系统分析与仿真

——MATLAB语言及应用

黄文梅　杨　勇　熊桂林　编著

国防科技大学出版社

系统分析与仿真

——MATLAB语言及应用

黄文梅　杨　勇　熊桂林　编著

国防科技大学出版社
· 长沙 ·

图书在版编目（CIP）数据

系统分析与仿真：MATLAB 语言及应用/黄文梅等编著．—长沙：国防科技大学出版社，1999.8

ISBN 7-81024-573-2

Ⅰ.系…　Ⅱ.黄…　Ⅲ.①电子计算机-系统分析②系统仿真③命令语言　Ⅳ.TP391.9

中国版本图书馆 CIP 数据核字（1999）第 34712 号

国防[redacted]社出版发行

电话:(0731)4555681　邮政编码:410073

E-mail: gfkdcbs@public.cs.hn.cn

责任编辑:何　晋　责任校对:文　慧

新华书店总店北京发行所经销

湖南大学印刷厂印装

*

787×1092　1/16　印张:14.75　字数:350 千

1999 年 8 月第 1 版第 1 次印刷　印数:1-4 000 册

*

定价：17.00 元

内 容 简 介

本书紧密结合控制系统仿真、分析和设计的基本知识，详细介绍了 MATLAB 语言的应用和编程技术。主要内容有：系统建模、时间响应及动态仿真、系统频率特性、连续和离散系统辅助设计、系统辨识、智能控制系统等。

本书将基本理论和 MATLAB 语言有机结合起来，并有丰富的应用实例和习题，可以帮助学生在学习基础知识和掌握计算机工程应用能力两方面都起到事半功倍的效果。

本书可作为自动控制、机电工程、机械设计与自动化、通信工程、电力等专业本科生、研究生有关课程的教材或参考书，也可作为有关工程技术人员、研究人员学习和运用 MATLAB 语言的自学用书。

前　　言

MATLAB 是一种面向科学和工程计算的高级语言，现已成为国际公认的最优秀的科技界应用软件，在世界范围内广为流行和使用。该软件的特点是：强大的计算功能、计算结果和编程可视化及极高的编程效率。这是其他语言无与伦比之处。MATLAB 包含的几十个工具箱，覆盖了通信，自动控制、信号处理、图像处理、财经、化工、生命科学等科学技术领域，汲取了当今世界这些领域最新研究成果，使之迅速成为从事科学研究和工程设计不可缺少的工具软件。今天，在欧美高等院校里，MATLAB 已成为大学生、硕士生、博士生、教师的必备的基本技能，广泛应用于科学研究、工程计算、教学、撰写论文等。国内不少高校，也正在推广应用 MATLAB 软件，用户愈来愈多。

本书专为自动控制、机械电子、机械设计、电力电气、通信工程等专业大学本科生、研究生、教师、科技工作者编写的教材。本书力图将自动控制原理、控制工程基础、计算机仿真、计算机辅助分析和设计、信号分析处理等有关教学内容和 MATLAB 语言紧密、有机地结合起来，使学生在学习基础理论知识的同时学会应用 MATLAB；在学习应用 MATLAB 的同时，加深对基础知识的理解，增强学生的计算机应用能力，提高教学效果。本书编入了智能控制内容，目的是使读者了解学科发展的最新动态。本书编入了大量的 MATLAB 应用程序，并配备了一定数量的习题，供读者能尽快掌握 MATLAB 编程技巧。

本书采用 MATLAB 5 的最新版本编写，和 MATLAB 以前的版本相比，MATLAB 5 包含功能更强大的编程工具，内容更加丰富的工具箱，编程语言更简捷、更友好的图形界面和可视化编程。这些特点使本书在内容上是完全崭新的，为了方便曾使用过 MATLAB 老版本的读者，第一章保留了 MATLAB 4.2 的部分内容。

本书和同时出版的《信号分析与处理——MATLAB 语言及应用》是一套丛书，内容相互联系，涉及自动控制和信号处理领域内主要研究课题和最新研究动向。

本书共分七章和附录。第一、五章和部分习题由杨勇编写，第三、七章由熊桂林编写，第二、四、六章由黄文梅编写，唐亚利编写了附录。全书由黄文梅汇总整理，湖南大学林丞教授主审。

由于时间仓促，不当之处在所难免，恳请读者不吝指教。

编　者

1999 年 6 月

目　　录

第一章　系统模型及转换

控制系统的数学模型对于控制系统的研究具有重要的意义。要对系统进行仿真研究，首先应建立系统的数学模型，在此模型的基础上建立系统的仿真模型，然后进行仿真研究，并可设计出相应的控制器对系统进行控制，使系统响应达到预期目标。

在线性系统中，常用的数学模型有微分方程模型、传递函数模型、状态方程模型以及零极点模型等。在一些场合下需用到其中一种模型，而在其它场合时可能又需要其它模型，所以掌握模型间的转换显得很有必要。本章将主要介绍系统类型、系统数学模型及转换、系统环节模型的连接、模型降阶及标准型实现等内容。此外，在所研究的系统模型未知时，可通过实测系统响应所获得的数据，采用系统辨识的方法来建立系统模型，这将在本书第六章中加以介绍。

1.1　系统分类

为研究、分析和综合问题方便起见，可对系统进行不同角度的分类。

1. 按有无反馈分类

(1) 开环系统——系统的输出端和输入端之间无反馈回路，称此系统为开环系统。

(2) 闭环系统——系统的输出端和输入端之间有反馈回路，称此系统为闭环系统。

2. 按系统中参数变化是否连续分类

(1) 连续系统——若一个系统中状态随时间连续变化，就称之为连续系统。

(2) 离散系统——系统的状态变化只在离散时刻点上发生，这种系统称之为离散系统。

3. 按系统是否为线性分类

(1) 线性系统——系统微分方程的系数是常数或者仅仅是自变量的函数。

(2) 非线性系统——系统只能用非线性微分方程来表示。

4. 按系统变量是否随时间变化分类

(1) 时变系统——系统参数不随时间变化的系统。

(2) 非时变系统——系统参数随时间变化的系统。

5. 按系统参数分布规律来分类

(1) 集中参数系统

(2) 分布参数系统

6. 按系统是否确定分类

(1) 确定型系统

(2) 随机型系统

1.2　系统数学模型及转换

1.2.1　系统的时域模型

对于连续时间系统用微分方程描述，系统数学模型的一般形式为

$$a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \ldots + a_0 y(t) = b_m u^{(m)}(t) + b_{m-1} u^{(m-1)}(t) + \ldots + b_0 u(t) \qquad (1.2\text{-}1)$$

其中，y 和 u 分别为系统的输出与输入量，$y^{(i)}$及 a_i分别表示输出量各阶导数及输出量的系数，$u^{(i)}$及 b_i分别表示输入量各阶导数及输入量的系数。

对于离散时间系统，可用差分方程描述系统模型，其一般形式为：

$$\begin{aligned} & g_n y[(k+n)T] + g_{n-1} y[(k+n-1)T] + \ldots + g_1 y[(k+1)T] + g_0 y(kT) \\ & = f_m u[(k+m)T] + f_{m-1} u[(k+m-1)T] + \ldots + f_1 u[(k+1)T] + f_0 u(kT) \end{aligned} \qquad (1.2\text{-}2)$$

1.2.2 系统的传递函数模型

传递函数是经典控制论描述系统的数学模型的一种方法，它表达了系统输入量和输出量之间的关系。它只和系统本身的结构、特性和参数有关，而与输入量的变化无关。传递函数是研究线性系统动态响应和性能的重要方法。

例如，假设有一个单输入单输出连续系统的输入信号为 u(t)，且输出信号为 y(t)，系统相应的微分方程如式(1.2−1)所示。对此微分方程作 Laplace 变换，则该连续系统的传递函数为

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_0}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_0} \qquad (1.2\text{-}3)$$

在 MATLAB 4.2 中，该系统可由其分子和分母多项式的系数（按 s 的降幂排列）所构成的两个向量唯一地确定下来，即

$NUM = [b_m, b_{m-1}, \cdots, b_0]$

$DEN = [a_n, a_{n-1}, \cdots, a_0]$

在 MATLAB 5 中，用函数 TF 可以建立一个系统传递函数模型，其调用格式为

SYS = tf (NUM, DEN)

【例 1.1】 若给定系统的传递函数为

$$G(s) = \frac{18(s+2)}{(s+15)(s+25)(s+0.4)}$$

则可以将其用 MATLAB 形式表达出来：

```
% MATLAB PROGRAM 1-1
num=18*[ 0, 1, 2 ]; den = conv (conv ([1 15],[1 25]),[1 0.4]);
printsys(num,den,'s')
```

运行结果:

```
num/den =
          18S + 36
    ----------------------
    S^3 + 40.4S^2 + 391S + 150
```

此例也可用 MATLAB 5 建立传递函数模型如下:

```
% MATLAB PROGRAM 1-2
num=18*[0, 1, 2];     den=conv(conv([1 15],[1 25]),[1 0.4]);
mod=tf (num, den)
```

运行结果：

Transfer function:

```
        18S + 36
---------------------------
S^3 + 40.4S^2 + 391S + 150
```

对于离散时间系统，对(1.2-2)进行 Z 变换，则可得到该离散系统的脉冲传递函数

$$G(z)=\frac{Y(z)}{U(z)}=\frac{f_m z^m+f_{m-1}z^{m-1}+\cdots+f_0}{g_n z^n+g_{n-1}z^{n-1}+\cdots+g_0} \tag{1.2-4}$$

其中，对线性时不变离散系统来讲，式(1.2-4)中 f_i 与 g_i 均为常数，且 $g_n\neq 0$。

在 MATLAB 4.2 中，该离散系统也可以由其分子和分母多项式的系数(按 z 的降幂排列)所构成的两个向量来唯一确定，即

$$NUM=[f_m, f_{m-1},\cdots, f_0]$$
$$DEN=[g_n, g_{n-1},\cdots, g_0]$$

在 MATLAB 5 中，可用函数 TF 来建立系统传递函数模型，调用格式为

```
SYS = tf [NUM, DEN, T]
```

其中，NUM 和 DEN 的意义同上，T 为采样时间间隔。

如果已知二阶系统的两个参数——自然频率ω_n和阻尼比ζ，则可通过函数 ORD2 求出此系统传递函数模型和状态方程模型，其调用格式分别为

$$[A, B, C, D] = \mathrm{ord2}\,(\omega_n, \zeta)$$

调用结果返回连续二阶系统状态方程的系数矩阵 A, B, C, D 的表达式；

$$[NUM, DEN] = \mathrm{ord2}\,(\omega_n, \zeta)$$

调用结果返回连续二阶系统用分子和分母多项式表示的传递函数。

传递函数表达式用于单输入单输出(SISO)系统建模非常方便，也可用它来表达多输入多输出(MIMO)系统，MATLAB 提供用传递函数矩阵表达多输入输出(MIMO)系统模型的方法，建议读者查阅有关工具书。

1.2.3 系统的状态空间模型

状态方程是现代控制理论描述系统模型的一种方法。一个线性系统可以采用状态空间形式来表达：

$$\begin{aligned}\dot{x}&=Ax+Bu\\ y&=Cx+Du\end{aligned} \tag{1.2-5}$$

式中，x 为状态向量，u 为输入向量，y 为输出向量。

在 MATLAB 4.2 中，该线性系统可以用状态方程系数矩阵(A, B, C, D)来描述。

在 MATLAB 5 中，用函数 SS 可以建立一个系统状态方程模型，调用格式为

```
SYS = ss (A, B, C, D)
```

其中，A, B, C, D 为系统状态方程系数矩阵。

【例 1.2】 若给定系统的状态方程系数矩阵为

$$A=\begin{bmatrix}-40.4 & -139 & -150\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix},\ B=\begin{bmatrix}1\\0\\0\end{bmatrix},\ C=[0\quad 18\quad 360],\ D=0$$

则可以将其用 MATLAB 形式表达出来：

```
% MATLAB PROGRAM 1-3
a=[-40.4 -391 -150; 1 0 0; 0 1 0]; b=[1 0 0]'; c=[0  18  360]; d=0;
printsys (a, b, c, d)
```

运行结果：

```
a =
                x1            x2            x3
      x1   -40.40000    -391.00000    -150.00000
      x2     1.00000             0             0
      x3           0       1.00000             0
b =
                u1
      x1     1.00000
      x2           0
      x3           0
c =
                x1            x2            x3
      y1           0      18.00000     360.00000
d =
                u1
      y1           0
```

在 MATLAB 5 中，上例可直接调用函数 SS 求得相同结果。

对于离散系统来讲，状态空间模型可以写成

$$\begin{aligned}\dot{x}(k+1)&=Fx(k)+Gu(k)\\ y(k+1)&=Cx(k+1)+Du(k+1)\end{aligned}\tag{1.2-6}$$

同样地，离散系统状态空间模型在 MATLAB 4.2 中也可以简记为(F, G, C, D)。

在 MATLAB 5 中，用函数 SS 也可以建立一个离散系统传递函数模型，其调用格式为

SYS = ss (F, G, C, D, T)

其中，F, G, C, D 为离散系统状态方程系数矩阵；T 为采样时间间隔。

1.2.4 系统的零极点模型

零极点模型实际上是传递函数模型的另一种形式，其方法是对原系统传递函数的分子和分母多项式进行因式分解，以获得系统的零极点表达形式。对于 SISO 系统来讲，其零极点模型可以简记为

$$G(s)=K\frac{(s-z_1)(s-z_2)\dots(s-z_m)}{(s-p_1)(s-p_2)\dots(s-p_n)}\tag{1.2-7}$$

式中，$z_i(i=1,2,\cdots,m)$和$p_j(j=1,2,\cdots,n)$分别为系统的零点和极点，K 为系统增益。

在 MATLAB 4.2 中，不能由零极点表达式建立系统零极点模型，只能由其它模型形式转换得到。

在 MATLAB 5 中，系统的零极点模型可以用函数 ZPK 来直接建立系统零极点模型，其调用格式为

```
SYS = zpk ( [Z], [P], [K] )
```

其中，[Z], [P], [K]分别为系统的零点、极点和增益。

【例 1.3】 系统模型同例 1.1，求系统的零极点模型。

```
% MATLAB PROGRAM 1-4
Sys=zpk(-2,[-15,-25,-0.4],18)
```

运行结果：

```
zero/pole/gain:
          18 (s + 2)
  ----------------------
  (s +15) (s + 25) (s + 0.4)
```

同时，MATLAB 提供了多项式求根的函数 ROOTS 来求系统的零极点，调用格式为

```
Z = roots (NUM)
```

或

```
P = roots (DEN)
```

其中，NUM、DEN 分别为传递函数模型的分子和分母多项式系数所构成的向量。

对于离散系统，也可以用类似的方法获得零极点模型。

1.2.5 系统的模型转换

如上所述，对系统的数学模型描述主要有微分方程模型、传递函数模型、状态空间模型和零极点模型等形式，而这些模型之间确实存在着内在的等效关系。在一些场合下，需要用其中一种形式的模型，而在另一种场合下可能又需要另外形式的模型，所以研究由一种模型转换为另一种模型具有非常重要的意义。

MATLAB 4.2 控制系统工具箱提供了系统模型之间相互转换的函数。如：

(1) 函数 TF2SS 是用来由系统传递函数模型来求取其状态空间模型，调用格式为

```
[A, B, C, D] = tf2ss (NUM, DEN)
```

其中，NUM、DEN 分别为传递函数模型的分子和分母多项式系数向量；返回结果[A, B, C, D]为系统状态空间模型系数矩阵。

(2) 函数 SS2TF 是用来由系统状态空间模型来求取其传递函数模型，调用格式为

```
[NUM, DEN] = ss2tf (A, B, C, D, IU )
```

其中，A, B, C, D 为系统状态方程系数矩阵，IU 指定是哪个输入，返回结果 DEN 和 NUM 分别为传递函数分母和分子多项式系数向量。

(3) 函数 SS2ZP 是用来由系统状态空间模型来求取其零极点模型，调用格式为

[Z, P, K] = ss2zp (A, B, C, D, IU)

其中，Z, P, K 分别为系统零极点模型的零点、极点和增益，其余参数说明同上。

(4) 函数 TF2ZP 是用来由系统传递函数模型来求取其零极点模型，调用格式为

[Z，P，K]= tf2zp (NUM, DEN)

参数说明同上。

(5) 函数 ZP2SS 是用来由系统零极点模型来求取其状态空间模型，调用格式为

[A, B, C, D]= zp2ss (Z, P, K)

参数说明同上。

(6) 函数 SS2SS 是用来由系统状态空间模型来求取其具有相似性的状态空间模型，调用格式为

[At,Bt,Ct,Dt] = ss2ss (A, B, C, D, T)

其中，T 为转换矩阵；(A,B,C,D)和(At,Bt,Ct,Dt)分别为系统转换之前和之后的状态空间表达式系数矩阵。

MATLAB 5 也提供了更为简单的模型转换形式：

Newsys = tf (sys)

可将非传递函数形式的系统模型 sys 转换成传递函数模型 Newsys；

Newsys = zpk (sys)

可将非零极点形式的系统模型 sys 转换成零极点模型 Newsys；

Newsys= ss (sys)

可将非状态空间形式的系统模型 sys 转换成状态空间模型。

【例 1.4】 将系统 $G(z)=\dfrac{0.04z+0.04}{z^3-z^2+81z+0.9}$ 转换为零极点形式，采样周期为 0.01。

用 MATLAB 5 编写程序如下：

```
% MATLAB PROGAM 1-5
num = [0.04   0.04];
den = [1   -1   81   0.9];
sys1 = tf (num, den, 0.01);                 %  传递函数模型
sys2 = zpk (sys1)                           %  转换为零极点形式
```

运行结果：

```
zero / pole / gain:
          0.04 ( z +1)
---------------------------------
(z + 0.01111) (z^ 2 - 1.011z + 81.01)
```

sampling time: 0.01

1.2.6 系统模型参数的获取

在系统分析中，往往需要知道建立的系统模型的某参数以便直接利用参数中的有关数据。为此，MATLAB 5 提供了专用函数 TFDATA, ZPKDATA 和 SSDATA，调用格式分别为

```
[num, den ] = tfdata (sys, 'v')
[z, p, k ] = zpkdata (sys)
[A, B, C, D ] = ssdata (sys)
```

【例 1.5】 求取例 1.4 给定的传递函数模型分子多项式和分母多项式系数向量。

在 MATLAB COMMAND 窗口下键入：

```
[num, den, Ts] = tfdata (sys, 'v')
num =
              0         0       0.0400      0.0400
den = 1.0000   -2.0000     81.0000      0.9000
Ts  = 0.0100
```

1.2.7 系统变量名的设置

在系统运算、连接和分析时，为一目了然地知道系统变量的物理意义及这些物理量之间相互关系，可在系统建模后，对系统变量"定名"。在 MATLAB 5 中可用函数 SET 给变量定名。

【例 1.6】 建立系统 $G(s)=\frac{1}{s+10}$ 模型，且输入名为 Threst，输出名为 Velocity。

用 MATLAB 5 编写程序如下：

```
% MATLAB PROGRAM 1-6
sys = tf (1   [1   10] );
set (sys, 'inputName', 'Thrust', 'outputName', 'Velocity');
sys
```

运行结果：

Transfer function form input "Thrust" to output "Velocity":

$$\frac{1}{s+10}$$

1.3 系统模型的连接

一般地说，一个系统是由许多环节或子系统按一定方式连接起来组合而成，它们之间连接方式有串联、并联、反馈、附加等。MATLAB 5 提供了模型连接函数。

1.3.1 模型串联

函数 SERIES 用于两个线性模型串联，调用格式为

```
sys = series (sys1, sys2 )
```

其中，sys1, sys2 和 sys 如图 1.1 所示。

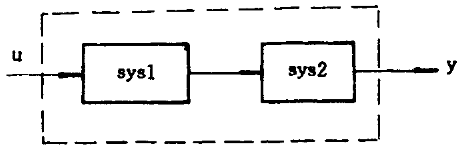

图 1.1 SISO 模型

该函数的执行结果等价于 sys =sys1 * sys2。

对于 MIMO 系统，函数 SERIES 的调用格式为

sys = series (sys1, sys2, outputs1, inputs2)

函数执行系统 sys1 和系统 sys2 串联时，将系统 sys1 的输出端 1 和系统 2 的输入端 2 连接。如图 1.2 所示。系统端口名称可用函数 SET 定名（见 1.2 节）。

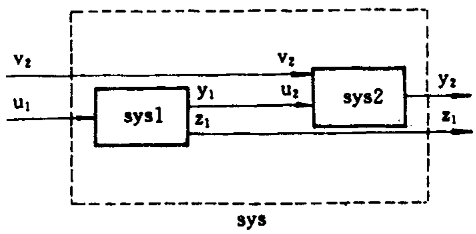

图 1.2 MIMO 模型串联

1.3.2 模型并联

函数 PARALLEL 用于两个模型并联，调用格式为

sys = parallel (sys1, sys2)

其中，sys1, sys2 和 sys 如图 1.3 所示。

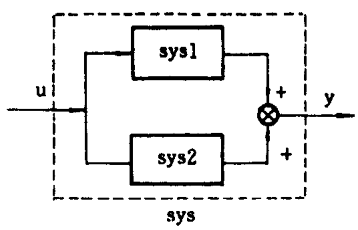

图 1.3 SISO 模型并联

该函数执行结果等价于 sys = sys1 + sys2。

对于 MIMO 系统，函数 PARALLEL 的调用格式为

sys = parallel (sys1, sys2, IN1, IN2, OUT1, OUT2)

函数执行系统 sys1 和 sys2 并联时，将 sys1 的输入端 IN1 和 sys2 的输入端 IN2 连接，sys1 的输出端 OUT1 和 sys2 的输出端 OUT2 连接起来，如图 1.4 所示。