

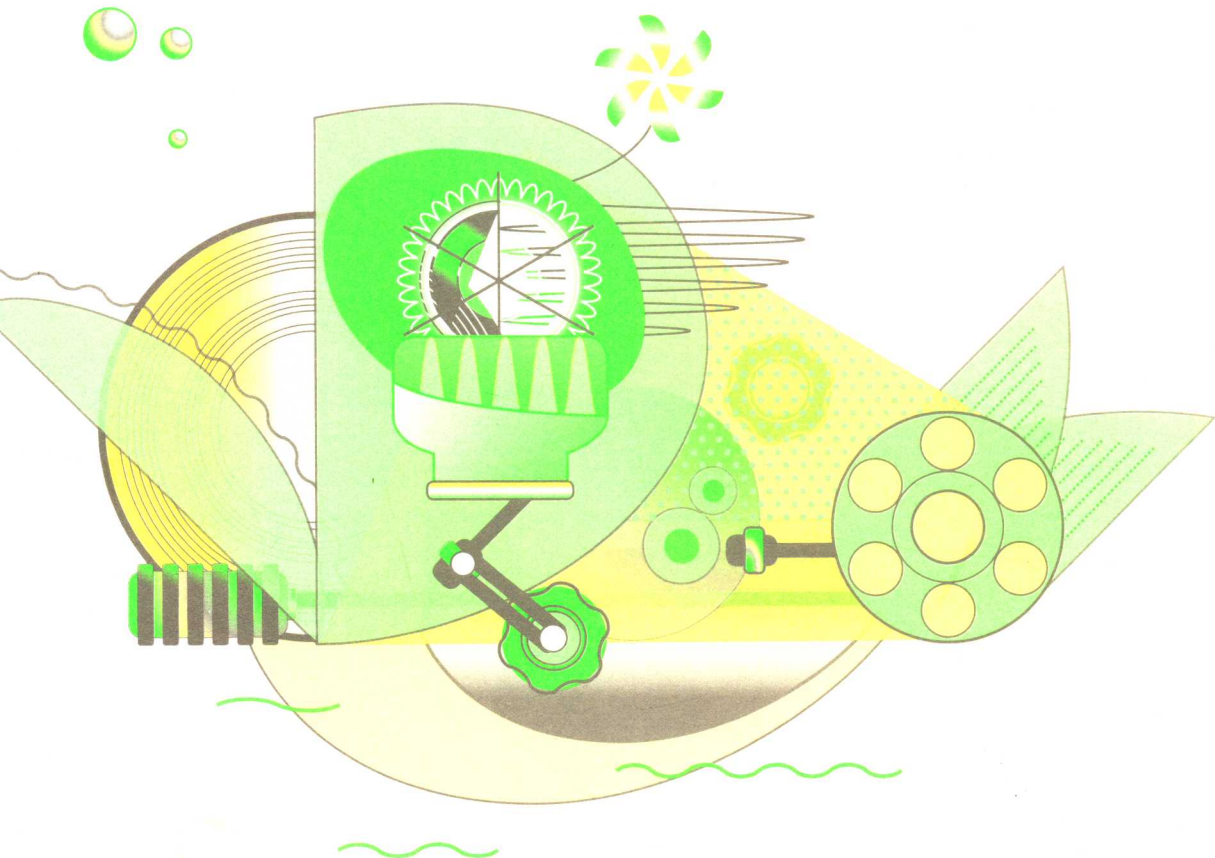


全国教育科学“十一五”  
规划课题研究成果

# 控制系统的 MATLAB 仿真与设计

第二版

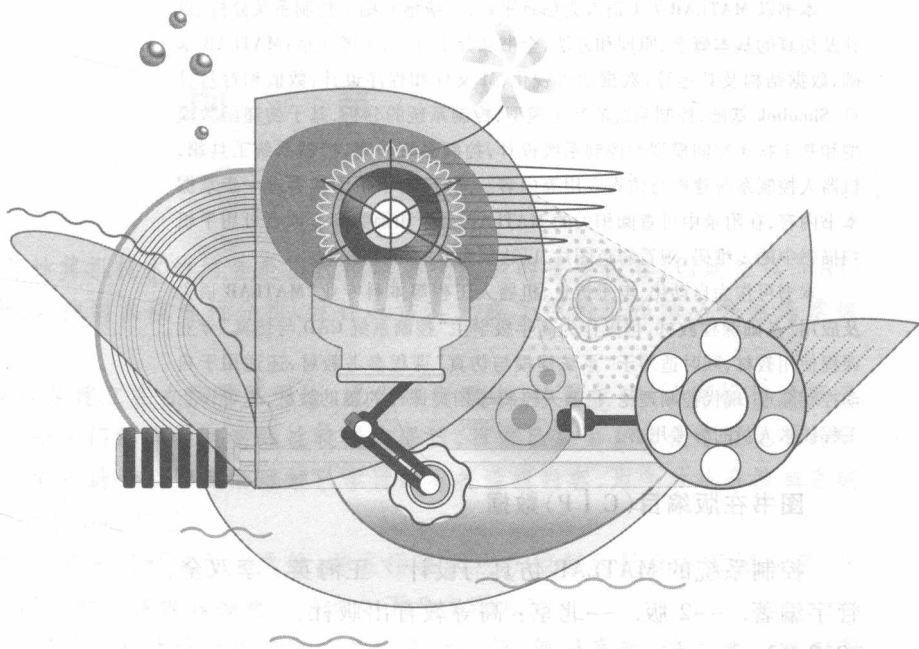
王海英 李双全 管宇 编著



高等教育出版社



全国教育科学“十一五”  
规划课题研究成果



# 控制系统的 MATLAB 仿真与设计

第二版

王海英 李双全 管宇 编著

高等教育出版社·北京

## 内容简介

本书以 MATLAB 7.1 语言为仿真平台,系统地介绍了控制系统分析、设计及仿真的基本概念、原理和方法,全书共分十五章,主要包括:MATLAB 基础,数据结构及其运算,数据的可视化,M 文件和程序设计,数值和符号计算,Simulink 基础,控制系统的数学模型,控制系统的分析,基于传递函数模型和基于状态空间模型的控制系统设计,控制系统仿真,控制系统工具箱,机器人控制系统建模与仿真应用等内容。同时,为了帮助读者进一步掌握本书内容,在附录中可查阅相关的 MATLAB 命令和函数库。读者可用手机扫描书中的二维码,观看彩色图片,以达到更好的仿真效果。

本书可作为自动化、电子信息、机器人工程等本科专业“MATLAB 语言及应用”基础课程教材,也可作为高年级学生“控制系统 CAD 与仿真”专业课程使用教材,同时适用于“系统建模与仿真”课程参考教材,还适用于自动控制原理、现代控制理论、机器人控制等相关课程的辅助教材,也可作为工程技术人员的参考用书。

## 图书在版编目(CIP)数据

控制系统的 MATLAB 仿真与设计 / 王海英, 李双全, 管宇编著. --2 版. --北京: 高等教育出版社, 2019.12

ISBN 978-7-04-052863-3

I. ①控… II. ①王… ②李… ③管… III. ①算法语言-应用-自动控制系统-系统仿真 IV. ①TP273

中国版本图书馆 CIP 数据核字(2019)第 225677 号

策划编辑 韩颖 责任编辑 韩颖 封面设计 张申申 版式设计 杨树  
插图绘制 于博 责任校对 吕红颖 责任印制 赵义民

出版发行	高等教育出版社	网 址	<a href="http://www.hep.edu.cn">http://www.hep.edu.cn</a>
社 址	北京市西城区德外大街 4 号		<a href="http://www.hep.com.cn">http://www.hep.com.cn</a>
邮政编码	100120	网上订购	<a href="http://www.hepmall.com.cn">http://www.hepmall.com.cn</a>
印 刷	北京中科印刷有限公司		<a href="http://www.hepmall.com">http://www.hepmall.com</a>
开 本	787mm×1092mm 1/16		<a href="http://www.hepmall.cn">http://www.hepmall.cn</a>
印 张	28.25	版 次	2009 年 2 月第 1 版
字 数	640 千字		2019 年 12 月第 2 版
购书热线	010-58581118	印 次	2019 年 12 月第 1 次印刷
咨询电话	400-810-0598	定 价	54.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究

物料号 52863-00

# 前 言

控制原理、计算方法与计算机技术的结合是当代控制系统发展的重要内容,因此在以 MATLAB 为代表的软件平台上,对控制系统进行分析、设计与仿真就成了控制工程师必须熟练掌握的重要知识与技能。

控制系统的仿真与设计是建立在古典控制理论、现代控制理论、计算方法、计算机技术等多方面知识上的综合性学科,是一门综合性与实践性较强的课程,目前已成为自动化专业重要的研究分支,灵活地掌握与运用它有助于我们深刻理解已学过的有关课程内容,为今后从事控制系统领域的研究与开发工作提供坚实的基础。

本书是编者对教学、实验与科研工作的总结,并在借鉴国内外控制和机器人领域专家、学者研究成果的基础上编写而成的。在内容编排上具有如下几方面的特点:

1. 将 MATLAB 的使用方法和程序设计以精练的篇幅进行介绍,使得直接、快速地了解 and 掌握 MATLAB 软件平台成为可能,由浅到深,重点针对控制系统涉及的 MATLAB 功能和函数进行阐述;

2. 在已学习的古典与现代控制理论的基础上,介绍在 MATLAB 软件平台上对系统进行分析 and 设计的方法;

3. 重点介绍目前控制系统仿真技术中状态空间法和 Simulink 仿真的主要思想,达到学以致用的效果;

4. 所有的例题、习题都经过精心选择,书中所有用 MATLAB 描述的程序都经过严格的上机调试,保证所写程序的可用性,是本书最重要的特色;

5. 精选控制系统及机器人的经典案例,从控制系统及机器人理论建模开始,逐步地介绍了 Simulink 建模方法、控制器设计方法以及系统的 MATLAB 综合仿真方法,能对控制系统及机器人建模与仿真有更全面、完整的掌握。

本书不仅适于自动化、电气工程、机器人工程等本科专业学生学习,同时也可供控制及机器人领域工程师及相关专业技术人员参考。

本书授课按 40 ~ 60 学时编写,共分 15 章,哈尔滨理工大学李双全教授编写了第 1、2、3、4 章,管宇编写了第 5、6、7 章,杨柳编写了第 8、9 章,齐丽华编写了第 10 章,王海英教授编写了第 14、15 章及附录,黑龙江工程学院高辉编写了第 11、12、13 章,全书由王海英教授统稿定稿。编写过程中还得到孟庆松、孙晓波、袁丽英等教师的大力支持与帮助,合肥学院王俊教授及哈尔滨工业大学杨学博教授对本书进行了认真的审阅,并提出了宝贵的、有建设性的改进意见,同时参

考文献所列资料为本书的编写提供大量素材,本书的出版还得到高等教育出版社和美国 The MathWorks 公司图书计划的支持,谨此一同表示衷心感谢。

由于编者水平有限,错误或不当之处,敬请广大读者批评指正。选用本书作为教材使用的老师可与高等教育出版社联系索要电子课件。作者 E-mail: why.69@163.com。

编者

2019年5月于哈尔滨理工大学

# 第1版前言

控制原理、计算方法与计算机技术的结合是当代控制系统发展的重要内容,因此在以 MATLAB 为代表的软件平台上,对控制系统进行分析、设计与仿真就成了控制工程师必须熟练掌握的重要知识与技能。

控制系统的仿真与设计是建立在古典控制理论、现代控制理论、计算方法、计算机技术等多方面知识上的综合性学科,是一门综合性与实践性较强的课程,目前已成为自动化学科重要的研究分支,灵活地掌握与运用它有助于我们深刻理解已学过的有关课程内容,为今后从事控制系统领域的研究与开发工作提供坚实的基础。

本书是编者对教学、实验与科研工作的总结,并在借鉴国内外控制领域专家、学者研究成果的基础上编写而成的。在内容编排上具有如下几方面的特点:

1. 将 MATLAB 的使用方法和程序设计以精练的篇幅进行介绍,使得直接、快速地了解 and 掌握 MATLAB 软件平台成为可能,并由浅到深,重点针对控制系统涉及的 MATLAB 功能和函数进行介绍。

2. 在已学习的古典与现代控制理论的基础上,介绍在 MATLAB 软件平台上对系统进行分析 and 设计的方法。

3. 重点介绍目前控制系统仿真技术中状态空间法和 Simulink 仿真的主要思想,达到学以致用效果。

4. 所有的例题、习题都经过精心选择,书中所有的用 MATLAB 描述的程序都经过严格的上机调试,保证所写程序的可用性,这是本书最重要的特色。

5. 精选控制系统的经典案例,从控制系统理论建模开始,逐步地介绍了 Simulink 建模方法、控制器设计方法以及系统的 MATLAB 综合仿真方法,能对控制系统建模与仿真有更全面、完整的掌握。

本书不仅适于自动化、电气工程、系统工程等本科专业学生学习,同时也可供控制领域工程师及相关专业技术人员参考。

本书授课按 40~60 学时编写,由王海英、袁丽英、吴勃编著,袁丽英编写了第一、二、三、四章,张颖编写了第五章,吴勃编写了第六、七、十三章,马静编写了第八章,姜滨玲编写了第九章,齐鑫编写了第十章,第十一、十二、十四章及附录由王海英编写,全书由王海英统稿定稿。孙晓波、李双全认真细致地对本书进行了审阅,编写过程中还得到孟庆松、景旭等的大力支持与帮助,合肥学院的王俊教授精心地审阅了全书,并提出了宝贵的、有建设性的改进意见,同时参考文献

所列资料为本书的编写提供了大量素材,本书的出版还得到高等教育出版社和美国 The Math Works 公司图书计划的支持,谨此一并表示衷心感谢。

由于编者水平有限,错误或不当之处,敬请广大读者批评指正。编者可向选用本书作为教材的老师提供电子课件。E-mail:why.69@163.com。

编者

2008 年 11 月

# 目 录

第一章 MATLAB 基础	1	第二章 数据结构及其运算	17
1.1 MATLAB 简介	1	2.1 数据类型	17
1.1.1 MATLAB 的发展历程	1	2.2 一维数组	18
1.1.2 MATLAB 平台的组成	2	2.2.1 一维数组的创建	18
1.1.3 MATLAB 语言的特点	3	2.2.2 一维数组的访问	19
1.2 MATLAB 的安装和使用	4	2.3 二维数组	20
1.2.1 MATLAB 的安装	4	2.3.1 二维数组的创建	20
1.2.2 MATLAB 操作界面	5	2.3.2 二维数组的访问	20
1.3 MATLAB 的工作空间	8	2.4 多维数组	23
1.3.1 工作空间	8	2.4.1 多维数组的创建	23
1.3.2 内存变量的查阅和删除	8	2.4.2 多维数组的访问	24
1.3.3 数组编辑器	9	2.5 数组操作	25
1.3.4 数据文件的操作	10	2.5.1 标准数组的生成	25
1.4 MATLAB 的帮助系统	11	2.5.2 数组操作	27
1.4.1 普通文本帮助	11	2.6 数组运算与矩阵运算	30
1.4.2 超文本帮助	12	2.6.1 数组运算	30
1.5 MATLAB 平台的系统设置	13	2.6.2 矩阵运算	31
1.5.1 路径设置	13	2.6.3 特殊的运算结果	33
1.5.2 个性选项	14	2.7 多项式	33
1.6 MATLAB 编程基础	14	2.7.1 多项式的创建	34
1.6.1 变量	15	2.7.2 多项式的运算	34
1.6.2 基本数据类型	15	2.8 关系运算、逻辑运算和运算符	36
1.6.3 基本语句	15	2.8.1 逻辑值	36
1.6.4 关于程序编辑	16	2.8.2 关系运算符	36
小结	16	2.8.3 逻辑运算符	37
习题	16	2.8.4 逻辑函数	37

2.8.5 运算符及其优先级 .....	39	3.5.2 三维图形的精细控制 .....	60
2.9 字符串数组 .....	40	3.5.3 图像 .....	61
2.9.1 字符与字符串 .....	40	小结 .....	61
2.9.2 字符串数组的访问 .....	41	习题 .....	61
2.9.3 字符串转换函数 .....	41	<b>第四章 M 文件和程序设计</b> .....	63
2.9.4 字符串操作函数 .....	41	4.1 程序控制语句 .....	63
2.10 细胞数组 .....	42	4.1.1 分支控制 .....	63
2.10.1 细胞数组的数据结构 .....	42	4.1.2 循环控制 .....	65
2.10.2 细胞数组的访问 .....	42	4.1.3 异常检测 .....	66
2.11 结构体数组 .....	44	4.1.4 人机交互控制 .....	67
2.11.1 结构体数组的数据结构 .....	44	4.1.5 其他程序流控制语句 .....	67
2.11.2 结构体数据的访问 .....	44	4.2 命令文件和函数文件 .....	68
2.11.3 结构体数组的创建 .....	44	4.2.1 命令文件 .....	68
小结 .....	46	4.2.2 函数文件 .....	69
习题 .....	46	4.2.3 函数文件的组成 .....	69
<b>第三章 数据的可视化</b> .....	48	4.2.4 函数的输入和输出变量 .....	70
3.1 数据可视化基础 .....	48	4.2.5 有关工作空间 .....	73
3.1.1 图形窗口 .....	48	4.2.6 子函数和私有函数 .....	74
3.1.2 离散数据的准备 .....	49	4.3 字符串的求值 .....	75
3.1.3 数据可视化的基本步骤 .....	50	4.3.1 字符串表达式计算 .....	75
3.2 二维图形 .....	50	4.3.2 字符串函数计算 .....	76
3.2.1 基本二维图形 .....	51	4.3.3 内联函数 .....	76
3.2.2 绘图属性控制 .....	52	4.4 程序性能优化 .....	79
3.2.3 绘图窗口的属性控制 .....	53	4.4.1 源代码级的优化 .....	79
3.2.4 图形标注 .....	54	4.4.2 程序加速器 .....	80
3.2.5 图形交互命令 .....	56	4.5 面向对象编程 .....	80
3.3 三维图形 .....	56	4.5.1 类和对象 .....	81
3.3.1 基本三维图形 .....	56	4.5.2 属性和方法 .....	81
3.3.2 三维网线 and 曲面 .....	57	4.5.3 构造函数 .....	81
3.4 常用函数的绘图 .....	58	4.5.4 重载 .....	81
3.4.1 一元函数的绘图 .....	58	4.5.5 继承 .....	82
3.4.2 二元函数的绘图 .....	59	4.5.6 创建新类 .....	82
3.5 数据可视化的其他方面 .....	60	小结 .....	82
3.5.1 特殊图形 .....	60	习题 .....	82

第五章 数值计算	84	6.2.2 符号表达式分解、展开与化简	109
5.1 线性代数	84	6.2.3 符号表达式的置换操作	110
5.1.1 LU 分解	84	6.2.4 符号函数的反函数	111
5.1.2 特征值和特征向量	85	6.2.5 符号函数的复合函数	111
5.1.3 奇异值分解	86	6.3 符号微积分	112
5.1.4 线性方程组的解	86	6.3.1 符号微分和雅可比矩阵	112
5.2 函数分析	88	6.3.2 函数极限	113
5.2.1 函数的零点	88	6.3.3 符号积分	113
5.2.2 函数的极值点	90	6.3.4 符号序列求和	114
5.2.3 数值微分	91	6.4 符号方程求解	115
5.2.4 数值积分	91	6.4.1 符号代数方程组的解	115
5.3 数据拟合	93	6.4.2 符号微分方程	116
5.3.1 多项式拟合	93	6.5 积分变换	117
5.3.2 最小二乘拟合	94	6.5.1 傅里叶变换及其反变换	117
5.4 插值和样条	96	6.5.2 拉普拉斯变换及其反变换	118
5.4.1 插值	96	6.5.3 $z$ 变换及其反变换	118
5.4.2 样条	97	6.6 Maple 的应用	119
5.5 常微分方程的数值解	99	6.6.1 经典特殊函数的调用	119
5.5.1 ODE 文件的编写	99	6.6.2 Maple 函数的调用	120
5.5.2 solver 解算指令	99	6.6.3 Maple 工具的帮助系统	120
小结	101	小结	121
习题	102	习题	121
第六章 符号计算	103	第七章 Simulink 基础	122
6.1 符号对象的创建	103	7.1 Simulink 简介	122
6.1.1 符号对象的生成	103	7.2 Simulink 的基本操作	123
6.1.2 符号常量	103	7.2.1 Simulink 的运行	123
6.1.3 符号变量与符号表达式	104	7.2.2 常用的标准模块	123
6.1.4 符号数学函数	105	7.2.3 模块的操作	124
6.1.5 符号数值计算的精度	106	7.3 系统仿真及参数设置	127
6.1.6 符号对象与其他数据类型之间的转换	107	7.3.1 算法设置	127
6.2 符号对象的代数运算	108	7.3.2 工作空间设置	128
6.2.1 符号对象的运算	108	7.4 Simulink 模块库	130
		7.4.1 信号源模块库	130

7.4.2	连续系统模块库	131	8.5	数学模型的分析函数	171
7.4.3	离散系统模块库	132	8.5.1	模型分析函数	171
7.4.4	数学运算模块库	133	8.5.2	模型分析举例	172
7.4.5	输出模块库	134	小结		175
7.4.6	非线性系统模块库	135	习题		175
7.5	Simulink 子系统	135	<b>第九章 控制系统的分析</b>		177
7.5.1	Simulink 子系统简介	135	9.1	线性系统的时域响应	177
7.5.2	Simulink 高级子系统应用	138	9.1.1	单位阶跃响应	177
7.5.3	封装子系统	138	9.1.2	理想单位脉冲响应	182
7.6	S 函数的建立	146	9.1.3	零输入响应	183
7.6.1	S 函数的调用格式	146	9.1.4	任意输入响应	184
7.6.2	S 函数的模板格式	147	9.2	线性系统的根轨迹	186
小结		150	9.2.1	根轨迹绘制	186
习题		150	9.2.2	计算根轨迹增益	186
<b>第八章 控制系统的数学模型</b>		152	9.3	线性系统的频域响应	188
8.1	线性定常系统的数学模型	152	9.3.1	奈奎斯特(Nyquist)图	188
8.1.1	传递函数模型	152	9.3.2	伯德(Bode)图	189
8.1.2	零极点模型	153	9.3.3	幅值和相角裕度计算	190
8.1.3	状态空间模型	155	9.3.4	尼柯尔斯(Nichols)图	194
8.1.4	频率响应数据模型	156	9.4	线性系统的稳定性分析	194
8.1.5	离散系统的数学模型	156	9.4.1	直接判定方法	194
8.1.6	随机生成稳定的控制系统	157	9.4.2	间接判定方法	195
8.2	LTI 模型的属性	157	9.5	离散系统的分析	197
8.2.1	LTI 模型的属性描述	157	小结		199
8.2.2	访问 LTI 模型的属性	161	习题		199
8.3	数学模型之间的转换	162	<b>第十章 基于传递函数模型的控制 系统设计</b>		200
8.3.1	LTI 对象之间的转换	162	10.1	概述	200
8.3.2	LTI 对象属性之间的转换	162	10.2	根轨迹法	201
8.3.3	连续系统和离散系统之间的 转换	163	10.2.1	串联超前校正	201
8.4	系统模型的运算	165	10.2.2	串联滞后校正	207
8.4.1	子系统	165	10.3	伯德图法	211
8.4.2	数学运算符	165	10.3.1	串联超前校正	211
8.4.3	系统组合	166			

10.3.2 串联滞后校正 .....	215	小结 .....	278
10.3.3 串联超前-滞后校正 .....	221	习题 .....	278
10.3.4 反馈校正 .....	223		
10.4 PID 控制 .....	228	<b>第十二章 控制系统仿真</b> .....	280
10.4.1 比例、积分、微分控制作用的 分析 .....	228	12.1 状态空间法仿真 .....	280
10.4.2 齐格勒-尼柯尔斯(Ziegler- Nichols)整定法则 .....	235	12.1.1 四阶龙格-库塔法 .....	280
小结 .....	240	12.1.2 闭环模型的建立 .....	281
习题 .....	241	12.2 非线性环节的仿真 .....	284
<b>第十一章 基于状态空间模型的控制 系统设计</b> .....	242	12.2.1 饱和非线性环节仿真 .....	284
11.1 概述 .....	242	12.2.2 死区非线性环节仿真 .....	285
11.2 极点配置 .....	243	12.2.3 间隙非线性环节仿真 .....	285
11.2.1 单输入系统的极点配置 .....	244	12.3 离散系统仿真 .....	287
11.2.2 多输入系统的极点配置 .....	244	12.3.1 差分方程法 .....	287
11.2.3 用极点配置设计调节系统 .....	246	12.3.2 z 变换法 .....	289
11.2.4 用极点配置设计伺服系统 .....	248	12.4 典型环节的 Simulink 仿真 .....	291
11.3 线性二次型最优控制 .....	255	12.4.1 带延迟环节系统的 Simulink 仿真 .....	291
11.3.1 无限时间 LQ 状态调节问题 .....	255	12.4.2 含非线性环节系统的 Simulink 仿真 .....	292
11.3.2 无限时间 LQ 输出调节问题 .....	258	12.4.3 离散系统的 Simulink 仿真 .....	295
11.3.3 最优跟踪问题 .....	259	小结 .....	296
11.4 解耦控制 .....	261	习题 .....	297
11.4.1 输入-输出动态解耦 .....	262	<b>第十三章 控制系统工具箱</b> .....	299
11.4.2 输入-输出静态解耦 .....	263	13.1 线性时不变系统浏览器—— LTI Viewer .....	299
11.5 状态观测器设计 .....	264	13.1.1 LTI Viewer 简介 .....	299
11.5.1 全维状态观测器设计 .....	265	13.1.2 LTI Viewer 命令菜单及窗口 设置 .....	299
11.5.2 降维状态观测器设计 .....	267	13.1.3 LTI Viewer 对象模型 .....	302
11.6 包含状态观测器的状态反馈 控制系统 .....	270	13.1.4 LTI Viewer 属性设置 .....	302
11.6.1 基于全维状态观测器的 控制器 .....	271	13.2 单输入、单输出系统设计 工具——SISO Design Tool .....	305
11.6.2 基于全维状态观测器的 调节器 .....	273	13.2.1 SISO Design Tool 简介 .....	305

13.2.2 SISO Design Tool 命令菜单		
简介 .....	306	
小结 .....	317	
习题 .....	317	
<b>第十四章 控制系统建模与仿真的应用</b> .....	318	
14.1 数字 PID 控制器的仿真 .....	318	
14.2 一级倒立摆系统的控制与仿真 .....	320	
14.2.1 系统建模 .....	320	
14.2.2 模型仿真 .....	321	
14.2.3 PID 控制器设计及仿真 .....	325	
14.2.4 LQR 控制器设计及仿真 .....	328	
14.3 二级倒立摆系统的控制与仿真 .....	332	
14.3.1 系统建模 .....	332	
14.3.2 LQR 控制器设计及仿真 .....	336	
14.3.3 数字再设计 .....	339	
14.4 双闭环调速系统的设计与仿真 .....	348	
14.4.1 系统建模 .....	348	
14.4.2 控制器设计 .....	349	
14.4.3 系统仿真 .....	349	
14.5 过程控制系统的仿真 .....	352	
14.5.1 系统建模 .....	353	
14.5.2 控制器参数整定 .....	354	
14.5.3 系统仿真 .....	358	
小结 .....	359	
习题 .....	359	
<b>第十五章 机器人控制系统建模与仿真应用</b> .....	361	
15.1 机器人工具箱 .....	361	
15.1.1 rvctools 工具箱安装方法 .....	361	
15.1.2 rvctools 工具箱常用函数 .....	363	
15.2 移动机器人控制与仿真 .....	367	
15.2.1 移动机器人运动学建模与仿真 .....	367	
15.2.2 移动机器人跟踪控制与仿真 .....	369	
15.2.3 移动机器人姿态控制与仿真 .....	373	
15.2.4 移动机器人位置估计与仿真 .....	376	
15.3 四旋翼飞行器控制与仿真 .....	380	
15.3.1 四旋翼飞行器动力学建模 .....	381	
15.3.2 四旋翼飞行器控制与仿真 .....	382	
15.4 机械手控制与仿真 .....	391	
15.4.1 机械手运动学建模与仿真 .....	391	
15.4.2 机械手独立关节控制与仿真 .....	394	
15.4.3 机械手柔性控制与仿真 .....	409	
小结 .....	413	
习题 .....	413	
<b>附录 A MATLAB 常用命令</b> .....	415	
<b>附录 B 控制系统工具箱</b> .....	421	
<b>附录 C Simulink 仿真系统模块库</b> .....	424	
<b>附录 D 部分源程序清单</b> .....	427	
<b>参考文献</b> .....	437	

# 第一章 MATLAB 基础

## 1.1 MATLAB 简介

### 1.1.1 MATLAB 的发展历程

MATLAB 名称由 Matrix 和 Laboratory 两词的前三个字母组合而成,顾名思义其主要功能是矩阵运算,其首创者是 Cleve Moler 教授。1980 年前后,当时的新墨西哥大学计算机系主任 Moler 教授在讲授线性代数课程时,发现用其他高级语言编程极为不便,便构思并开发了 MATLAB,这一软件利用了当时数值线性代数领域最高水平的 EISPACK 和 LINPACK 两大软件包中可靠的子程序,用 Fortran 语言编写了一套集命令翻译、科学计算于一身的交互式软件系统。该语言无需像 C 语言和 Fortran 语言那样编写源程序、编译、连接,最终形成可执行文件。早期的 MATLAB 是用 Fortran 语言编写的,只能做矩阵运算,绘图也只能用极其原始的方法,内部函数也只提供了几十个。1984 年, Cleve Moler 和 John Little 等人成立了 Math Works 公司,正式将 MATLAB 推向市场。从此, MATLAB 的内核采用 C 语言编写,除了原有的数值计算能力外,还增加了丰富多彩的图形图像处理、多媒体功能、符号运算和与其他流行软件的接口功能,这使得 MATLAB 的功能越来越强大。MATLAB 以其良好的开放性和运行的可靠性,使很多领域封闭式的数值计算软件包纷纷被淘汰,而改在 MATLAB 平台上重建。20 世纪 90 年代, MATLAB 已经成为国际控制界公认的标准计算软件,在国际上 30 多个数学类科技应用软件中, MATLAB 在数值计算方面独占鳌头。

Math Works 公司于 1993 年推出了具有划时代意义的基于 Windows 平台的 MATLAB 4.0 版,使之应用范围越来越广。1994 年推出的 4.2 版扩充了 4.0 版的功能,尤其在图形界面设计方面更提供了新的方法。1997 年推出了 MATLAB 5.0 版,定义了更多的数据结构,如单元数据、数据结构体、多维矩阵、对象与类等,使其成为一种更方便编程的语言。1999 年初推出的 MATLAB 5.3 版在很多方面又进一步改进了 MATLAB 语言的功能。2000 年 10 月底推出了其全新的 MATLAB 6.0 正式版,在核心数值算法、界面设计、外部接口、应用界面等诸多方面有了极大的改进。2002 年夏推出的 MATLAB 6.5 版,其最大特点是采用了 JIT 加速器,使 MATLAB 的运算速度进一步加快。2005 年 9 月发布的 MATLAB 7.1 完整版,提供了 MATLAB、Simulink 的升级以及其他最新的 75 个模块的升级,并具有用于数据分析、大规模建模、固定点开发、编码等的新特征。

2007 年秋发布了 MATLAB 7.4 版,该版本对以前版本的很多模块做了升级改进,同时增加了 MATLAB Builder for .net,扩展了 MATLAB Compiler 的功能,使网络程序员可以通过 C#、VB、.net 等语言使用 MATLAB。

虽然 MATLAB 是计算数学专家倡导并开发的,但其普及和发展离不开自动控制领域学者的贡献,因为在 MATLAB 的发展进程中,许多有代表性的成就与控制界的要求是分不开的,其大多数工具箱也都是有关控制方面的。MATLAB 具有强大的数学运算能力、方便实用的绘图功能及语言的高度集成性,它在其他科学与工程领域的应用也越来越广,并且有着更广阔的应用前景和无穷无尽的潜能。MATLAB 是一个十分有效的工具,能解决在教学与研究中遇到的问题,可以将使用者从繁琐的底层编程中解放出来,把有限的宝贵时间更多地花在解决问题中,大大提高了工作效率。目前, MATLAB 已经成为国际上最流行的科学与工程计算的软件工具,它不仅是一个“矩阵实验室”和一张“手写式计算纸”,而且已经成为一种具有广泛应用前景的全新的计算机高级编程语言,有人称它为“第四代”计算机语言,它在国内外高校和研究部门正扮演着重要的角色。MATLAB 语言的功能也会越来越强大,不断适应新的要求,提出新的解决方法。MATLAB 将长期在科学运算、自动控制与科学绘图领域处于重要地位。

### 1.1.2 MATLAB 平台的组成

MATLAB 不仅仅是一门编程语言,还是一个集成的软件平台,它包含以下几个主要部分。

1. MATLAB 语言  
MATLAB 语言是一种高级编程语言,它提供了多种数据类型、丰富的运算符和程序控制语句供用户使用。用户可以根据需求,按照 MATLAB 语言的约定,编程完成特定的工作。
2. MATLAB 集成工作环境  
MATLAB 集成工作环境包括程序编辑器、变量查看器、系统仿真器和帮助系统等。用户在集成工作环境中可以完成程序的编辑、运行和调试,输出和打印程序的运行结果。
3. MATLAB 图形系统  
用 MATLAB 的句柄图形,可以实现二维、三维数据的可视化、图像处理,也可以完全或局部修改图形窗口,还可以方便地设计图形界面。
4. MATLAB 数学函数库  
MATLAB 提供了丰富的数值计算函数库,既包括常用的数学函数,又包含了各个专业领域独有的数值计算实现,用户通过简单的函数调用就可以完成复杂的数学计算任务。
5. Simulink 交互式仿真环境  
通过交互式的仿真环境 Simulink,用户可以采用图形化的数学模型,完成对各类系统的模型建立和系统仿真,仿真结果也能够以直观的图形方式显示。Simulink 可以接受用户的键盘、鼠标输入,也可以通过程序语句来实现数据交换,应用方便灵活。
6. MATLAB 编译器  
通过编译器,用户可以将用 MATLAB 语言编写的程序编译成脱离 MATLAB 环境的 C 语言源

代码、动态链接库或者可以独立运行的可执行文件。

### 7. 应用程序接口 API

API 是 MATLAB 的应用程序接口,它提供了 MATLAB 和 C、Fortran、VB、VC 等多种语言之间的接口程序库,使用户可以在这些语言的程序里调用 MATLAB 程序。

### 8. MATLAB 工具箱

MATLAB 包含了各种可选的工具箱。工具箱是由各个领域的高水平专家编写的,所以用户不必编写该领域的基础程序就可以直接进行更高层次的研究。例如:控制领域可以使用的工具箱有 Control System(控制工具箱)、System Identification(系统辨识工具箱)、Robust Control(鲁棒控制工具箱)、Optimization(最优化工具箱)等。

### 9. Notebook 工具

Notebook 能够让用户在 Word 环境中使用 MATLAB 的各种资源,为用户营造融文字处理、科学计算、工程设计于一体的完美的工作环境。用 Notebook 制作的 M-Book 文档不仅拥有 Word 的全部字处理功能,而且具备 MATLAB 的数学运算能力和计算结果可视化能力。

## 1.1.3 MATLAB 语言的特点

MATLAB 被称为第四代计算机语言,利用其丰富的函数资源,可使编程人员从繁琐的代码中解脱出来。MATLAB 用更直观、符合人们思维习惯的代码,代替了 C 语言的冗长代码,给用户带来的是最直观、最简洁的程序开发环境。MATLAB 语言的主要特点是:

(1) 语言简洁紧凑,语法限制不严格,程序设计自由度大,使用方便灵活。

在 MATLAB 里可以不用先定义或声明变量就可以使用它们,MATLAB 程序的书写格式自由,数据的输入、输出语句简洁,很短的代码就可以完成其他语言要经过大量代码才能完成的复杂工作。

例如: $A = [1, 2, 3; 4, 5, 6; 7, 8, 9]$  一条语句实现了对  $3 \times 3$  矩阵的输入。

(2) 数值算法稳定可靠,库函数十分丰富。

MATLAB 的一个最大特点是强大的数值计算能力,它提供了许多调用十分方便的数学计算的函数,可以随意使用而不必考虑数值的稳定性。

```
例如:e = eig(A)           % 求矩阵 A 的特征值
      [L,U] = lu(A)        % 求矩阵 A 的 LU 分解
      polyder(b)           % 求多项式 b 的微分
```

(3) 运算符丰富。

MATLAB 是用 C 语言编写的,所以 MATLAB 提供了和 C 语言几乎一样多的丰富的运算符,而且还重载了一些运算符,并给它们赋予了新的含义。

```
例如:C = A * B           % 矩阵的乘法
      B = C'              % 求矩阵 C 的共轭复转置
      x = A \ B           % 求 Ax = B 的最小二乘解,即矩阵的左除
```

(4) MATLAB 既具有结构化的控制语句(if、for、while),又支持面向对象的程序设计。

(5) 程序的可移植性好。

MATLAB 程序几乎不用修改就可以移植到其他机型和操作系统中运行。

(6) MATLAB 的图形功能强大,支持数据的可视化操作,方便地显示程序的运行结果。

(7) 源程序的开发性、系统的可扩充能力强。

除了内部函数外,所有的 MATLAB 核心文件和工具箱文件都提供了 MATLAB 源文件,用户可通过对源文件的修改生成自己所需要的工具箱。

(8) MATLAB 是解释执行语言。

MATLAB 程序不用编译生成可执行文件就可以运行,程序是解释执行的。解释执行的程序执行速度较慢,效率比 C 语言等高级语言要低,而且无法脱离 MATLAB 环境运行,这是 MATLAB 的缺点。但是 MATLAB 的编程效率远远高于一般的高级语言,这使我们可以把大量的时间花费在对控制系统的算法研究上,而不是浪费在大量的基础代码上,这是 MATLAB 能够被广泛应用于科学计算和系统仿真的主要原因。

## 1.2 MATLAB 的安装和使用

### 1.2.1 MATLAB 的安装

MATLAB 能够运行于多种机型 (IBM-PC、Macintosh、Unix 工作站) 和操作系统 (Windows XP、Windows 2000、Linux), 本节仅介绍其在 PC 机上的安装和使用。要在 PC 机上流畅运行 MATLAB, PC 机要具有较好的硬件配置, 推荐的配置是: CPU 主频大于 2.0GHz, 内存大于 1GB, 安装空间大于 3GB。下面基于 MATLAB 7.1 版介绍其安装过程。

将 MATLAB 安装光盘插入光驱后, 会自动启动“安装向导”。如果没有自动启动“安装向导”, 用户需要运行安装光盘中的 setup.exe 应用程序来启动安装向导, 图 1.2.1 所示为 MATLAB 安装组件选择界面。安装过程中, 用户按照向导的提示进行操作即可, 其中比较重要的输入和选

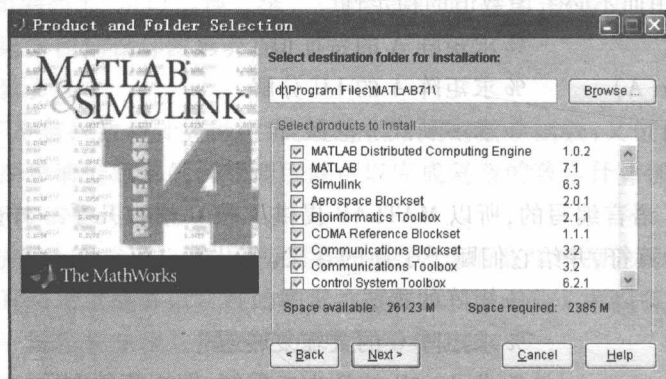


图 1.2.1 MATLAB 安装组件选择界面