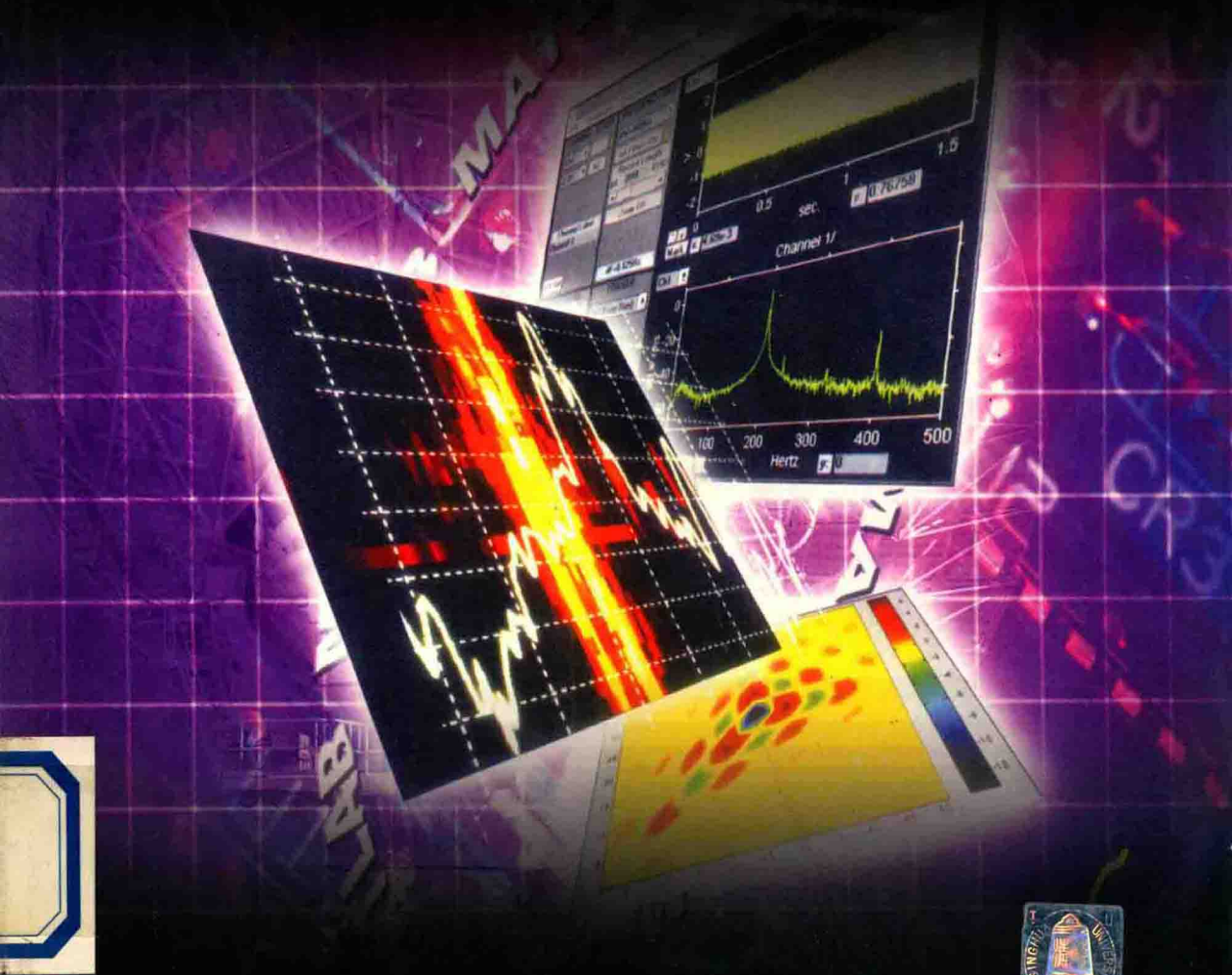


# 控制系统设计与仿真

李宜达 编著

逻辑电路的设计已经进入以软件、语言方式实现电路的时代。VHDL语言具有功能强大，可以灵活运用优势，让你轻松地规划硬件控制电路！



清华大学出版社

内容简介

前言

# 控制系统设计与仿真

李宜达 编著

清华大学出版社

北京

清华大学出版社  
 地址：北京清华大学学研大厦A座  
 邮编：100084  
 电话：(010) 62770175  
 网址：<http://www.tup.tsinghua.edu.cn>  
 0000-1111  
 0000-1111

本书主要介绍控制系统的分析与设计方法，包括时域分析、频域分析、根轨迹法、状态空间法等。同时，本书还介绍了控制系统的仿真方法，包括MATLAB/Simulink等软件的应用。本书可作为高等院校自动化专业及相关专业的教材，也可供从事控制工程工作的工程技术人员参考。

本书共分8章。第1章介绍控制系统的组成及分类；第2章介绍控制系统的时域分析；第3章介绍控制系统的频域分析；第4章介绍根轨迹法；第5章介绍状态空间分析；第6章介绍控制系统的综合设计；第7章介绍控制系统的仿真；第8章介绍控制系统的鲁棒控制。本书可作为高等院校自动化专业及相关专业的教材，也可供从事控制工程工作的工程技术人员参考。

## 内 容 简 介

本书不拘泥于严谨的控制理论,而使用 MATLAB/SIMULINK 来了解控制工程的问题,内容包含两个部分:第 1 部分介绍 SIMULINK 的使用方法,从基本功能到高级功能都配合范例进行了详细介绍,并分析了每一个模块;第 2 部分介绍控制系统的设计、分析与仿真,如时域响应分析、频域响应分析、根轨迹法、状态空间设计法及离散控制系统等。

本书内容简洁实用,循序渐进,是一本适合专科以上理工科学生及工作人员自学的工具书。

本书繁体字版书名为《控制系统设计与模拟》,由台湾全华科技图书股份有限公司出版,版权属李宜达所有。本书简体字中文版由台湾全华科技图书股份有限公司授权清华大学出版社独家出版,仅限于中国大陆地区出版发行。未经本书原版出版者和本书出版者书面许可,任何单位和个人不得以任何形式或任何手段复制或传播本书的部分或全部内容。

北京市版权局著作权合同登记号 图字:01-2004-3953

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933  
本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

### 图书在版编目(CIP)数据

控制系统设计与仿真/李宜达编著.—北京:清华大学出版社,2004.8  
ISBN 7-302-09036-X

I. 控… II. 李… III. ①控制系统—系统设计 ②控制系统—系统仿真 IV. TP271

中国版本图书馆 CIP 数据核字(2004)第 070114 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦  
http://www.tup.com.cn 邮 编:100084  
社 总 机:010-62770175 客 户 服 务:010-62776969

责任编辑:李春明

封面设计:陈刘源

印 刷 者:北京季蜂印刷有限公司

装 订 者:三河市召亮装订有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印 张:23 字 数:544 千字

版 次:2004 年 8 月第 1 版 2004 年 8 月第 1 次印刷

书 号:ISBN 7-302-09036-X/TP·6382

印 数:1~4000

定 价:35.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

# 前 言

MATLAB 是一个功能相当强大的数值分析仿真软件,它在图形影像的处理上,有堪称一流的表现。另外,它还提供了众多不同功能的工具箱(Toolbox),适合应用于不同的工程领域,这更增加了它的完整性。SIMULINK 为后起之秀,俨然成了 Math Works 的明日之星,全依赖其图形化易学易用的特性,用户已不需要记忆很多函数命令,只需根据模块的功能,就能轻易地组合要仿真的模型。

本书包含两部分内容,第 1 部分说明 SIMULINK 的用法,第 2 部分则使用 MATLAB/SIMULINK 阐述控制系统中的一些基本原理,如时域响应分析法、频域响应分析法、根轨迹法、状态空间设计法及离散控制系统等。笔者并非数学专家,所学有限,故本书不以严谨的控制理论为基础,而着重于应用 MATLAB/SIMULINK 来解决控制工程问题。读者若能从本书中获得一些知识,就是笔者最大的愿望。

本书分为两大部分共 14 章,具体如下:

第 1 章说明 MATLAB/SIMULINK 的简要功能,并说明其在控制系统分析仿真上的方便性,并介绍 MATLAB 的基本使用法,作为全书学习的基础。

第 1 部分为 SIMULINK 篇,包括第 2~6 章。

第 2 章说明 SIMULINK 快速使用法,目的是让读者研读完此章后,就能对 SIMULINK 立即上手、使用。列举 3 个简单模型作为范例,介绍 SIMULINK 的使用手法。

第 3 章进一步说明使用 SIMULINK 建构模型的一些基本方法。

第 4 章说明在 SIMULINK 环境中仿真及仿真结果的分析。讨论设置仿真参数值以及如何从 SIMULINK 菜单或 MATLAB 命令视窗中,开始仿真的执行,并且讨论不同迭代演算法。以及讨论 SIMULINK 所提供线性化以及平衡点相关的工具指令。

第 5 章说明如何使用 SIMULINK 提供的 Masking 功能,可以自定义模块(block)或子系统(subsystem),也可以产生新的对话框和新的模块图标(icon)。

第 6 章说明 SIMULINK 所提供模块库(block library)的功能,并对内含的每一个模块都有功能及使用上的说明,有些还附有范例。本章对需要深入了解 SIMULINK 的读者而言是重要的一章。

第 2 部分为控制系统篇,包括第 7~14 章。

第 7 章内容以不同的系统为例,说明如何在 MATLAB/SIMULINK 环境中,建构动态系统的数学模型。

第 8 章说明控制系统的时域响应,并介绍根轨迹分析法。

第 9 章说明频率响应分析法,包含有波德图、奈奎斯特稳定准则、相位边限与增益边限等相关论述。

第 10 章说明如何根据系统规格设计控制器(或称补偿器),包含有 PID 控制器、相位滞后补偿器,相位超前补偿器和相位滞后-超前补偿器等。

第 11 章说明状态空间分析法,内容包含有极点安置设计、观测器设计和线性二次最佳控制器设计等。

第 12 章简要说明离散控制系统, 内容包含有取样频率的选择、时域分析法、频域分析法。

第 13 章说明控制系统工具箱所提供一种用于分析模型的工具——LTI Viewer, 具有图形化使用界面, 方便于线性非时变系统的分析工具。

第 14 章说明控制系统工具箱所提供一种用于设计控制器或补偿器的工具——SISO Design Tool, 同样具有图形化使用界面, 方便简化设计工作。

MATLAB/SIMULINK 在专科以上学校电机、电子、机械、航空、医学工程等工程学方面, 都有很不错的应用。

## 范例说明

本书范例可执行于 MATLAB 4.x、5.x、6.x/Simulink 1.2c、2.x、3.x、4.x 上, 程序内容请参见书中相关介绍, 并按下列步骤进行:

1. 登录 <http://www.wenyuan.com.cn>, 查找本书的相关链接, 下载范例程序。
2. 在计算机硬盘上创建一个目录, 用来存放范例程序。例如创建的目录为 `c:\matexamp`, 则键盘输入 `C:\md matexamp`。

对于 MATLAB Ver. 4.2/Simulink 1.2c 系统, 则执行以下操作:

3. 将范例程序中的 `startup4.m` 复制到 MATLAB 目录下。
4. 将 `c:\matlab` 目录下的 `startup4.m` 更名为 `startup.m`。

对于 MATLAB Ver. 5.x/Simulink 2.x 系统, 则执行以下操作:

5. 将范例程序中的 `startup5.m` 复制到 `MATLAB\toolbox\local` 目录下。
6. 将 `c:\matlab\toolbox\local` 目录下的 `startup5.m` 更名为 `startup.m`。
7. 将范例程序中所有 .m 程序(`startup4.m`、`startup5.m` 除外)复制到硬盘的 `matexamp` 目录下。
8. 重新启动 MATLAB, 便可在 MATLAB 命令窗口(Command Window)下执行范例程序。

对于 MATLAB Ver. 6.x/Simulink 4.x 系统, 则执行以下操作:

9. 直接将所有的范例程序(.m、.mdl)复制到 MATLAB 安装目录\work 下, 即可在 MATLAB 命令窗口执行范例程序。

由于笔者所学有限, 书中难免会有一些错误及不够完善之处, 请读者不吝批评指正。

# 目 录

<b>第 1 章 概论</b> .....	1
1.1 控制系统简介 .....	1
1.2 MATLAB/SIMULINK 简介 .....	2
1.3 系统需求 .....	5
1.4 MATLAB 的基本使用方法.....	5
1.4.1 基本操作 .....	5
1.4.2 多维数组 .....	8
1.4.3 矩阵的运算 .....	10
1.4.4 数组的运算 .....	11
1.4.5 向量和矩阵的操作方法.....	15
1.4.6 矩阵函数 .....	19
1.4.7 字符串 .....	21
1.4.8 多项式运算 .....	23
1.4.9 Script 文件与函数.....	25
1.4.10 程序控制流程 .....	27
1.4.11 输入与输出 .....	30
<b>第 1 部分 SIMULINK</b>	
<b>第 2 章 快速开始</b> .....	33
2.1 如何开始 .....	33
2.2 一个简单的模型.....	34
2.3 快速上手 .....	36
2.4 另一个简单模型.....	37
2.5 PID 控制器模型 .....	41
<b>第 3 章 建构模型</b> .....	45
3.1 建构新模型 .....	45
3.2 编辑已存在的模型.....	45
3.3 选择对象 .....	46
3.3.1 选择单个对象 .....	46
3.3.2 选择多个对象 .....	46
3.3.3 选择模型内的所有对象.....	47
3.4 在不同窗口间复制或移动模块.....	47
3.5 在模型窗口中移动模块.....	48

3.6 在模型窗口中复制模块.....	48
3.7 设置模块的参数.....	48
3.8 删除模块.....	48
3.9 改变模块的放置方向.....	49
3.10 改变模块图标的形状大小.....	49
3.11 修改模块的名称.....	50
3.11.1 修改模块的名称.....	50
3.11.2 改变模块名称的位置.....	50
3.12 输入值与参数值的标量 扩展(Scalar Expansion) .....	51
3.13 在模块间连接线段.....	51
3.14 删除线段.....	53
3.15 移动线段.....	53
3.16 移动顶点.....	54
3.17 分割直线段.....	54
3.18 生成子系统(Subsystem).....	55
<b>第 4 章 仿真与分析</b> .....	57
4.1 简介 .....	57
4.1.1 SIMULINK 如何工作 .....	57
4.1.2 代数回路结构.....	58
4.2 仿真 .....	58
4.2.1 从 SIMULINK 菜单中 执行系统仿真.....	58
4.2.2 从命令窗口执行系统仿真.....	65
4.2.3 指定模块的初始值.....	65
4.2.4 观察输出轨迹.....	67
4.3 离散时间系统.....	69
4.3.1 多采样率系统.....	70
4.3.2 不同采样时间的颜色区别.....	71
4.4 线性化.....	72
4.5 平衡点的决定.....	74
<b>第 5 章 自定义模块</b> .....	75
5.1 简介 .....	75

5.2	Masking 程序概述.....	75	9.2	波德图.....	249
5.3	生成一个 Masked 模块.....	76	9.3	奈奎斯特图.....	252
5.4	另一个例子.....	83	9.4	相对稳定性.....	259
5.5	S 函数.....	86	9.5	标准二阶系统频率响应.....	262
5.6	一个完整的例子.....	89	9.6	尼可士图.....	266
<b>第 6 章</b>	<b>模块解析.....</b>	<b>95</b>	<b>第 10 章</b>	<b>控制器设计.....</b>	<b>268</b>
6.1	Sources 模块库.....	95	10.1	引言.....	268
6.2	Sinks 模块库.....	113	10.2	PID 控制器.....	268
6.3	Continuous 模块库.....	123	10.3	PID 控制器的积分终结.....	273
6.4	Discrete 模块库.....	133	10.4	相位超前补偿器.....	278
6.5	Math 模块库.....	142	10.5	相位滞后补偿器.....	285
6.6	Functions & Tables 模块库.....	159	10.6	相位滞后-超前补偿器.....	291
6.7	Nonlinear 模块库.....	171	<b>第 11 章</b>	<b>状态空间设计法.....</b>	<b>297</b>
6.8	Signals & Systems 模块库.....	184	11.1	引言.....	297
6.9	Subsystems 模块库.....	204	11.2	可控性与可观测性.....	297
	<b>第 2 部分 控制系统</b>		11.2.1	可控性.....	297
<b>第 7 章</b>	<b>动态系统模型的创建.....</b>	<b>208</b>	11.2.2	可观测性.....	298
7.1	引言.....	208	11.3	极点放置设计.....	300
7.1.1	经典控制学的数学模型 表示法——传递函数.....	208	11.4	观测器设计.....	304
7.1.2	现代控制学的数学模型 表示法——动态方程式.....	209	11.5	线性二次最佳控制器设计.....	308
7.1.3	数学模型转换.....	211	<b>第 12 章</b>	<b>离散时间控制系统.....</b>	<b>313</b>
7.2	机械系统.....	216	12.1	引言.....	313
7.3	电路系统.....	226	12.2	连续系统的离散化.....	313
7.4	电机械系统.....	227	12.3	时域分析.....	318
<b>第 8 章</b>	<b>时域响应分析法.....</b>	<b>233</b>	12.4	频域分析.....	324
8.1	引言.....	233	12.5	自动引导车系统.....	327
8.2	时域响应.....	233	12.5.1	路径追踪调节器的设计.....	329
8.2.1	瞬时响应的性能指针.....	234	12.5.2	时变最佳控制.....	330
8.2.2	二阶系统.....	237	12.5.3	计算机仿真.....	332
8.2.3	高阶系统的近似简化.....	242	12.5.4	仿真结果.....	334
8.3	根轨迹分析.....	244	<b>第 13 章</b>	<b>LTI Viewer.....</b>	<b>337</b>
<b>第 9 章</b>	<b>频域响应分析法.....</b>	<b>249</b>	13.1	简介.....	337
9.1	引言.....	249	13.2	直流伺服电机模型.....	337
			13.3	显示多个模型的响应图形.....	343
			13.4	多输入多输出模型响应.....	344
			13.5	Simulink LTI Viewer.....	345

第 14 章 SISO Design Tool.....	348	14.3 增加积分器.....	352
14.1 简介.....	348	14.4 增加超前(领先)网络.....	353
14.2 直流伺服电机模型.....	348	14.5 凹陷滤波器.....	356

## 1.1 控制系统简介

控制系统按受给定或外界的扰动信号, 经检测、放大、运算、控制, 使被控对象(如电动机、水箱、锅炉、机床、飞机等)按照给定的期望运行(如位置、速度、温度、流量、压力、转速等)运行。图 1-1 所示, 典型闭环控制系统由参考输入(Reference)、比较元件(Comparator)、控制器(Controller)、被控对象(Process 或 Plant)、反馈元件(Feedback Element)和反馈通道(Feedback Channel)组成。



图 1-1 典型闭环控制系统组成

按系统是否存在反馈, 又可划分为:

① 开环控制系统(Open-loop Control System)

开环控制系统的控制系统定义为开环控制系统, 如图 1-2 所示。开环控制系统是指系统的输出量对系统的控制作用没有影响(如图 1-2 所示), 开环控制系统中, 系统的输出量对系统的控制作用没有影响。开环控制系统的结构图如图 1-2 所示。

② 闭环控制系统(Closed-loop Control System)

闭环控制系统是指系统的输出量对系统的控制作用有影响。闭环控制系统的结构图如图 1-1 所示。系统接受一个输入信号, 经放大、运算、控制, 使被控对象按照给定的期望运行(如位置、速度、温度、流量、压力、转速等)运行。闭环控制系统的输出量对系统的控制作用有影响。闭环控制系统的结构图如图 1-1 所示。系统接受一个输入信号, 经放大、运算、控制, 使被控对象按照给定的期望运行(如位置、速度、温度、流量、压力、转速等)运行。闭环控制系统的输出量对系统的控制作用有影响。闭环控制系统的结构图如图 1-1 所示。



图 1-2 开环控制系统组成示意图



# 第1章 概 论

## 1.1 控制系统简介

一个控制系统能够接收外来的激励信号，并对这种特定的输入命令(Command)产生调整(Regulation)或追踪(Tracking)的动态行为。控制系统主要由控制组件、受控组件和反馈组件组成(如图 1.1 所示)。受控组件一般称之为受控系统(System)、受控程序(Process)或受按钮装置(Device 或 Plant)，而控制组件则称为控制器(Controller)或补偿器(Compensator)，反馈组件称之为传感器(Transducer)。

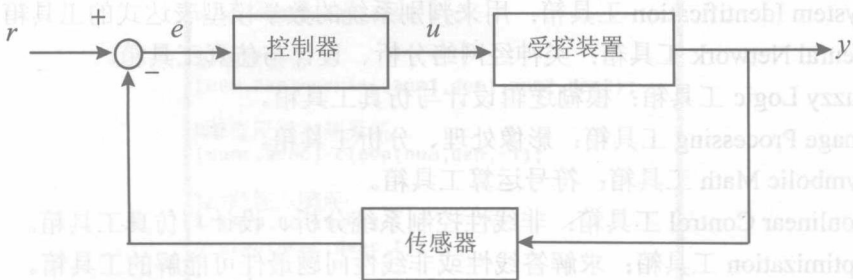


图 1.1 典型控制系统的组成架构图

控制系统根据是否存在反馈，又可划分为：

- 开环控制系统(Open-loop Control System)

没有反馈路径的控制系统定义为开环控制系统，也就是说输出信号不会影响输入信号对系统的操作(如图 1.2 所示)。开环控制系统适用于系统输入与输出间关系已知，且无外来干扰信号的控制系统，它的结构固然简单，但受到外来信号干扰时，系统的精度低。

- 闭环控制系统(Closed-loop Control System)

输出与输入间加有反馈路径的控制系统定义为闭环控制系统，即输出信号会直接影响输入信号对系统的操作(如图 1.1 所示)。系统接受一个输入命令  $r$  后，与经传感器而来的  $y$  输出信号比较后(一般而言是相减)，产生误差信号  $e$ ，控制器接收误差信号后，产生制动信号  $u$  来驱动受控装置以减少系统误差，这样能降低外来噪声(Noise)及干扰(Disturbance)对系统的影响，使输出渐渐地达到设(预)定的值。闭环控制系统因为有反馈组件的存在，使得系统成本提高且更为复杂，但系统的控制精度有很大提高。

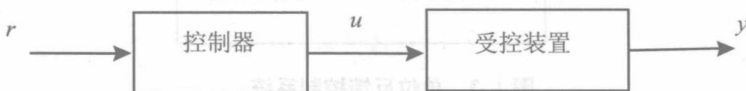


图 1.2 开环控制系统组成框图

## 1.2 MATLAB/SIMULINK 简介

时至今日, MATLAB 软件包在工程领域应用中已经非常普遍,可以说它是在线性及非线性动态系统仿真与分析上不可或缺的工具软件。MATLAB 是 MATrix LABoratory 两单词的缩写,由字意可知,其基本的操作对象是矩阵,最初是为了解决线性代数所衍生的应用问题而发展起来的。而 Laboratory 更强调了 MATLAB 在教育及学术研究领域上所提供的能力。

MATLAB 除了一些内置的函数外,针对一些特殊的应用领域也设计了专门的工具箱(一些.m文件的函数),一些常用的工具箱如下:

- Control System 工具箱: 控制系统分析、设计与仿真工具箱。
- Signal Processing 工具箱: 数字信号分析、处理工具箱。
- Robust Control 工具箱: 鲁棒控制系统分析、设计与仿真工具箱。
- System Identification 工具箱: 用来判别系统的数学模型表达式的工具箱。
- Neural Network 工具箱: 类神经网络分析、设计与仿真工具箱。
- Fuzzy Logic 工具箱: 模糊逻辑设计与仿真工具箱。
- Image Processing 工具箱: 影像处理、分析工具箱。
- Symbolic Math 工具箱: 符号运算工具箱。
- Nonlinear Control 工具箱: 非线性控制系统分析、设计与仿真工具箱。
- Optimization 工具箱: 求解线性或非线性问题最佳可能解的工具箱。
- Wavelet 工具箱: 功能强大的信号影像分析、压缩工具箱。
- Partial Differential 工具箱: 求解偏微分方程(PDE)问题工具箱。
- 此外,还有 QFT、LMI、Model Predictive、Statistics、Spline、 $\mu$ -Analysis and Synthesis、Frequency Domain System Identification 和 Financial 等工具箱。

SIMULINK 为 MATLAB 提供了一个工具箱(Toolbox)。在最近几年,后来崛起的 SIMULINK 已成为学术领域及工业领域在构建、仿真与分析动态系统上使用最为广泛的软件包,它支持线性及非线性系统,能创建连续时间、离散时间或两者混合的系统模型。系统也能够是多采样频率的(Multirate),也就是不同的系统能够以不同的采样频率组合起来。

以前仿真一个系统必须先把系统用微分或差分方程式表示出来,再使用一种语言(例如 C 语言)来程序化系统,这种方式既费时又费力。在建构模型方面, SIMULINK 提供图形化用户界面(Graphical User Interface, GUI),只要在窗口中使用鼠标拖动(Click-and-Drag)就能轻易地像组合框图那样创建出模型,如同用笔在纸上设计模型一样容易。以如图 1.3 所示的典型单位反馈控制系统为例,求取单位阶跃输入的输出响应。



图 1.3 单位反馈控制系统

其中:

$$G_c(s) = \frac{10s+1}{s}, \quad G_p(s) = \frac{400}{s(s+50)}$$

如果用 C 语言仿真此系统, 首先求取闭环传递函数(Transfer Function), 再利用 z 转换求取其离散域表达式, 再以差分方程式表示输出, 最后利用递归算法求出每一个采样时间的输出值。读者若有兴趣不妨自己做做看, 保证能有一些成就感。也可以考虑以 MATLAB 软件来仿真此系统, 在【记事本】窗口中输入如图 1.4 所示的程序, 并以文件名 m1\_1.m 存储, 在 MATLAB 命令窗口下, 输入程序 m1\_1(如图 1.5 所示), 便可得到如图 1.6 所示的阶跃输入响应曲线图。



```

01 - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
% m1_1.m
% 求阶跃输入响应曲线
num1=[0 1];den1=[1 0];
num2=400;den2=[1 50 0];

% 串联组合
[num,den]=series(num1,den1,num2,den2);

%单位反馈控制系统
[numc,denc]=cloop(num,den,-1);

%阶跃输入响应
t=[0:0.001:0.5];
y=step(numc,denc,t);

%绘图
plot(t,y)
xlabel('时间(秒)')
title('阶跃输入响应曲线图')
grid
  
```

图 1.4 MATLAB 的仿真程序

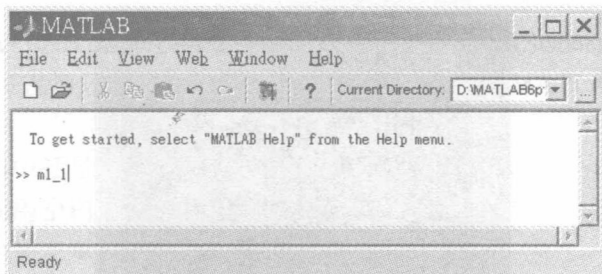


图 1.5 启动 MATLAB 的仿真程序

最后若以 SIMULINK 仿真此系统, 则可在 SIMULINK 窗口构建出如图 1.7 所示的框图(文件名: m1\_2.mdl), 启动仿真此框图, 从 Scope 窗口中可以看出输出的仿真结果(如图 1.8 所示)。

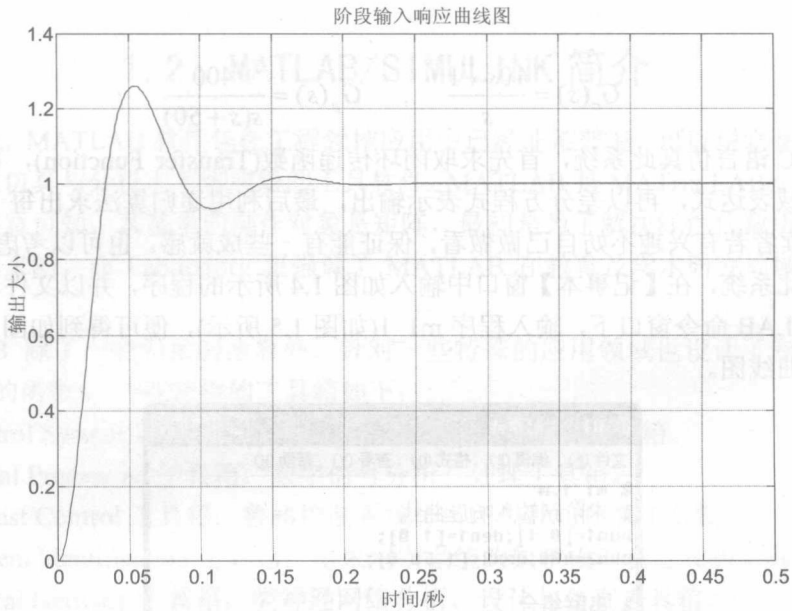


图 1.6 MATLAB 仿真程序的阶跃输入响应图

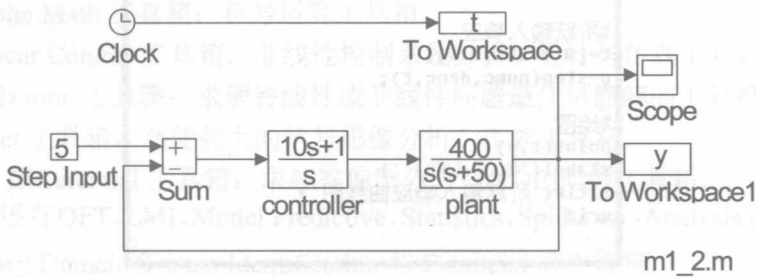


图 1.7 在 SIMULINK 中建立仿真模型

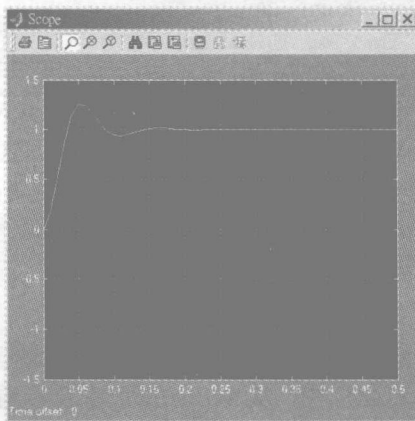


图 1.8 SIMULINK 仿真的结果

比较这 3 种方法，可以看出第 2 种和第 3 种方法比第 1 种方法要方便得多，第 3 种方

法在建构模型方面又比第2种方法简单得多,事实上在 SIMULINK 中所仿真出的一些变量值可以返回到 MATLAB 窗口中,利用控制工具箱(Control Toolbox)或其他工具箱所提供的功能作进一步分析,所以 SIMULINK 在仿真控制系统方面能提供更快速且方便的工具与方法。

SIMULINK 在使用上分为两个方面:模型的定义(创建)和模型的分析。下面首先学习如何定义一个模型(Model)及如何调出以前创建好的模型,再启动仿真函数去分析此模型,以便得到仿真结果。事实上这两种程序是交互使用的,利用它们不断地修改模型,改变模型框图的参数值,再不断地仿真,以得到控制系统所需的性能规格。

建构(定义)好模型后,可以在 SIMULINK 窗口上的菜单(Menu)中去仿真所建构的模型,或者是在 MATLAB 的命令窗口下,输入命令去仿真所建构的系统。SIMULINK 提供的内置分析工具包括各种仿真演算法则,其中有系统线性化工具(见 linmod 命令)及寻找平衡(稳态)点工具(见 trim 命令)。正在仿真的过程可以用 Scope、XY Graph 或其他框图来观察仿真结果,仿真结果也可以用变量存储到 MATLAB 命令窗口中,然后再进行进一步地分析。

## 1.3 系统需求

SIMULINK 与 MATLAB 具有相同的系统环境需求(可参考 MATLAB 用户指南)。电子技术发展迅速,使得计算机不论在速度还是在功能上,与以前相比都已不可同日而语。为了使 SIMULINK 执行起来更有效率,可以选择现在主流的计算机系统(例如 Pentium 4),操作系统可使用微软 Windows 95、98、2000、Windows Millenium 或 Windows NT, Compaq Alpha, Linus, SGI 和 Sun Solaris 等。

## 1.4 MATLAB 的基本使用方法

### 1.4.1 基本操作

矩阵数据是 MATLAB 最基本和最重要的数据形式,标量(Scalar)可视为  $1 \times 1$  阶的矩阵数据,向量(Vector)可视为只含有 1 列或 1 行的矩阵数据。在 MATLAB 中矩阵数据的构成要素有:

- 矩阵数据由中括号([])包含。
- 矩阵元素间用空白或逗号(,)来分隔。
- 使用分号(;)来表示矩阵行(Row)数据的结束。

例如在 MATLAB 命令窗口中,输入:

```
» A=[1 2 3;4 5 6;7 8 9]
```

得到结果为:

```
A =  
 1     2     3  
 4     5     6  
 7     8     9
```

MATLAB 会存储矩阵  $A$  的数据,以后程序中如果用到变量  $A$ ,即表示使用上面列出的矩阵数据值,直到关闭 MATLAB 软件为止。另外矩阵数据可直接以矩阵形式输入,例如输入:

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

矩阵数据也可由.m 文件读入,例如某一名称为 gen\_mtr.m 的文件,其内容包含 3 行文本(Text)数据,如:

```
A=[1 2 3
    4 5 6
```

或

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

在 MATLAB 命令窗口中,输入:

```
» gen_mtr
```

则会产生矩阵数据  $A$ 。

矩阵内的元素可为任意 MATLAB 表达式(Expression),例如:

```
» x=[-1.5 sqrt(3^2+4^2) (1+2+3)/4*5]
```

得到结果为:

```
x = -1.5000    5.0000    7.5000
```

其中矩阵  $x$  内第 1 个元素为  $x(1)$ ,第 2 个元素为  $x(2)$ ,依此类推,例如输入:

```
» x(1)
ans =
    -1.5000
» x(3)
ans =
     7.5000
```

可以直接指定矩阵内的元素值,方法为:

```
» x(5)=x(2)^2;
» x
x =
    -1.5000    5.0000    7.5000         0    25.0000
```

注意,未定义的  $x(4)$  将会被自动指定为 0。

另外,矩阵数据间也可以合并,例如输入:

```
» p=[10 11 12];
» B=[A;p]
B =
     1     2     3
     4     5     6
     7     8     9
    10    11    12
```

矩阵元素也允许是复数值，复数以  $i(i = \sqrt{-1})$  或  $j$  表示虚部，例如：

```
» z=1+sqrt(3)*i;
```

也可以用极坐标表示，例如：

```
» w=2*exp(i*pi/3)
w =
    1.0 + 1.7321i
```

另外，也可以直接输入复数矩阵，方法为：

```
» C=[1 2;3 4]+i*[5 6;7 8]
C =
    1.0000 + 5.0000i    2.0000 + 6.0000i
    3.0000 + 7.0000i    4.0000 + 8.0000i
```

或

```
» C=[1+5i 2+6i;3+7i 4+8i];
```

在本章后面的小节中还会对矩阵运算做进一步说明。

## 1. MATLAB 表达式

MATLAB 是一种表达式语言，整个 MATLAB 程序可看成是对表达式的解释及计算，经常用到的形式为：

```
variable=expression;
```

或简单地只有：

```
expression;
```

如果 expression 后紧接有分号(;), 则表示会抑制表达式的结果不显示出来。例如：

```
» 99999/9      % 99999/9 为表达式
ans =
    11111
» m=1*2*3*4*5    % m 为变量, 1*2*3*4*5 为表达式
m =
    120
```

即变量  $m$  存储值为 120。如果表达式太长以至于在一行写不下时，或为了程序浏览方便需要分行编写表达式时，可在一行的结尾加上“...”，再按 carriage return(换行)键，即可在下一行中输入，例如：

```
» n=1.1-2.2+3.3-4.4+5.5...
    -6.6+7.7-8.8+9.9
n =
    5.5000
```

MATLAB 的变量是有大小写区分的，也就是说， $A$  和  $a$  是不同的变量名称，而 MATLAB 提供的函数(Function)名称是小写的， $\text{inv}(A)$  是求矩阵  $A$  的逆矩阵，而  $\text{INV}(A)$  则不是 MATLAB 定义的函数名称。

MATLAB 的变量或函数名称必须以字母开始，再加上任何数目的字母或数字，其间可用下划线分隔，但只有前 19 个字母和数字有效，例如 moon\_9、m3n2p4q5、g5\_1 等。

至此所列举的变量都存储在 MATLAB 的工作区(Workspace)中, 为了列举出工作区中的变量名称, 可以在 MATLAB 命令窗口中, 输入:

```
» who
Your variables are:
A          C          m          p          x
B          ans         n          w          z
```

如果要更详细地显示变量的大小及类型, 可输入:

```
» whos
Name      Size      Bytes Class
A         3x3       72  double array
B         4x3       96  double array
C         2x2       64  double array (complex)
ans       1x1        8  double array
m         1x1        8  double array
n         1x1        8  double array
p         1x3       24  double array
w         1x1       16  double array (complex)
x         1x5       40  double array
z         1x1       16  double array (complex)
Grand total is 38 elements using 352 bytes
```

## 2. 输出格式

可以使用 `format` 命令更改输出格式, 如果矩阵数据内的元素都为整数(integer)类型, 则矩阵数据的输出格式即为整数类型, 但若有一个元素不为整数, 则将以实数类型输出, 输出格式默认值为 `short`, 其格式为小数点后的 4 位数是有效数, 例如:

```
» x=[34/3 0.000012345]
x =
    11.3333    0.0000
» format short e
x =
    1.1333e+001    1.2345e-005
» format long
x =
    11.333333333333334    0.00001234500000
» format long e
x =
    1.1333333333333334e+001    1.234500000000000e-005
» format bank
x =
    11.33    0.00
» format hex
x =
    4026aaaaaaaaaaaab    3ee9e3abel16fc70d
» format +
x =
    ++
```

### 1.4.2 多维数组

MATLAB 5.0 版在数据结构方面提供多维数组的数据表示法, 如 `zeros`、`ones`、`rand` 和 `randn` 等都能产生超过两个输入自变量的数据, 例如:

```
» ones(2,2,2)
```



```
ans(:,:,1) =
    1     1
    1     1
ans(:,:,2) =
    1     1
    1     1
```

上式命令产生 2-by-2-by-2 的数组，共有  $2 \times 2 \times 2 = 8$  个元素。

下面这个例子是将 `magic` 函数所产生的  $3 \times 3$  阶矩阵，以 `perms` 函数所产生的排列来改变其列向量位置。式子：

```
p=perms(1:4)
```

将产生  $4! = 24$  种的  $[1\ 2\ 3\ 4]$  排列方式，例如：

```
>> p=perms(1:3)
p =
     3     2     1
     2     3     1
     3     1     2
     1     3     2
     2     1     3
     1     2     3
```

而 `magic(n)` 函数将产生  $n \times n$  阶矩阵，其行向量、列向量及对角线上元素的和会相同，例如：

```
>> x=magic(4)
x =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

执行下列程序，其中所得 `M` 变量以三维数组类型存储 24 个不同列向量位置的 `magic` 数。`size(M)` 可得  $3 \times 3 \times 24$ 。

```
p=perms(1:4);
x=magic(3);
M=zeros(3,3,24);
For k=1:24
    M(:,:,k)=x(p(k,:),:);
End
```

`Cell` 数组也是多维数组，但其元素是由数组构成的，例如：

```
>> x=cells(4,1);
>> for n=1:4
    x{n}=magic(n);
>> end
>> x
x =
    [
        [2x2 double]
        [3x3 double]
        [4x4 double]
    ]
>> x{1}
ans =
    1
```