3}]$
一阶	$[-\beta_i]$	$[1]$	$[K_{i1}]$	$[K_{i3}]$

$$x_{i2}(0) = \frac{y_i(0)}{\gamma_i}, \quad x_{i1}(0) = -\frac{\dot{y}_i(0)}{\gamma_i}$$

而当 δ_i 、 $K_i \neq 0$ 时，我们可将它拆成三个环节的并联，然后再在其中一个加入初始条件，同样可以进行初始条件下的系统仿真。

三、系统结构图

下面我们来确定由图 1 所示典型环节构成的系统的状态空间表达式。假定有一个如图 2 所示的系统结构图，其中每个有数字标号的方块都由上述典型环节组成。

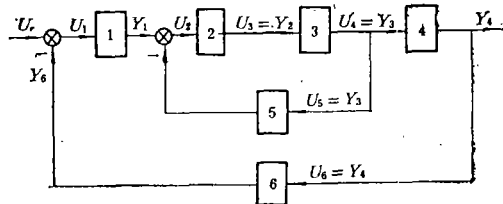


图 2 被仿真系统结构方块图

将各个环节的参数按次序排列成一个矩阵，称作参数矩阵，即

$$G = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & \delta_1 & K_1 \\ \alpha_2 & \beta_2 & \gamma_2 & \delta_2 & K_2 \\ \dots & \dots & \dots & \dots & \dots \\ \alpha_l & \beta_l & \gamma_l & \delta_l & K_l \end{bmatrix}$$

其中 l 为环节的数目，这里 $l = 6$ 。

各个环节之间的联结方式可用一个联结矩阵 Z 来表示，

$$Z = (Z_{ij})_{l \times l}$$

其中每一个元素 Z_{ij} ， $i = 1, 2, \dots, l$ ； $j = 1, 2, \dots, l$ 表示各个环节之间

的联结关系。当 $Z_{ij} = K$ 时，则表示第 j 个环节的输出乘上 K 之后再输入到第 i 个环节中去。所以，当 $Z_{ij} = 1$ 时，表示直接联结；当 $Z_{ij} = 0$ 时，表示没有联结；负号则表示负输入。例如对于图 2 所示系统

$$Z = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

参考输入或扰动输入可在任意一个或几个环节上同时加入，并且输入函数可以是相同的，也可以是不同的类型，这完全由键盘键入 I_1 、 $NI(I_1)$ 和 $U\$ (I_1)$ 来实现。其中， I_1 为输入维数， $NI(I_1)$ 为输入变量的序号， $U\$ (I_1)$ 表示输入函数的类型。并且规定

$U\$$	输入函数类型
A	阶跃函数
B	斜坡函数
C	脉冲函数
D	指数函数
E	正弦函数
F	均匀分布白噪声
G	正态分布白噪声

对于每一种函数，已事先编好了它的仿真子程序。调用时，机器将根据不同的输入函数类型询问其特征参数，如阶跃函数的大小，斜坡函数的斜率，脉冲函数的强度，指数函数的系数和时间常数，正弦函数的振幅、频率和相角，均匀分布白噪声的最大、最小值，正态分布白噪声的均值和均方根偏差等。

四、系统方程

由参数矩阵 G 经过 (1) 式组成以下四个对角阵

$$\begin{aligned} A &= \text{diag}(A_1 \ A_2 \cdots A_l) \\ B &= \text{diag}(B_1 \ B_2 \cdots B_l) \\ C &= \text{diag}(C_1 \ C_2 \cdots C_l) \\ D &= \text{diag}(D_1 \ D_2 \cdots D_l) \end{aligned}$$

这时，系统的状态空间表达式为

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (3)$$

式中， $x = \text{col}(x_1 \ x_2 \cdots x_l)$ ， $y = \text{col}(y_1 \ y_2 \cdots y_l)$ 。输入向量 u 应包括参考输入或扰动输入和环节交联输入两部分，即

$$u = u_0 + Zy \quad (4)$$

将 (4) 式代入 (3) 式，经整理最后得

$$\begin{cases} \dot{x} = \bar{A}x + \bar{B}u_0 \\ y = \bar{C}x + \bar{D}u_0 \end{cases} \quad (5)$$

式中

$$\begin{aligned} \bar{A} &= A + BZ(I - DZ)^{-1}C \\ \bar{B} &= B[Z(I - DZ)^{-1}D + I] \\ \bar{C} &= (I - DZ)^{-1}C \\ \bar{D} &= (I - DZ)^{-1}D \end{aligned}$$

再将 (5) 式所表示的连续方程离散化^[3]，得

$$\begin{cases} x[(k+1)T] = \phi(T)x(kT) + \bar{H}(T)u_0(kT) \\ y(kT) = \bar{C}x(kT) + \bar{D}u_0(kT) \end{cases} \quad (6)$$

$k = 0, 1, 2, \dots$

式中， $\phi(T) = e^{\bar{A}T}$ ， $\bar{H}(T) = Q\bar{B}$ ， $Q = \int_0^T e^{\bar{A}t} dt$ ， T 为采样周期，在这里就是计算步长。矩阵 ϕ 和 \bar{H} 与采样周期 T 有关，一旦 T 确定之后， ϕ 和 \bar{H} 都为常阵。

系统输出可以在任意一个或几个环节上同时取出，这由键盘键入 I_2 和 $NO(I_2)$ 来实现。其中， I_2 为输出维数， $NO(I_2)$ 为输出变量的序号。

五、程序设计

在上述矩阵 \bar{A} 、 B 、 \bar{C} 、 D 的运算中，都需要计算 $(I - DZ)^{-1}$ 阵。为此，令 $H = I - DZ$ ，并将 H 首先扩展成 $[H : I]$ ，然后利用初等行变换，将其变换成 $[I : F]$ ，则 $F = H^{-1}$ ，其 Invmat 子程序框图如图 3 所示。

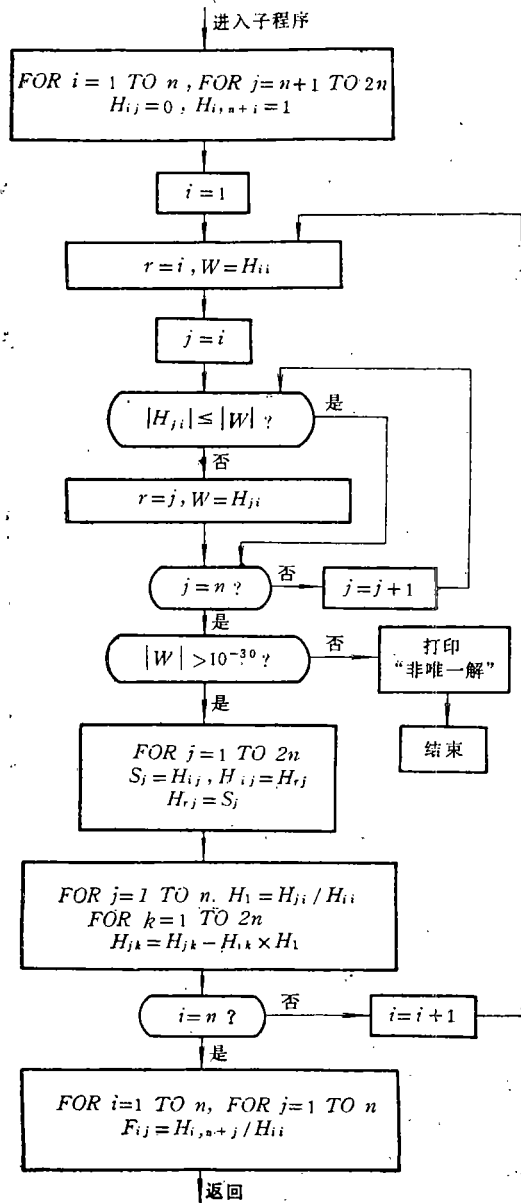


图 3 Invmat子程序框图

为了求解 (6) 式所表示的离散方程，需要计算 $\phi(T)$ 和 $\bar{R}(T)$ ，下面介绍本程序设计中采用的一种算法。

和计算纯量指数相似，也可将矩阵指数展开成幂级数的形式

$$e^{\bar{A}T} = \sum_{k=0}^{\infty} \frac{\bar{A}^k T^k}{k!} \quad (7)$$

令 $e^{\bar{A}T} = \phi(T) + R(T)$

其中

$$\phi(T) = \sum_{k=1}^{k_1} \frac{\bar{A}^k T^k}{k!}$$

为 $e^{\bar{A}T}$ 的近似解，

$$R(T) \approx \frac{\bar{A}^{(k_1+1)} T^{(k_1+1)}}{(k_1+1)!}$$

为截断误差。与此相似

$$Q(T) = T[Q'(T) + R'(T)]$$

其中

$$Q'(T) = \sum_{k=0}^{k_1} \frac{\bar{A}^k T^k}{(k+1)!}$$

$$R'(T) \approx \frac{\bar{A}^{(k_1+1)} T^{(k_1+1)}}{(k_1+2)!}$$

k_1 为迭代次数，它取决于所要求的计算精度。假定要求矩阵 $Q'(T)$ 中的每一个元素都精确到小数点 ρ 位，即

$$\frac{|r'_{ij}|}{|Q'_{ij}|} \leq 10^{-\rho} \quad (8)$$

其中 Q'_{ij} 、 r'_{ij} 分别表示矩阵 Q' 和 R' 的元素。这样每次迭代后都用 (8) 式检验，一直到满足不等式时为止。为了防止计算时间过长，对迭代次数仍应规定一个上限 K_{max} ，并且当 $K = K_{max}$ 时，由机器给出最大的 $|r'_{ij}|/|Q'_{ij}|$ 值。计算 $\phi(T)$ 、 $Q(T)$ 的 Expmat 子程序框图如图 4 所示。

LBTR 主程序框图如图 5 所示，其中部分变量说明如下：

- l 环节数目 Q_1 计算步长
 - l_1 输入维数 Q_2 仿真长度
 - l_2 输出维数 Q_3 打印间隔数
 - G 参数矩阵 $NI(l_1)$ 、 $US(l_1)$
- 输入序号和类型

Z 联结矩阵 $NO(l_2)$ 输出序号值得指出, 当输入为阶跃函数时, 由 (6) 式所表示的系统方程是准确的表达式, 其计算精度同计算步长大小无关。但实际上 e^{AT} 的计算不是准确的, 它取决于 Expmat 子程序的计算精度。很显然, 当计算

步长 T 小时, 所需 k_1 小, 而当 T 大时 k_1 也增大, 但最终的计算精度是相同的。所以在这意义上说, 系统仿真精度不受计算步长的限制。关于计算步长大小的选取, 可根据开环系统的剪切频率 ω_c 来定, 通常选择 $Q_1 = 1/10\omega_c$ 。

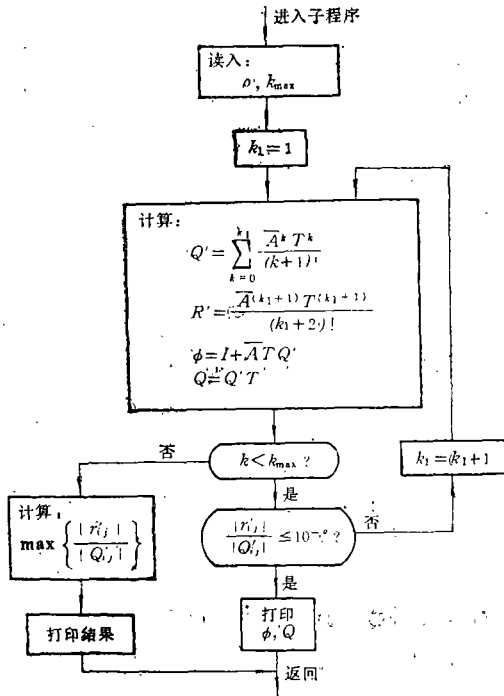


图 4 Expmat 子程序框图

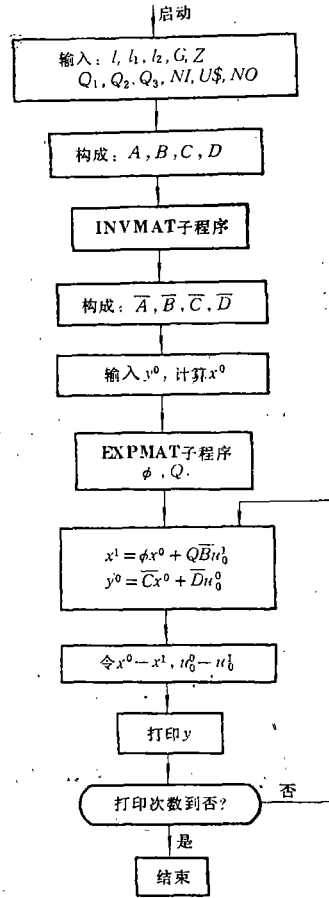


图 5 LBTR主程序框图

六、举 例

图 6 所示飞机俯仰速率控制系统方块图, 输入 $u_1 = 1(t)$, 输出 y_3 。现利用 LBTR 程序进行数字仿真。

数据输入:

$$l = 4, l_1 = 1, l_2 = 1, Q_1 = .1, Q_2 = 6, Q_3 = 2, NI(1) = 1, U\$ (1) = A, C = 1, NO(1) = 3$$

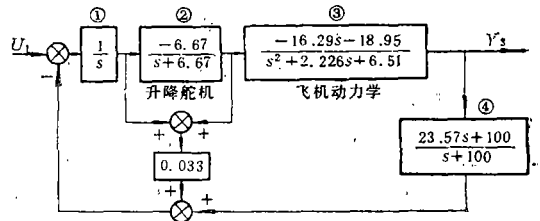


图 6 飞机俯仰速率控制系统

$$G = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 6.67 & 0 \\ 6.51 & 2.226 & -18.95 \\ 0 & 100 & 0 \\ & & 1 & 0 \\ & & -6.67 & 0 \\ & & -16.29 & 0 \\ & & 100 & 23.57 \end{pmatrix}$$

$$Z = \begin{pmatrix} -.033 & -.033 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

在Expmat子程序中选择 $\rho=5, k_{max}=50$ 。
 y_3 的仿真曲线如图 7 所示，其仿真精度达到
 1%~0.1%，一次仿真所需时间约1分钟。

参 考 文 献

[1] Meerinan, J. W., Interactive Simulation language THTSIM, Department of Electrical Engineering, Twente University of Technology, Enschede, The Netherlands, (1980).

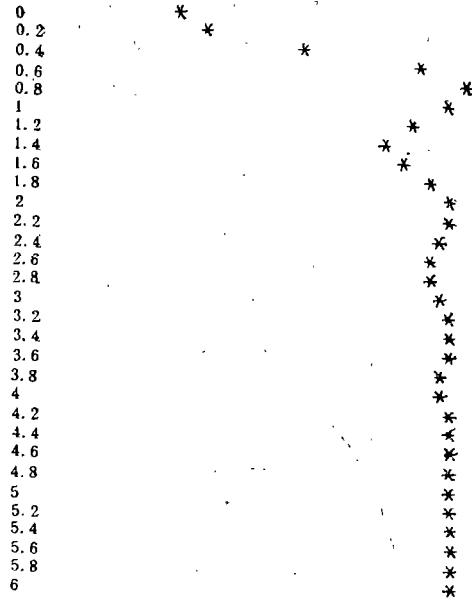


图 7 仿真曲线

[2] Kochenburger, K. J., Computer Simulation of Dynamic Systems, Prentice-Hall, (1972).
 [3] Ogata, K., Modern Control Engineering, Prentice-Hall, (1970).
 [4] 熊光楞, 控制系统数字仿真, 清华大学出版社, (1982)。

LBTR Block Diagram Approach to the Digital Simulation of Linear Systems

Gu Qitai

(Tsinghua University)

Abstract

A new linear block diagram approach is presented for digital simulation of linear systems. Owing to use a typical block in which numerator and denominator are both second order, it becomes more general and more usable. The mathematical expression of transforming linear block diagram into state equation is derived. The procedure for calculating the exponential matrix guarantees that the simulation accuracy is not restricted by the calculating step length, then the calculating efficiency has been raised greatly. Many types of input function are provided. The initial conditions can be added to any block. The Approach is very suitable for engineering applications.