

“十三五”普通高等教育本科规划教材

ZIDONG KONGZHI LILUN ZONGHE SHIYAN JIAOCHENG

自动控制理论 综合实验教程

胡 钊 主编
司马莉萍 编写



“十三五”普通高等教育本科规划教材

自动控制理论 综合实验教程

主编 胡 钊

编写 司马莉萍

主审 华小梅



中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 提 要

本书为“十三五”普通高等教育本科规划教材。

本书为自动控制理论课程的配套实验教材。全书分为两篇，第1篇为自动控制原理 MATLAB 实验，包括 MATLAB 与 Simulink 基础知识和仿真实验，其内容涵盖了经典控制理论和现代控制理论的基本概念和仿真实验方法。第2篇为自动控制理论硬件模拟实验，共包括 15 个控制理论基础实验和 14 个综合创新实验。基础实验既有模拟部分的实验，又有离散部分的实验。综合创新实验涵盖了工程中以转速、温度、液位为控制对象的综合性实验，以及模糊控制、神经元控制、二次型最优控制等智能控制的创新性实验。

本书突出课程的工程实践性，采用“软件仿真+硬件模拟+综合创新”循序渐进的实验模式，既紧密结合实际控制问题，又融入控制理论新成果。

本书既可用作高等院校控制理论课程的实验教材，也可供相关工程技术人员参考。

图书在版编目 (CIP) 数据

自动控制理论综合实验教程 / 胡钊主编; 司马莉萍编写. —北京: 中国电力出版社, 2018.3

“十三五”普通高等教育本科规划教材

ISBN 978-7-5198-1531-8

I. ①自… II. ①胡…②司… III. ①自动控制理论—实验—高等学校—教材 IV. ①TP13-33

中国版本图书馆 CIP 数据核字 (2017) 第 321730 号

出版发行: 中国电力出版社

地 址: 北京市东城区北京站西街 19 号 (邮政编码 100005)

网 址: <http://www.cepp.sgcc.com.cn>

责任编辑: 牛梦洁

责任校对: 马 宁

装帧设计: 张 娟

责任印制: 吴 迪

印 刷: 北京雁林吉兆印刷有限公司

版 次: 2018 年 3 月第一版

印 次: 2018 年 3 月北京第一次印刷

开 本: 787 毫米 × 1092 毫米 16 开本

印 张: 13

字 数: 311 千字

定 价: 30.00 元

版权专有 侵权必究

本书如有印装质量问题, 我社发行部负责退换

前 言

自动控制理论实验课程是自动化类,电气、电子类专业的一门理论性和工程应用性较强的基础性实践教学课程。通过本实验课程,可以巩固和深入理解已学的自动控制理论知识,掌握科学实验方法和技能,为后续课程的学习及今后从事相关控制领域的工作奠定良好的基础。

本书为自动控制理论课程的配套实验教材。全书分为两篇,第1篇包含9章,包括MATLAB与Simulink基础知识和自动控制原理仿真实验,其内容涵盖了控制理论的基本理论和仿真实验方法。第2篇包含3章,为硬件模拟实验,共包括15个控制理论基础实验和14个综合创新实验。基础实验既有模拟部分的实验,又有离散部分实验。综合创新实验涵盖了工程中以转速、温度、液位为控制对象的综合性实验,以及模糊控制、神经元控制、二次型最优控制等智能控制的创新性实验。

本书按照教学规律,从基础到高级,从简单到复杂,以“软件仿真+硬件模拟+综合创新”的实验形式,层层递进,软硬件配合,理论联系实际,使学生真正理解和掌握课堂教学中自动控制概念、原理和方法,实现由“理论知识学习”到“综合运用创新”的转变,培养学生分析探究能力、工程实践能力,达到学以致用目的。本书突出课程的工程性、实践性,实验内容不仅在教学上具有典型性、代表性,而且在工程上具有实用性,尤其是工程中常见的以转速、温度、液位等为控制变量的综合性实验,这些控制方法和实验方案可供工程技术人员开发设计控制系统时借鉴参考。同时,紧跟控制领域前沿科技,紧密结合电气自动化的工程背景,增加了无线电能传输控制等实验,还融入控制理论新的科技发展成果,增加了模糊控制、神经元控制、二次型最优控制等先进控制实验项目。每项实验精心编写,操作性强,有详细的“参考实验步骤”和明确的“实验报告要求”,有助于学生独立完成实验和规范撰写实验报告,“实验思考题”引导学生进行更深入的思考,对实验的数据、现象进行深层次的分析,培养学生的科学研究能力。

本书由胡钊教授任主编,司马莉萍老师任副主编。第1~9章由胡钊编写,第10~12章由司马莉萍编写,全书由胡钊负责统稿和校订。华小梅副教授任主审。在本书的编写过程中,武汉大学电气工程学院的领导和自动控制理论课程组的全体教师提出了很多宝贵的建议,在此向他们表示衷心的感谢。此外,本书中所用硬件实验平台为浙江天煌科技实业有限公司的THKCL-B型模块化自控原理实验系统,部分素材取自该公司友情提供的实验指导书,在此致以深深的谢意。

限于编者水平,书中难免存在不妥之处。衷心希望读者提出宝贵意见。

编 者

2018年1月于珞珈山

目 录

前言

第 1 篇 自动控制原理 MATLAB 实验

第 1 章 MATLAB 的基本操作	1
1.1 命令窗口	1
1.2 MATLAB 的工作空间	1
1.3 M 文件	2
1.4 矩阵运算	3
1.5 MATLAB 常用常量与常用初等函数	5
1.6 实验习题	6
第 2 章 控制系统模型	7
2.1 连续系统模型的生成	7
2.2 系统模型转换	13
2.3 模型的变换与简化	15
2.4 线性系统典型环节的数学模型	23
2.5 实验习题	24
第 3 章 线性系统的时域分析	25
3.1 拉氏变换法	25
3.2 时间响应函数及其说明	28
3.3 系统的动态性能指标	31
3.4 实验习题	33
第 4 章 根轨迹法	35
4.1 根轨迹概念	35
4.2 根轨迹方程	35
4.3 根轨迹绘制	35
4.4 根轨迹分析	36
4.5 实验习题	38
第 5 章 频域分析法	39
5.1 奈奎斯特 (Nyquist) 图	39
5.2 伯德 (Bode) 图	40
5.3 控制系统的相对稳定性	42
5.4 实验习题	43

第 6 章 控制系统的设计与校正	45
6.1 串联超前补偿	45
6.2 串联滞后补偿	47
6.3 串联滞后—超前校正	49
6.4 反馈校正	52
6.5 实验习题	55
第 7 章 离散系统	57
7.1 离散系统建模	57
7.2 离散系统分析	59
7.3 实验习题	61
第 8 章 状态变量控制系统	63
8.1 状态空间模型的建立	63
8.2 状态模型的转化	66
8.3 线性系统的可控性、可观性判定	68
8.4 状态反馈与极点配置	71
8.5 状态观测器	72
8.6 状态空间系统稳定性分析	74
8.7 实验习题	76
第 9 章 Simulink 仿真	78
9.1 Simulink 仿真环境	78
9.2 Simulink 模块库	80
9.3 Simulink 基本操作	84
9.4 Simulink 仿真运行	85
9.5 基于 Simulink 的各类分析	90
9.6 实验习题	96

第 2 篇 自动控制理论硬件模拟实验

第 10 章 实验系统概述	99
10.1 硬件组成与使用	99
10.2 上位机软件的使用	101
第 11 章 基础实验	105
实验 1 典型环节的电路模拟	105
实验 2 二阶系统的瞬态响应	112
实验 3 高阶系统的瞬态响应和稳定性分析	115
实验 4 线性定常系统的稳态误差	117
实验 5 典型环节和系统频率特性的测量	122
实验 6 线性定常系统的串联校正	126
实验 7 典型非线性环节的静态特性	133
实验 8 非线性系统的描述函数法	137

实验 9	非线性系统的相平面分析法	143
实验 10	系统能控性与能观性分析	147
实验 11	控制系统极点的任意配置	149
实验 12	具有内部模型的状态反馈控制系统	155
实验 13	采样控制系统的分析	159
实验 14	采样控制系统的动态校正	162
实验 15	线性系统的根轨迹分析	164
第 12 章	综合创新实验	167
实验 1	数字 PID 调节器算法的研究	167
实验 2	串级控制算法的研究	169
实验 3	解耦控制算法的研究	171
实验 4	模糊控制系统的研究	174
实验 5	具有单神经元控制器的控制系统	177
实验 6	二次型状态调节器	180
实验 7	单闭环直流调速系统	182
实验 8	步进电机转速控制系统	186
实验 9	单闭环温度恒值控制系统	188
实验 10	单容水箱液位定值控制系统	189
实验 11	双容水箱液位定值控制实验	191
实验 12	无线电能传输系统频率特性分析实验	193
实验 13	无线电能传输系统最优负载分析实验	195
实验 14	自激振荡式无线电能传输系统实验	197

第 1 篇 自动控制原理 MATLAB 实验

第 1 章 MATLAB 的基本操作

MATLAB 编程有两种工作方式，一种称为行命令方式，就是在命令窗口中一行一行地输入程序，计算机对每一行命令做出反应，因此也称为交互式指令行操作方式，另一种是 M 文件的编程工作方式。

1.1 命 令 窗 口

在交互式指令行操作方式的命令窗口下，MATLAB 命令语句能即时执行，只要用回车键确定后，MATLAB 就立即对其处理，并得出中间结果。MATLAB 直接赋值语句的一般形式为

>> 赋值变量 = 赋值表达式

这一过程把等号右边的表达式直接赋给左边的赋值变量，当键入回车键时，该语句被执行。语句执行之后，窗口自动显示出语句执行的结果。如果希望结果不显示，则只要在语句之后加上一个分号即可，此时尽管结果没有显示，但它依然被赋值并在 MATLAB 工作空间中分配了内存。

```
例如：>> a=5;
      >> b=9;
      >> c=a * b
```

执行后显示：

```
c=
    45
```

1.2 MATLAB 的工作空间

(1) MATLAB 的工作空间包含一组可以在命令窗口中调整（调用）的参数。MATLAB 的常用帮助命令见表 1-1。

表 1-1 常用帮助命令

命令	功 能
clear	清除当前工作空间中的所有变量
clear a	清除当前工作空间的指定变量 a
home	清除命令窗口中所有内容并将光标移到左上角

续表

命令	功 能
clc	清除命令窗口中所显示的所有内容
pack	整理内存碎片以扩大内存空间
exist	检查指定名字的变量或函数文件是否存在
what	按扩展名分类列出指定目录上的文件名
which	列出指定名字文件所在的目录
who	显示当前工作空间中所有变量的一个简单列表
whos	列出变量的大小、数据格式等详细信息

(2) 保存和载入 workspace。MATLAB 工作空间中的变量在退出 MATLAB 时会丢失。可以调用 save 命令将变量保存到文件中。需要时调用 load 命令将变量从文件中调出到工作空间。

格式为 save 文件名 变量；load 文件名 变量

例如：>> save mydate a b c

即将工作空间的 a、b、c 变量存到文件 mydate.mat 中。

1.3 M 文 件

M 文件可分为脚本文件 (Script File) 和函数文件 (Function File) 两大类，这两种文件的扩展名相同，都是“.m”，故统称为 M 文件，其特点和适用领域均不同。

1. 脚本文件

脚本文件是由 MATLAB 语句构成的文本文件，以 .m 为扩展名。运行命令文件的效果等价于从 MATLAB 命令窗口中按顺序逐条输入并运行文件中的指令。

脚本文件运行过程所产生的变量保留在 MATLAB 的工作空间中，脚本文件也可以访问 MATLAB 当前工作空间的变量，其他脚本文件和函数可以共享这些变量，因此脚本文件常用于主程序的设计。

【例 1-1】 求函数 $y = \sin(|x|) + x^3$ 在 $x=4$ 时的值。

解 在 MATLAB 的 M 文件编辑器下 (左上角有个十号新建脚本)，编辑如图 1-1 所示文本文件 myfun4_1.m。

```

编辑器 - Untitled*
Untitled* x +
1 % myfun4_1
2 % This is a m-file for y=sin(|x|)+x^3
3 x1=abs(x)
4 y=sin(x1)+x.^3

```

图 1-1 【例 1-1】图

以上脚本文件 myfun4_1.m 建立后, 在 MATLAB 命令窗口输入命令

```
>>x = 4;myfun4_1
```

执行后显示

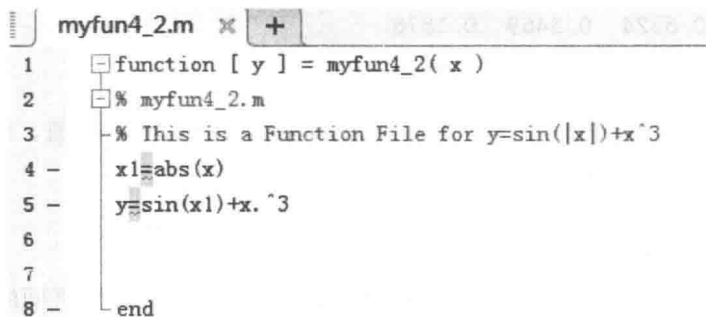
```
x =  
    4  
y =  
   63.2432
```

2. 函数文件

函数文件是 M 文件的另一种类型, 它也是由 MATLAB 语句构成的文本文件, 并以 .m 为扩展名。函数文件的功能是建立一个函数, 它的第一句可执行语句是以 function 引导的定义语句。一般情况下不能单独键入文件名来运行函数文件, 必须由其他语句来调用, 在函数文件中的变量都是局部变量。

【例 1-2】 建立函数文件求 $y = \sin(|x|) + x^3$ 在 $x = 4$ 时的值。

解 在 MATLAB 的 M 文件编辑器下, 编辑如图 1-2 所示的函数文件 myfun4_2.m。



```
myfun4_2.m x +  
1 function [ y ] = myfun4_2( x )  
2 % myfun4_2.m  
3 % This is a Function File for y=sin(|x|)+x^3  
4 x1=abs(x)  
5 y=sin(x1)+x.^3  
6  
7  
8 end
```

图 1-2 【例 1-2】图

以上文本文件 myfun4_2.m 建立后, 在 MATLAB 命令窗口输入命令

```
>>[y] = myfun4_2(4)
```

执行后显示

```
y =  
   63.2432
```

说明: 符号 “%” 引导的行是注释行, 不予执行, 供 help 指令在线查询用。

1.4 矩阵运算

1. 矩阵的表示

矩阵的元素直接排列到方括号中, 每行内的元素间用空格或逗号分开, 行与行间用分号

隔开。

【例 1-3】 在 MATLAB 中表示矩阵 $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, $B = [\pi \quad 1+2i \quad 3-4i]$ 。

解 在 MATLAB 中输入

```
>> A = [1,2,3;4,5,6;7,8,9]
```

```
或>> A = [1 2 3;4 5 6;7 8 9]
```

```
>> B = [pi 1 + 2i 3 - 4i]
```

【例 1-4】 调用 rand () 函数创建一个 3 行 4 列的随机矩阵。

解 在 MATLAB 命令窗口中输入下面语句并按回车键确认。

```
>> A = rand(3,4)
```

运行结果为

A =

```
0.8147 0.9134 0.2785 0.9649
```

```
0.9058 0.6324 0.5469 0.1576
```

```
0.1270 0.0975 0.9575 0.9706
```

MATLAB 提供了一个便利高效的表达式来给等步长的行向量赋值，即冒号表达式，它的基本调用格式为

```
X = a:b:c
```

其中：a、c 为标量，分别代表起始值和终止值，b 代表向量元素之间的步长值。

【例 1-5】 输入冒号表达式 $X = 0 : 0.2 : 1$ 。

解 在 MATLAB 命令窗口中输入下面语句并按回车键确认。

```
>> X = 0 : 0.2 : 1
```

运行结果为

X =

```
0 0.2000 0.4000 0.6000 0.8000 1.0000
```

2. 矩阵元素的表示和赋值

矩阵元素的行号和列号称为该元素的下标，通过“()”中的数字来标识，如 $A(i, j)$ 表示矩阵 A 第 i 行第 j 列的元素。

【例 1-6】 取矩阵 $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ 第二行所有元素。

解 在 MATLAB 命令窗口中输入下面语句并按回车键确认。

```
>> A=[1,2,3;4,5,6;7,8,9];B=[A(2,1),A(2,2),A(2,3)]
```

运行结果为

```
B =
    4    5    6
```

此处冒号有很大用处，如 $A(2, :)$ 表示 A 第二行全部元素， $A(:, 2)$ 表示 A 第二列全部元素， $A(1, 1:2)$ 表示 A 第 1 行第 1~2 列全部元素。

如在 MATLAB 命令窗口中输入下面语句并按回车键确认。

```
>> A=[1,2,3;4,5,6;7,8,9];B1=A(2,:),B2=A(:,2),B3=A(1,1:2)
```

运行结果为

```
B1 =
    4    5    6
```

```
B2 =
    2
    5
    8
```

```
B3 =
    1    2
```

1.5 MATLAB 常用常量与常用初等函数

在使用 MATLAB 的过程中，需要记住其中的一些常用常量及常用初等函数。表 1-2 与表 1-3 为 MATLAB 常用常量与常用初等函数。

表 1-2 MATLAB 常用常量

常量	含义	常量	含义
i, j	虚数单位	Inf	正无穷大的 MATLAB 表示
pi	圆周率的双精度浮点表示	NaN	不定式
eps	机器的浮点运算误差线	lasterr	存放最新的错误消息
realmax	最大的正实数	lastwarn	存放最新的警告信息
realmin	最小的正实数	nargin	函数的输入变量数目
ans	缺省变量名，以应答最近一次操作结果	nargout	函数的输出变量数目

表 1-3 常用初等函数

sin	正弦函数	real	求复数的实部
cos	余弦函数	image	求复数的虚部
tan	正切函数	conj	求复数的共轭

续表

abs	求实数绝对值或复数的模	exp	自然指数函数
sqrt	平方根函数	log	自然对数函数
angle	求复数的幅角	log10	以 10 为底的对数函数

1.6 实 验 习 题

1-1 在命令窗口中直接计算出 $\sqrt{351 \times 12 - 64 \times 5}$ 的值。

1-2 试编写 M 文件和函数文件分别求函数 $y = \log(|x-3|+1)$ 在 $x=5$ 时的值。

1-3 有矩阵 $A = \begin{bmatrix} 2 & 5 & 4 \\ -1 & 12 & 3 \\ 2 & 4 & 6 \end{bmatrix}$, $B = \begin{bmatrix} 3 & 2 \\ 4 & 4 \\ 5 & 1 \end{bmatrix}$, 试取出 A 中第二列全部元素和 B 中前两

行的元素。

第 2 章 控制系统模型

2.1 连续系统模型的生成

要对系统进行仿真处理，首先要知道系统的数学模型，才能对系统进行分析设计。在线性系统理论中，一般常用的数学模型有传递函数模型（系统的外部模型）、状态方程模型（系统的内部模型）、零极点增益模型和部分分式模型。这些模型之间都有内在联系，可以相互进行转换。控制系统的数学模型按系统性能分为线性系统和非线性系统、连续系统和离散系统、定常系统和时变系统、确定系统和不确定系统。

在 MATLAB 中，用于生成连续线性系统模型的函数有传递函数建模函数 `tf()`、零极点增益函数 `zpk()`、状态空间建模函数 `ss()`。常用连续系统建模的函数命令格式及说明见表 2-1。

表 2-1 常用连续系统建模的函数命令格式及说明

函数命令格式	功能说明
<code>S=tf('s')</code>	生成以 s 为变量的传递函数 s 。此时 s 既是传递函数也是制定变量
<code>sys=tf(num, den)</code>	生成传递函数模型， num 、 den 分别为模型的分子和分母多项式系数向量
<code>sys=tf(num, den, 'td', v)</code>	生成延迟时间 $t_d=v$ 的传递函数模型
<code>sys=(z, p, k)</code>	生成零极点增益模型， z 、 p 、 k 分别为零点、极点和增益向量
<code>[num, den]=rmodel(n, P)</code>	随机生成一个 n 阶连续的传递函数模型，该系统具有 P 个输出
<code>[num, den]=ord2(wn, z)</code>	生成固有频率为 w_n ，阻尼系数为 z 的连续二阶系统模型系统
<code>[num, den]=pade(L, n)</code>	返回延迟环节 $G(s)=e^{-Ls}$ 近似为 n 阶多项式传递函数的 num 和 den

2.1.1 有理多项式分式传递函数模型的建立

系统传递函数的有理多项式分式模型通常表示为

$$G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}, m \leq n \quad (2-1)$$

式 (2-1) 中，分子和分母均为多项式表达式。对于线性定常系统， s 的系数均为常数，且 $a_n \neq 0$ 。

多项式分式由其分子和分母的各项系数唯一确定，因此在 MATLAB 中，多项式是用其降幂系数的排列构成的数组来表示，称为多项式系数向量。MATLAB 的多项式运算是指对系数向量的运算。

在 MATLAB 中，分子和分母的多项式系数向量分别表示为

$$\begin{aligned} \text{num} &= [b_m, b_{m-1}, \dots, b_1, b_0] \\ \text{den} &= [a_n, a_{n-1}, \dots, a_1, a_0] \end{aligned}$$

注意：它们都是按 s 的降幂排列，分子为 $m+1$ 项，分母为 $n+1$ 项，在系数向量中的相应位置上用 0 补充。

【例 2-1】 已知系统的分式多项式传递函数为

$$G(s) = \frac{6s^3 + 2s^2 + 12s + 20}{2s^4 + 4s^3 + 6s^2 + 2s + 2}$$

试在 MATLAB 中建立系统的传递函数模型。

解 方法一：直接用分子和分母多项式系数建立传递函数模型。

```
>> clear;
>> num = [6,2,12,20];
>> den = [2,4,6,2,2,];
>> G = tf(num,den)
```

运行结果

```
G =
    6 s^3 + 2 s^2 + 12 s + 20
-----
    2 s^4 + 4 s^3 + 6 s^2 + 2 s + 2
Continuous-time transfer function.
```

方法二：用 s 因子和数学运算符建立传递函数模型。

```
>> s = tf('s'); G = (6 * s^3 + 2 * s^2 + 12 * s + 20) / (2 * s^4 + 4 * s^3 + 6 * s^2 + 2 * s + 2)
G =
    6 s^3 + 2 s^2 + 12 s + 20
-----
    2 s^4 + 4 s^3 + 6 s^2 + 2 s + 2
Continuous-time transfer function
```

注意：如果传递函数的分子分母为多个多项式相乘，如

$$G(s) = \frac{(s^2 + 2s + 3)(s^2 + 0.1s + 0.2)}{(s^2 + 1.1s + 2.3)(s^2 + 4s + 1)}$$

可以使用 conv() 函数来解决多项式展开的问题，上述传递函数可用如下所示代码实现

```
>> num = conv([1 2 3],[1 0.1 0.2]);den = conv([1 1.1 2.3],[1 4 1]);sys = tf(num,den)
sys =
    s^4 + 2.1 s^3 + 3.4 s^2 + 0.7 s + 0.6
-----
    s^4 + 5.1 s^3 + 7.7 s^2 + 10.3 s + 2.3
Continuous-time transfer function.
```

【例 2-2】 已知系统的传递函数为

$$G(s) = \frac{3(s+3)(s^2+5s+3)^2}{s(s+1)^3(s^3+3s^2+2s+5)}$$

试在 MATLAB 中建立上述系统模型。

解 当传递函数中分子、分母含有多项式混合乘项时，可用以下两种方法处理：

(1) 先利用多项式乘法命令函数 conv() 对分子或分母进行运算，直接得到多项式系数向量，再建立传递函数模型。

(2) 先建立函数模型 s，再对 s 进行传递函数模型 G 的运算。

方法一：

```

>> num = 3 * conv([1,3],conv([1,5,3],[1,5,3])); % 分子多项式相乘运算
>> den = conv([1,0],conv([1,1],conv([1,1],conv([1,1],[1,3,2,5])))); % 分母多项式相乘
>> sys2 = tf(num,den) % 建立 MATLAB 模型
sys2 =
    3 s^5 + 39 s^4 + 183 s^3 + 369 s^2 + 297 s + 81
-----
s^7 + 6 s^6 + 14 s^5 + 21 s^4 + 24 s^3 + 17 s^2 + 5 s
Continuous-time transfer function.

```

方法二:

```

>> s = tf([1 0],1) % 建立 s 的传递函数模型,并用 s 表示该模型
s =
s
Continuous-time transfer function.
>> G = 3 * (s+3) * (s^2+5*s+3)^2/s/(s+1)^3/(s^3+3*s^2+2*s+5) % 对传递函数 s 进行 G 的运算
G =
    3 s^5 + 39 s^4 + 183 s^3 + 369 s^2 + 297 s + 81
-----
s^7 + 6 s^6 + 14 s^5 + 21 s^4 + 24 s^3 + 17 s^2 + 5 s

```

或者

```

>> s = tf('s');
>> G = 3 * (s+3) * (s^2+5*s+3)^2/s/(s+1)^3/(s^3+3*s^2+2*s+5) % 对传递函数 s 进行 G 的运算
G =
    3 s^5 + 39 s^4 + 183 s^3 + 369 s^2 + 297 s + 81
-----
s^7 + 6 s^6 + 14 s^5 + 21 s^4 + 24 s^3 + 17 s^2 + 5 s

```

2.1.2 零极点传递函数模型的建立

控制系统的数学模型可用零点、极点和增益来表示,其原理是分别对传递函数的分子和分母进行因式分解处理,以获得系统的零点和极点的表示形式。MATLAB 称这种数学模型为零极点增益模型,即 zpk 模型,并用 zpk() 函数来建立这种数学模型。

零极点传递函数模型一般可表示为

$$G(s) = K \frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)} \quad (2-2)$$

式(2-2)中, K 为增益, z 为零点, p 为极点。

【例 2-3】 已知系统的零极点增益模型为

$$G(s) = \frac{5(s+2)}{s(s^2+2s+2)}$$

试建立零极点增益模型。

解 方法一: 直接建立 zpk 模型。

```

>> k=5;z=[-2];P=[0,-1+j,-1-j];
>> G = zpk(z,p,k)
G =

```


$$\frac{5(s+2)}{s(s^2+2s+2)}$$

Continuous-time zero/pole/gain model.

方法二：用 s 因子和数学运算符号建立模型。

```
>> s = zpk('s');
>> H = 5 * (s + 2) / (s * (s^2 + 2 * s + 10));
```

【例 2-4】 试生成一个零点为 2 和 3，极点为 $-1 \pm i$ ，增益为 5 的零极点增益模型。

解 建立该零极点条件下模型的程序如下：

```
>> z = [2;3];p = [-1; -1 + i; -1 - i];k = [5];G = zpk(z,p,k)
```

G =

$$\frac{5(s-2)(s-3)}{(s+1)(s^2+2s+2)}$$

Continuous-time zero/pole/gain model.

2.1.3 二阶系统模型

标准二阶系统的规范表示为

$$G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (2-3)$$

常用命令格式：`[num, den]=ord2(wn, z)`;

$$G = \text{wn}^2 * \text{tf}(\text{num}, \text{den}),$$

其中：wn 为自然振荡角频率 ω_n ，z 为阻尼比 ξ 。

【例 2-5】 试生成一个自然振荡频率 $\omega_n=3$ ，阻尼比为 0.45 的二阶系统模型参数。

解

```
>> clear;wn = 3;z = 0.45; % 设置自然振荡角频率和阻尼比
>> [num,den] = ord2(wn,z)
```

num =

1

den =

1.0000 2.7000 9.0000

注意：MATLAB 生成的二阶系统模型分子系数为 1，而标准二阶模型的分子为 ω_n^2 。所以正确的结果应该使分子乘以 ω_n^2 ，即可用 `G=wn^2*tf(num,den)` 直接的二阶系统的 MATLAB 模型。程序为

```
>> clear;wn = 3;z = 0.45;
>> [num,den] = ord2(wn,z);
>> G = wn^2 * tf(num,den)
```

G =