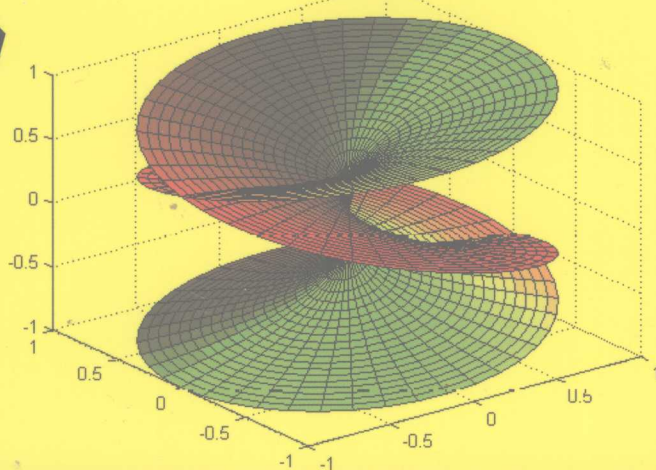


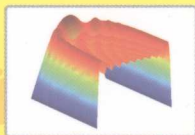
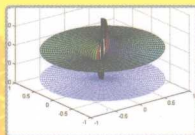
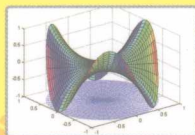
$z(1/3)$



# MATLAB

## 实用教程

### ——控制系统仿真与应用



于浩洋 初红霞 王希凤 等编著



化学工业出版社

图在空域(3D)自激振荡并图

MATLAB应用——控制系统的仿真与应用

ISBN 978-7-122-04492-4

# MATLAB

## 实用教程

### ——控制系统仿真与应用

于浩洋 初红霞 王希凤 等编著



化学工业出版社

· 北京 ·

图书在版编目(CIP)数据

MATLAB 实用教程——控制系统仿真与应用 / 于浩洋, 初红霞, 王希凤等编著. —北京: 化学工业出版社, 2009.6

ISBN 978-7-122-04988-9

I. M… II. ①于…②初…③王… III. 自动控制系统—系统仿真—软件包, MATLAB—教材 IV. TP273

中国版本图书馆 CIP 数据核字 (2009) 第 028554 号

MATLAB

实用教程

——  
控制系统仿真软件包

---

责任编辑: 宋 辉

文字编辑: 徐卿华

责任校对: 王素芹

装帧设计: 王晓宇

---

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 三河市延风印装厂

720mm×1000mm 1/16 印张 21¼ 字数 392 千字 2009 年 5 月北京第 1 版第 1 次印刷

---

购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

---

定 价: 48.00 元

版权所有 违者必究

# 前 言

MATLAB 作为当前国际控制界最流行的面向工程与科学计算的高级语言,近年来得到了业界的一致认可,在控制系统的分析、仿真与设计方面得到了非常广泛的应用,其自身也因此得到了迅速的发展,功能不断扩充,现已发展到 7.7 版本。为了更好地推动 MATLAB 在控制系统分析、仿真与设计中的应用,在借鉴以往类似书籍与教材经验并弥补其中不足的基础上,编者结合多年来的教学和科研工作编著了此书。

本书内容深入浅出、图文并茂,力求通俗易懂,各章之间既相互联系又相互独立,读者可以根据自己的需要进行选择。全书从实际出发,对常规函数的功能、格式和参数作了较详细的说明,通过大量的典型实例对 MATLAB/Simulink 的功能、操作及其在自动控制中的应用进行了较为详细的论述。书中所述内容和例子,在教学和科研中均做过试验与验证,是编者多年教学与科研的结晶。

本书共分 7 章,包括: MATLAB 简介、控制系统的分析与仿真、神经网络控制的分析与仿真、模糊逻辑控制的分析与仿真、模型预测控制的分析与仿真、系统辨识分析与仿真、工程应用。所有的例子都提供了相应的 MATLAB 程序或仿真模型,便于读者掌握和巩固所学的知识。

本书第 1 章、第 4 章的 4.1、4.2、4.6~4.8 和第 7 章由黑龙江工程学院于浩洋编写;第 2 章的 2.1~2.3 和第 3 章由黑龙江工程学院初红霞编写;第 5 章的 5.7 和第 6 章由黑龙江工程学院王希凤编写;第 4 章的 4.3~4.5 由哈尔滨理工大学的袁丽英编写;第 5 章的 5.1~5.6 由黑龙江科技学院朱显辉编写;第 2 章的 2.4~2.7 由天津天士力之骄药业有限公司设备部周强编写;全书由于浩洋统稿,由哈尔滨工程大学王科俊教授负责主审。

本书通用性和实用性较强,既可供自动控制专业人员参考,也可作为高等院校控制类专业本科生及研究生教材。

由于编者水平有限,加之本书的知识覆盖面较广,因此难免有遗漏与不当之处,恳请广大读者批评指正。

编著者

# 目 录

<b>第 1 章</b>	<b>1</b>
<b>MATLAB 简介</b>	<b>1</b>
1.1 MATLAB 语言 .....	1
1.2 机器配置要求 .....	2
1.3 MATLAB 的接口 .....	2
1.4 Simulink 的新特性 .....	3
<b>第 2 章</b>	<b>4</b>
<b>控制系统的分析与仿真</b>	<b>4</b>
2.1 LTI 对象模型的建立及转换 .....	4
2.1.1 模型建立函数 .....	4
2.1.2 控制系统模型转换与化简 .....	7
2.2 控制系统时域分析的 MATLAB 实现 .....	10
2.2.1 MATLAB 函数指令方式下的时域响应仿真 .....	10
2.2.2 利用 Simulink 动态结构图的时域响应仿真 .....	11
2.3 控制系统频域分析的 MATLAB 实现 .....	12
2.3.1 Bode 图 .....	12
2.3.2 Nyquist 曲线 .....	12
2.3.3 用频率法判定系统稳定性的 MATLAB 实现 .....	13
2.4 控制系统稳定性分析的 MATLAB 实现 .....	16
2.4.1 控制系统稳定性分析方法简述 .....	16
2.4.2 稳定性分析 MATLAB 实现的方法与举例 .....	16
2.5 控制系统设计及实例 .....	22

2.5.1	简单闭环控制系统的 MATLAB 计算及仿真	22
2.5.2	多闭环控制系统的 MATLAB 计算及仿真	30
2.6	系统分析的 GUI 函数	39
2.7	现代控制系统设计与仿真	42
2.7.1	系统的可控性分析	42
2.7.2	系统的可观测性分析	48
2.7.3	极点配置	52
2.7.4	状态观测器设计	55

## 第 3 章

57

### 神经网络控制的分析与仿真

3.1	神经网络工具箱概述	57
3.2	神经网络工具箱中的通用函数	57
3.3	感知器网络及其 MATLAB 实现	64
3.3.1	重要的感知器神经网络函数	64
3.3.2	感知器神经网络的 MATLAB 仿真程序设计	65
3.4	线性神经网络及其 MATLAB 实现	69
3.4.1	重要的线性神经网络函数	69
3.4.2	线性神经网络 MATLAB 仿真程序设计	71
3.5	BP 网络及其 MATLAB 实现	73
3.5.1	重要的 BP 网络函数	73
3.5.2	BP 网络的 MATLAB 仿真程序设计	79
3.6	径向基神经网络及其 MATLAB 实现	83
3.6.1	重要的径向基神经网络函数	83
3.6.2	径向基神经网络的 MATLAB 仿真程序设计	84
3.7	其他常用神经网络及其 MATLAB 实现	85
3.7.1	重要的自组织竞争神经网络函数	85
3.7.2	重要的回归网络函数	91
3.8	图形用户界面	92
3.8.1	图形用户界面简介	92
3.8.2	图形用户界面应用示例	93
3.8.3	图形用户界面的其他操作	97
3.9	Simulink 神经网络仿真	98
3.9.1	Simulink 神经网络仿真模型库简介	98

3.9.2 Simulink 应用示例	101
3.10 神经网络的应用实例	104

## 第 4 章

114

### 模糊逻辑控制的分析与仿真

4.1 概述	114
4.2 模糊逻辑工具箱的图形界面	115
4.2.1 模糊控制工具箱	115
4.2.2 模糊逻辑工具箱的图形界面	119
4.3 用模糊工具箱命令创建模糊系统	133
4.3.1 模糊语言变量的隶属度函数	133
4.3.2 模糊推理系统数据结构管理函数	140
4.3.3 常规模糊控制器的设计	157
4.4 Simulink 设计基础	168
4.4.1 运行 Simulink	168
4.4.2 Simulink 模块操作	169
4.5 模糊聚类	170
4.5.1 引言	170
4.5.2 模糊聚类的相关函数	171
4.5.3 聚类 GUI 工具	175
4.6 Sugeno-Type 模糊推理	175
4.6.1 引言	175
4.6.2 两个 lines	176
4.6.3 结论	178
4.7 模糊控制器的设计与仿真	179
4.7.1 模糊+PID 控制器的设计与仿真	179
4.7.2 模糊 PID 控制器的设计与仿真	181
4.7.3 多模糊控制器的设计与仿真	183
4.7.4 自调整因子模糊控制器的设计与仿真	186
4.7.5 模糊积分控制器的设计与仿真	188
4.8 模糊控制的应用实例	190
4.8.1 质量-弹簧-阻尼系统的模糊控制	190
4.8.2 对工业流化床锅炉床温的控制	193

## 第5章

197

### 模型预测控制的分析与仿真

5.1	模型预测基本原理及系统模型	198
5.2	系统模型辨识函数	199
5.2.1	数据向量或矩阵的归一化辨识函数	199
5.2.2	基于线性回归方法的脉冲响应模型辨识函数	201
5.2.3	脉冲响应模型转换为阶跃响应模型	202
5.2.4	模型的校验辨识函数	203
5.3	系统模型建立与转换函数	203
5.3.1	系统模型的建立	203
5.3.2	系统模型的转换函数	205
5.4	系统分析与绘图函数	213
5.5	控制器设计与仿真	219
5.5.1	基于阶跃响应模型预测控制器设计与仿真	219
5.5.2	基于状态空间模型的预测控制器设计与仿真	222
5.6	模型预测控制工具箱的通用功能函数	226
5.7	模型预测控制的应用实例	239

## 第6章

244

### 系统辨识分析与仿真

6.1	系统辨识的基本原理和常用辨识模型	245
6.1.1	系统辨识的基本原理	245
6.1.2	常用的模型类	246
6.2	系统辨识工具箱函数	247
6.2.1	模型类的建立和转换	247
6.2.2	非参数模型辨识	258
6.2.3	参数模型辨识	263
6.2.4	递推参数模型辨识	272
6.2.5	模型验证与仿真函数	276
6.2.6	其他常用功能函数	284
6.3	系统辨识工具箱的交互式图形界面	293



6.3.1	数据视图 .....	294
6.3.2	操作选择 .....	294
6.3.3	模型视图 .....	294
6.4	系统辨识仿真实例 .....	295

## 第 7 章

### 工程应用

298

7.1	倒立摆控制 .....	298
7.1.1	倒立摆模型 .....	298
7.1.2	开环响应 .....	299
7.1.3	PID 控制算法的 MATLAB 仿真 .....	303
7.2	烤箱控制 .....	306
7.2.1	烤箱模型 .....	307
7.2.2	具有零极点补偿的积分控制 .....	310
7.2.3	烤箱的离散状态表示 .....	311
7.3	电磁悬浮 .....	314
7.3.1	过程模型 .....	315
7.3.2	电流放大器控制系统 .....	318
7.3.3	$x(t)$ 位置控制系统的连续和离散模型 .....	319
7.3.4	模糊调节器控制 .....	321

### 参考文献

329

# 第 1 章

## MATLAB 简介

### 1.1 MATLAB 语言

MATLAB 语言是一种高级矩阵语言。随着 Mathworks 公司的不断研究, MATLAB 语言已成为带有独特的数据结构、输入/输出、结构控制语句和函数, 并且是面向对象的高级语言。MATLAB 语言在工程计算方面具有无可比拟的优异性能。它集计算、数据可视化和程序设计于一体, 并能将问题和解决方案以用户熟悉的数学符号表示出来。MATLAB 具有以下显著特点。

#### (1) 功能强大

##### ① 运算功能强大

a. MATLAB 的数值运算要素不是单个数据, 而是矩阵, 每个元素都可看作复数, 运算包括加、减、乘、除、函数运算等。

b. 通过 MATLAB 的符号工具箱, 可以解决在数学、应用科学和工程计算领域中常常遇到的符号计算问题。

② 功能丰富的工具箱 大量针对各专业应用的工具箱的提供, 使 MATLAB 适用于不同领域。

③ 文字处理功能强大 MATLAB 的 Notebook 为用户提供了强大的文字处理功能, 允许用户从 Word 访问 MATLAB 的数值计算和可视化结果。

#### (2) 人机界面友好, 编程效率高

① 语言规则与笔算式相似, 命令表达方式与标准的数学表达式非常相近。

② 采用解释方式工作, 键入算式无需编译立即得出结果, 若有错误也立即作出反应, 便于编程者立即改正。

#### (3) 强大而智能化的作图功能

① 工程计算的结果可视化, 使原始数据的关系更加清晰明了。

② 多种坐标系。

③ 能绘制三维坐标中的曲线和曲面。

#### (4) 可扩展性强

包括基本部分和工具箱两大部分, 具有良好的可扩展性, 工具箱可以任意增减。

### (5) Simulink 动态仿真功能

MATLAB 的 Simulink 提供了动态仿真的功能, 用户通过绘制框图来模拟一个线性、非线性、连续或离散的系统, 通过 Simulink 能够仿真并分析该系统。

## 1.2 机器配置要求

目前使用的 MATLAB 版本多在 7.0 以上, 它可以被安装在 PC 机和 Macintosh 上, 支持 UNIX、OS 和 Windows 操作系统, 而且无论是单机还是网络环境都可以发挥其卓越的性能。同时, 随着 MATLAB 版本的提高, MATLAB 的功能也越来越强大。MATLAB7.0 以上的版本对系统的要求较高, 为了加快运行速度, 机器的配置应该较好。Mathworks 公司对 Windows 操作系统下的机器配置提出了下述要求。

- ① 操作系统: WindowsXP、Windows2000(Service Pack 3 or 4)、WindowsNT4.0(Service Pack 5 or 7a)。
- ② 处理器: PentiumIII 以上, Xeon、Pentium M、AMD Athlon、Athlon XP、Athlon MP。
- ③ 内存要求: 最小 256MB, 推荐 512MB。
- ④ 硬盘空间: 至少 1G, 如果全部安装, 需要大约 2G 以上。
- ⑤ 显卡: 强烈推荐 16 位、24 位或 32 位支持 OpenGL 图形适配器的显卡。
- ⑥ 其他要求: Windows 支持的声卡、打印机等硬件, 运行 MATLAB7.0 以上版本的某些程序可能需要运行 Office2000 或 OfficeXP。

## 1.3 MATLAB 的接口

### (1) MATLAB 应用程序接口

MATLAB 应用程序接口 (API) 实际上是一个让 MATLAB 语言同 C、Fortran 等其他高级语言进行交互的函数库。该函数库的函数通过动态链接来读写 MATLAB 文件。

MATLAB 应用程序接口的主要功能如下。

- ① 从 MATLAB 中调用 C 和 Fortran 程序。
- ② 从 MATLAB 中输入、输出数据。
- ③ 在 MATLAB 和其他应用程序之间建立客户机/服务器关系。
- ④ 产生可被其他编程语言调用的 COM 组件。

## (2) MATLAB 的仿真及硬件接口

为了使设计者更快地将算法设计变为硬件实现, MATLAB 也提供了一套和硬件接口的方法。例如, MATLAB 可以自动产生基于硬件描述语言的滤波器, 使得滤波器从设计到实现变得简单快捷。MATLAB 还支持 TI 公司的 DSP 设计, 在 Simulink 中仿真完成以后, 直接生成 DSP 的 C 语言代码, 打破了从算法到实现之间的鸿沟。MATLAB 也可以将算法生成 COM 组件, 被其他程序调用, 结合 VB 和 MATLAB 的 COM 组件, 算法的编写容易了许多。

## 1.4 Simulink 的新特性

Simulink 是交互式动态系统建模、仿真和分析的图形环境, 是进行基于模型的嵌入式系统开发的基础开放环境。可以针对控制系统、信号处理以及通信系统等进行系统的建模、仿真和分析等工作。当前的版本不仅性能上有了改善, 而且针对大规模系统开放进行了性能优化, 这其中包括以下几项。

- ① 大系统建模: 支持将大系统建模分割为不同的文件, 每个文件为独立的系统模型。
- ② 支持系统不同模型文件的独立仿真测试。
- ③ 增强系统集成手段, 支持配置管理和版本控制软件。
- ④ 递增式模型加载与代码生成功能。
- ⑤ 针对大模型系统提高运行速度和效率。
- ⑥ 模型工作空间 (Model Workspace): 每个模型都有独立的工作空间用于参数管理和数据处理。
- ⑦ Simulink 和 Stateflow 增强总线支持。
- ⑧ 统一的模型浏览器 (Model Explorer): 用于浏览、维护、配置、搜索、定义所有模型中相关的参数、数据对象和属性。
- ⑨ 统一的仿真和代码生成选项。
- ⑩ 支持创建并保持多种仿真和代码生成选项。
- ⑪ 数据管理和可视化。
- ⑫ 新增数据对象属性。
- ⑬ 可选数据记录增加测试点, 无需在模型中增加额外的模块。
- ⑭ I/O 管理, 可以将必要的信号源和信宿连接到模型而不需要增加模块。

## 第 2 章

# 控制系统的分析与仿真

MATLAB 的控制系统工具箱, 主要处理以传递函数为要主要特征的经典控制和以状态空间为主要特征的现代控制理论中的问题。

MATLAB 的控制系统工具箱对线性时不变 (LTI) 系统提供了建模、分析、设计的一套完整方案, 总的来说, 该工具箱具有以下几方面的功能。

### (1) 系统建模

通过控制系统工具箱提供的函数, 可以方便地建立离散系统和连续系统的状态空间、传递函数、零极点增益和频率响应模型, 并可实现任意两种模型间的转换, 而且, 可以通过组合连接两种或多种系统, 从而实现一个复杂的系统模型。

### (2) 系统分析

MATLAB 控制系统工具箱支持对 MIMO 和 SISO 系统进行分析。

① 时域响应分析可支持对系统的单位阶跃响应、单位脉冲响应、零输入响应以及更广泛的对任意信号进行仿真。

② 频率响应分析支持 Bode 图、Nichols 图、Nyquist 图。

### (3) 系统设计

系统设计支持自动控制系统的设计及校正; 支持系统的可观、可控标准型实现; 可以进行系统的极点配置以及状态观测器的设计。

## 2.1 LTI 对象模型的建立及转换

MATLAB 控制系统工具箱对 LTI 系统的建模分析和设计提供了大量完善的工具函数。

### 2.1.1 模型建立函数

#### (1) 传递函数模型

在 MATLAB 语言中, 可以利用分别定义的传递函数分子、分母多项式系数向量方便地对其加以描述。通常定义系统传递函数的分子、分母多项式系数向量为

$$\text{num}=[a_1 \ a_2 \ \cdots \ a_{m-1} \ a_m]; \ \text{den}=[b_1 \ b_2 \ \cdots \ b_{n-1} \ b_n]$$

其中, 分子、分母多项式系数向量中的系数均按  $s$  的降幂排列。函数用 `tf` 来建立传递函数的系统模型。

格式: `sys=tf(num,den)`

说明: `sys=tf(num,den)`生成连续时间系统的传递函数模型, `num` 和 `den` 分别为分子和分母, 返回 `sys` 为 `tf` 对象。

注意: 此时系数必须按照  $s$  的降幂排列。具体 `tf` 函数命令见 `help`。

**【例 2-1】** 现有如下的 SISO 系统, 其传递函数为  $G(s) = \frac{3s+1}{2s^2+5s+11}$ ,

试创建该系统的传递函数模型。

解 依题意, 调用 MATLAB 函数命令 `tf()` 执行如下的 MATLAB 程序 `example2-1.m`:

```
%MATLAB PROGRAM example2-1.m
num=[3 1];
den=[2 5 11];
g=tf(num,den)
```

程序运行后即得到该系统的传递函数模型如下:

```
Transfer function:
      3 s + 1
-----
    2 s^2 + 5 s + 11
```

## (2) 零极点模型 (zpk)

连续系统传递函数表达式是用系统增益、系统零点与系统极点来表示的, 叫作系统零极点增益模型。可以说系统零极点增益模型是传递函数模型的一种特殊形式, 即有

$$G(s) = k \frac{(s+z_1)(s+z_2)\cdots(s+z_m)}{(s+p_1)(s+p_2)\cdots(s+p_n)}$$

格式: `sys=zpk(num,den)`

`sys=zpk(num,den,Ts)`,  $T_s$  为采样周期

说明: `sys=zpk(num,den)`函数返回的变量 `sys` 为连续时间系统的零极点增益模型, 函数输入参量的含义同 `tf()`函数命令的解释。需要指出, 对于已知的零极点增益模型传递函数, 其零点与极点可分别由 `sys.z{1}`与 `sys.p{1}`指令求出。

**【例 2-2】** 已知离散系统的状态空间方程为

$$x(k+1) = \begin{bmatrix} -2.8 & -1.4 & 0 & 0 \\ 1.4 & 0 & 0 & 0 \\ -1.8 & -0.3 & -1.4 & -0.6 \\ 0 & 0 & 0.6 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} u(k)$$

$$y(k) = [0 \ 0 \ 0 \ 1]x(k)$$

试求采样周期  $T_s=0.1s$  时系统的传递函数模型的两个向量 **num**、**den** 与系统的零极点增益模型。

解 由系统的状态空间模型, 输入下面的 MATLAB 程序 **example2-2.m**:

```
%MATLAB PROGRAM example2-2.m
a = [-2.8 -1.4 0 0;1.4 0 0 0;-1.8 -0.3 -1.4 -0.6;0 0
0.6 0];
b = [1;0;1;0];
c = [0 0 0 1];
d = [0];
sys = ss(a,b,c,d,0.1);
sys1 = tf(sys);
num = sys1.num{1}
den = sys1.den{1}
sys2 = zpks(sys)
```

运行以上语句段可得采样周期  $T_s=0.1s$  时系统的传递函数模型的两个向量 **num**、**den** 与系统的零极点增益模型分别为:

```
num = 0 0 0.6000 0.6000 0.9240
den = 1.0000 4.2000 6.2400 3.7520 0.7056
Zero/pole/gain:
0.6 (z^2 + z + 1.54)
-----
(z+1.4)^2 (z+1.061) (z+0.3394)
Sampling time: 0.1
```

### (3) 状态空间模型 (ss)

格式:  $\text{sys}=\text{ss}(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d})$

说明:  $\text{sys}=\text{ss}(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d})$  生成连续系统的状态空间模型, 其中 **a**、**b**、**c**、**d** 分别为相应系统的 **A**、**B**、**C**、**D** 参数矩阵。

$\text{sys}=\text{ss}(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d},T_s)$  生成离散系统的状态空间模型,  $T_s$  为采样周期, 当  $T_s=-1$  或者  $T_s=[]$  时, 则系统的采样周期未定义。

**【例 2-3】** 造纸工业中的一加压液流箱系统。该系统的状态变量是箱中的液位  $h(t)$  与料浆的总压头  $G(t)$ , 输入变量是料浆流入量  $u_1(t)$  与空气流入量  $u_2(t)$ , 输出变量是状态变量  $G(t)$  与  $h(t)$  本身。系统状态空间模型为

$$\begin{cases} \begin{bmatrix} \dot{G}(t) \\ \dot{h}(t) \end{bmatrix} = \begin{bmatrix} 0.5620 & 0.05114 \\ 0.254 & 0 \end{bmatrix} \begin{bmatrix} G(t) \\ h(t) \end{bmatrix} + \begin{bmatrix} 0.03247 & 1.145 \\ 0.112541 & 0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}, \text{ 试用矩阵} \\ \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} G(t) \\ h(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \end{cases}$$

组 $[a,b,c,d]$ 表示系统。

解 由系统的状态空间模型, 编写如下的 MATLAB 程序:

```
%MATLAB PROGRAM example2-3.m
a = [-0.5620 0.05114;-0.254 0];
b = [0.03247 1.145;0.112541 0];
c = [1 1];d=[0 0];sys=ss(a,b,c,d)
```

a =

```
          x1      x2
x1  -0.562  0.05114
x2  -0.254      0
```

b =

```
          u1      u2
x1  0.03247  1.145
x2  0.1125  0
```

c =

```
          x1  x2
y1  1  1
```

d =

```
          u1  u2
y1  0  0
```

Continuous-time model.

## 2.1.2 控制系统模型转换与化简

一个系统的数学模型的表达式(微分方程模型、传递函数模型、零极点模型、状态空间模型等)之间存在着内在的联系, 虽然它们外在形式不同, 但它们的实质内容是等价的。人们在进行系统分析研究时, 往往根据不同的要求选择不同形式的系统数学模型, 因此研究不同形式的数学模型之间的转换具有重要意义。表 2-1 为 MATLAB 常用模型转换函数。

表 2-1 MATLAB 常用模型转换函数

函数名	函数功能描述	常用格式
ss2tf	状态空间转换为传递函数模型	$[b,a]=ss2tf(A,B,C,D,iu)$
ss2zp	状态空间转换为零极点模型	$[z,p,k]=ss2zp(A,B,C,D,iu)$
tf2ss	传递函数转换为状态空间模型	$[A,B,C,D]=tf2ss(b,a)$
tf2zp	传递函数转换为零极点模型	$[z,p,k]=tf2zp(b,a)$
tf2zpk	传递函数转换为零极点模型	$[z,p,k]=tf2zpk(b,a)$
zp2ss	零极点转换为状态空间模型	$[A,B,C,D]=zp2ss(z,p,k)$



续表

函数名	函数功能描述	常用格式
zp2tf	零极点转换为传递函数模型	$[b,a]=zp2tf(z,p,k)$
chgnits	转换 FRD 模型的 nunits 属性	$sys=chgnits(sys,units)$
reshape	转换 LTI 阵列的形状	$sys=reshape(sys,s1,s2,\dots,sk)$ $sys=reshape(sys,[s1 s2\dots sk])$
residue	提供部分分式展开	$[z,p,k]=residue(b,a)$ $[b,a]=residue(z,p,k)$

### (1) 系统模型向传递函数形式的转换

功能：可以实现将状态空间方程转换为传递函数的形式。

格式： $[\text{num},\text{den}]=\text{ss2tf}(A,B,C,D,iu)$ 或 $[\text{num},\text{den}]=\text{zp2tf}(z,p,k)$

说明：其中  $iu$  用于指定变换所使用的输入量。为了获得传递函数的系统形式，还可以采用下述方式进行，即

$G1=(A,B,C,D)$ ;  $G2=\text{tf}(G1)$ 。

可以证明，由给定的状态方程模型转换为传递函数形式的结果是唯一的。

**【例 2-4】** 已知连续系统  $\Sigma(A,B,C,D)$  的系数矩阵是

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 1 \\ 0 & 0 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad C = [1 \ 1 \ 0], \quad D = 0, \quad \text{求取该系统相应的传递函数模型。}$$

解 应用 MATLAB 的 `ss2tf` 函数可以方便地实现这种转换。

```
%MATLAB PROGRAM example2-4.m
A=[2 0 0;0 4 1;0 0 4];B=[1 0 1]';C=[1 1 0];D=0;
[num,den]=ss2tf(A,B,C,D)
G=tf(num,den)
```

运行结果：

```
num = 0    1.0000   -7.0000   14.0000
den = 1   -10    32   -32
```

Transfer function:

```
      s^2 - 7 s + 14
```

```
-----
      s^3 - 10 s^2 + 32 s - 32
```

### (2) 系统模型向零极点形式的转换

功能：实现将各类系统模型转换为零极点形式模型的函数。

格式： $[z,p,k]=\text{ss2zp}(A,B,C,D,iu)$