



普通高等教育电气信息类规划教材



免费电子教案下载

www.cmpedu.com



自动控制原理的 MATLAB仿真与实践

刘超 高双 编著



机械工业出版社
CHINA MACHINE PRESS



普通高等教育电气信息类规划教材

自动控制原理的 MATLAB 仿真与实践

刘 超 高 双 编著



机械工业出版社

本书以分析自动控制原理的核心内容为主线，由浅入深地介绍基于 MATLAB 语言编程和基于 Simulink 模块编程的仿真系统搭建与分析方法。第 2~8 章介绍基于 MATLAB 的自控系统仿真模型搭建与分析方法；第 9 章介绍基于 Simulink 的自控系统仿真模型搭建与分析方法。本书注重介绍 MATLAB 函数命令格式、具体功能、一般特点、内在联系和使用方法与技巧，并通过大量实例进行详细解读、过程分析与归纳说明，以便于读者自学和上机实验。每章附有习题和实验练习，可满足相关专业不同层次的教学与实践需求。

本书可作为具备一定 MATLAB 与自动控制理论基础的学生的实验配套教材，也可作为工程技术人员从事控制系统仿真研究与设计的参考书。

本书配有电子教案，需要的教师可登录 www.cmpedu.com 免费注册，审核通过后下载，或联系编辑索取（QQ: 2966938356，电话: 010-88379739）。

图书在版编目（CIP）数据

自动控制原理的 MATLAB 仿真与实践 / 刘超，高双编著. —北京：机械工业出版社，2015.3

普通高等教育电气信息类规划教材

ISBN 978-7-111-49516-1

I. ①自… II. ①刘… ②高… III. ①自动控制系统—系统仿真—Matlab 软件—高等学校—教材 IV. ①TP273

中国版本图书馆 CIP 数据核字（2015）第 043015 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：和庆娣 责任校对：张艳霞

责任印制：李 洋

北京宝昌彩色印刷有限公司印刷

2015 年 4 月第 1 版·第 1 次印刷

184mm × 260mm · 18 印张 · 446 千字

0001—3000 册

标准书号：ISBN 978-7-111-49516-1

定价：39.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

网络服务

服务咨询热线：(010) 88379833

机工官网：www.cmpbook.com

读者购书热线：(010) 88379649

机工官博：weibo.com/cmp1952

教育服务网：www.cmpedu.com

封面防伪标均为盗版

金书网：www.golden-book.com

前 言

随着科学技术应用的综合性、复杂性和广泛性的不断提升，控制理论及其相关技术应用的研究成本与实施风险也越来越高。在教学方面，也要求教学实验设施能与时俱进、不断更新，以确保教学实践水平能紧跟时代不断提高。在实验设施更新成本高、投入周期长，但使用周期短的情况下，借助计算机进行辅助研究与虚拟实验的方法来降低成本、减小风险、提高效率，已成为当今教学、科研与工程应用中普遍采用的预研方法之一。在众多的计算机仿真软件中，MATLAB 集数学计算、图形绘制及系统仿真于一体，涉及领域广、运算功能强，已成为教学、学术与工程界公认和广泛采用的仿真软件之一。目前，许多涉及数学计算与建模分析和工程应用的教材中，正在不断引入基于 MATLAB 应用的仿真内容。

MATLAB 属于高级语言，可分为语句编程和模块（或结构）编程。前者属于数学运算类编程，基础性强、涉及面广、应用灵活；后者属于专业应用类编程，专业性强、建模直观、效率高。两者都可以达到同样的目的，只是在编程路线和表现形式上不同。在语句编程中，语言格式可以分为一般性函数命令格式（常用数学函数语言）和专业函数命令格式（专业性函数语言）。模块编程是由模块选择、模块参数设置及模块间信号连接 3 个主要部分组成的，相对独立地集成在 MATLAB 软件中的名为 Simulink 的环境中，所以模块编程通常称为 Simulink 仿真。模块编程后的仿真结果可以作为数据源提供给语句编程和仿真时使用。

由于 MATLAB 的特点，如果只是将 MATLAB 简单地引入教学，会造成专业内容的逻辑性与 MATLAB 语言的功能性之间的矛盾。如自动控制理论包括经典控制理论和现代控制理论，所涉及的系统又分为连续系统和离散系统。而许多 MATLAB 函数命令不仅适用于经典控制也适用于现代控制，不仅适合连续系统也适合离散系统，其差异只是在于函数命令格式中的参数设置上。另外，不同理论的分析方法之间还存在着相互联系（如经典与现代，连续与离散之间的模型转换与响应）。所以在初学时容易产生概念理解与使用方法上的混淆。多年的教学实践表明，MATLAB 的学习必须立足于围绕课程内容，实行渐进式学习和实践来掌握 MATLAB，才能学以致用，取得良好效果。

本书编写的主要特点体现在：以自动控制原理的教学内容为主线，按照教学章节由浅入深地介绍 MATLAB 的基本仿真方法和使用技巧；以表格方式注重介绍函数命令格式的一般性和特定功能性；在给出与知识点相关联的 MATLAB 函数命令格式及其功能说明的基础上，以例题及其多种实现方法来阐述函数命令格式的具体使用方法与技巧；全书力求既体现专业教学内容的逻辑性，又体现 MATLAB 辅助应用的灵活性，既强调函数命令的通用性，更注重函数命令格式的具体化和实用性。书中第 2~8 章主要介绍基于 MATLAB 的自控系统仿真模型

搭建与分析方法，并通过仿真实例加以详细说明；第9章与前几章相对应，简要介绍 Simulink 模块编程、仿真实验与分析的基本方法，以利于对两种编程方法的对应理解和灵活掌握。全书内容体现由浅入深、先易后难的渐进式讲解方法。其中的基本仿真方法可作为与控制理论课程配套的实验内容，复杂仿真方法可作为控制理论工程应用的扩展部分。每章后附有对应的习题和实验练习，以适应相关专业和不同层次的教学需求。

本书第1~8章由刘超执笔完成并负责全书的编辑，第9章由高双执笔完成并负责全书的校对。在稿件整理上得到孙银魁、郭策、程阔、王瑞东、京晋东、刘啸、郭子阳、任超、张瑞津的帮助，在此一并表示感谢！

由于作者水平有限，在编写过程中难免存在疏漏之处，诚请专家学者和广大读者给予批评指正。

编 者

目 录

前言

第 1 章	MATLAB 仿真简介	1
1.1	MATLAB 软件及其自动控制原理仿真	1
1.2	基于 MATLAB 的编程	2
1.2.1	基于 MATLAB 的语言编程	2
1.2.2	基于 MATLAB 的 Simulink 模块编程	9
第 2 章	线性连续系统模型的生成、转换与简化	12
2.1	连续系统模型的生成	12
2.1.1	生成模型的常用函数命令	12
2.1.2	有理多项式分式传递函数模型的建立	12
2.1.3	零极点传递函数模型的建立	15
2.1.4	二阶系统模型	16
2.1.5	其他模型的生成	18
2.2	系统模型的转换	19
2.2.1	常用模型转换的函数命令	19
2.2.2	多项式传递函数模型与零极点增益模型之间的转换	20
2.3	模型的变换与简化	21
2.3.1	简单模型结构的变换与简化	21
2.3.2	复杂模型结构的变换与简化	33
2.4	习题与实验	38
2.4.1	习题	38
2.4.2	实验	41
第 3 章	线性系统的时域分析	44
3.1	时域分析的拉普拉斯变换法	44
3.1.1	连续时间函数的拉普拉斯变换	44
3.1.2	时域函数的拉氏反变换法	46
3.1.3	时域函数的部分分式展开法	48
3.2	时域分析的函数命令	54
3.3	习题与实验	60
3.3.1	习题	60
3.3.2	实验	63
第 4 章	线性系统的根轨迹分析	65
4.1	根轨迹分析的基本思想	65
4.2	根轨迹分析的函数命令	66
4.3	根轨迹分析法	69

4.4	增加零、极点对系统性能的影响	73
4.4.1	增加零点的影响	73
4.4.2	增加极点的影响	74
4.5	根轨迹法设计系统	75
4.6	由开环根轨迹判断闭环系统的稳定性	77
4.7	习题与实验	79
4.7.1	习题	79
4.7.2	实验	80
第 5 章	频率特性分析	81
5.1	频域分析的函数命令	81
5.2	频率特性绘制	82
5.2.1	Nyquist 曲线和 Bode 图	82
5.2.2	闭环频率特性的 Nichols (尼科尔斯) 图	87
5.3	频率特性分析	90
5.3.1	稳定性分析	90
5.3.2	系统性能指标的估算	97
5.3.3	系统的频域设计	101
5.4	习题与实验	105
5.4.1	习题	105
5.4.2	实验	107
第 6 章	线性系统的校正	108
6.1	线性系统校正概述	108
6.2	串联校正	109
6.2.1	串联超前校正	109
6.2.2	串联滞后校正	112
6.2.3	串联滞后-超前校正	114
6.3	并联校正	116
6.3.1	并联校正的一般方法	117
6.3.2	综合法反馈校正	118
6.4	前馈校正	120
6.5	复合校正	122
6.6	线性连续系统的 PID 校正	123
6.6.1	PID 校正的一般结构及其物理实现	124
6.6.2	PID 控制器的作用	125
6.6.3	PID 控制器的工程最优设计	128
6.6.4	PID 参数整定方法	134
6.7	习题与实验	138
6.7.1	习题	138
6.7.2	实验	141

第 7 章	离散系统分析与设计	144
7.1	离散系统建模	144
7.1.1	离散系统的模型表示	144
7.1.2	离散系统传递函数模型的建立	145
7.2	离散系统模型的转换	146
7.2.1	利用专用函数命令转换	146
7.2.2	利用 Z 变换实现离散序列模型转换	147
7.3	离散系统分析	150
7.3.1	离散系统响应	151
7.3.2	离散系统稳定性判断	154
7.3.3	采样周期与开环增益对稳定性的影响	156
7.3.4	采样器和采样保持器对动态性能的影响	157
7.4	离散系统设计	159
7.4.1	离散系统的一般设计方法	159
7.4.2	离散系统的 PID 设计方法	160
7.4.3	数字控制器的实现	161
7.5	习题与实验	164
7.5.1	习题	164
7.5.2	实验	166
第 8 章	线性系统状态空间分析	168
8.1	状态空间模型的建立与转换	169
8.1.1	状态空间模型的生成	169
8.1.2	状态空间模型的转换与化简	175
8.2	状态模型的可控性、可观性及规范性	181
8.3	状态模型的连接	185
8.4	状态空间系统的动态分析	186
8.4.1	状态空间系统的时域分析	187
8.4.2	状态空间系统的频域分析	195
8.5	状态空间系统设计	197
8.5.1	状态反馈与极点配置设计	197
8.5.2	输出反馈与极点配置设计	202
8.5.3	状态观测器设计	203
8.5.4	降维状态观测器及状态反馈设计	206
8.6	线性二次型最优控制器设计	209
8.7	状态空间系统的稳定性分析	212
8.8	习题与实验	215
8.8.1	习题	215
8.8.2	实验	217

第 9 章 基于 Simulink 的系统建模与仿真分析	222
9.1 Simulink 的基本操作.....	222
9.1.1 Simulink 的启动及模块库的打开.....	222
9.1.2 Simulink 模型编辑窗口的启动.....	223
9.1.3 Simulink 仿真模型的编辑与保存.....	224
9.1.4 系统仿真运行.....	226
9.2 Simulink 模块库及其分类.....	227
9.2.1 输入模块.....	227
9.2.2 输出模块.....	229
9.2.3 功能运算模块.....	231
9.3 基于 Simulink 的时域特性分析.....	235
9.3.1 一阶系统的时间响应.....	235
9.3.2 二阶系统的时间响应.....	240
9.3.3 高阶系统的时间响应.....	244
9.3.4 稳定性与稳态误差.....	247
9.3.5 Simulink 在时域分析中的综合应用.....	249
9.4 PID 控制器设计.....	256
9.4.1 比例 (P) 控制.....	257
9.4.2 比例积分 (PI) 控制.....	257
9.4.3 比例微分 (PD) 控制.....	258
9.4.4 比例积分微分 (PID) 控制.....	258
9.4.5 PID 控制器参数的整定.....	261
9.5 离散系统的 Simulink 建模与分析.....	265
9.5.1 离散系统的 Simulink 建模方法.....	265
9.5.2 离散系统的 Simulink 分析.....	268
9.6 状态空间的 Simulink 建模与分析.....	269
9.6.1 状态空间的 Simulink 建模方法.....	269
9.6.2 状态空间的 Simulink 分析.....	272
9.7 习题与实验.....	275
9.7.1 习题.....	275
9.7.2 实验.....	276
参考文献	279

第 1 章 MATLAB 仿真简介

本章简要介绍 MATLAB 软件的特点，以及自动控制原理 MATLAB 仿真的意义；介绍 MATLAB 编程环境和编程方法的一般概念及要点，其中包括 MATLAB 语言编程和 MATLAB/ Simulink 模块编程环境的建立、基本操作与编程方法；以表格形式给出常用符号和函数命令格式及其功能说明。

1.1 MATLAB 软件及其自动控制原理仿真

“MATLAB”一词源于 Matrix Laboratory（矩阵实验室），其含义是用矩阵方法处理问题。MATLAB 软件是一套面向科学计算和系统建模与仿真分析的可视化软件。其主要特点是集数学计算与可视化功能于一体，并提供了几乎所有常用数学运算的函数命令和可用于不同领域的专业工具箱，以满足不同专业的特殊要求，如控制系统工具箱（Control System）、信号处理工具箱（Signal Processing）、图像处理工具箱（Image Processing）、通信工具箱（Communication）、金融（Financial）、统计（Statistics）和优化设计（Optimization）等。MATLAB 具有很高的内联性、可靠性和容错能力；独特的动态系统仿真模块（Simulink）提供了面向控制理论与控制系统的数学建模、系统仿真与分析设计等方面的功能，使用者置身于虚拟实验环境中，易于对复杂的数学和系统模型进行直观的仿真试验和分析处理。

MATLAB 软件自 20 世纪 80 年代初推出以来，其不断升级的软件内涵为各科学领域的发展需求提供了有力的后续支持。MATLAB 在学科上体现了完备性，在学术上体现了严谨性，在应用方面体现了实用性，使之成为后来者易于借助的“巨人肩膀上的工具”，使那些致力于学术研究与工程应用的非计算机专业人士从烦琐的底层编译和复杂的数值计算所需的编程中解脱出来，节省由编程、调试和验证所耗费的时间和精力，降低编程风险，提高工作效率和结果的可信度，因此 MATLAB 更适合科学计算分析与工程应用仿真。强大的内部指令和扩展功能为各个学科领域的研究与应用提供良好的编程基础和技术支持，已成为学术界和工程界普遍认可并广泛使用的科学计算与系统仿真语言之一。

“自动控制原理”属于自动化及机电一体化等理工科专业的专业基础课程，具有理论性强、过程抽象、数学运算复杂、实验室建立与实践成本高等特点。随着计算机技术、软件水平的不断提高与普及，为其复杂运算及虚拟实验提供了可能。目前 MATLAB 已成为该课程大学实践教学的基本内容，也成为在校硕士和博士研究生进行学术和应用研究以及在高端领域进行科学实践的一种有效工具。教学实践表明，由于 MATLAB 能够直观、快速地建立系统模型，并且能够灵活地改变系统的结构和参数，便于系统的动态性能和稳态性能分析，从而达到对系统的优化设计。学习掌握 MATLAB 不仅有利于学生对所学知识进行实践，也利于自身专业技术的可持续性发展与提高。利用 MATLAB 对“自动控制原理”课程进行实践学习，不仅可以加深学生对课程内容的理解，调动其学习积极性，又有利于学生对问题的进一步探索和思考，进而提高学生的分析能力和创新能力，也为以后专业课程的学习和工程应

用实践起到积极的促进和帮助作用。

1.2 基于 MATLAB 的编程

MATLAB 软件主要包括 MATLAB 语言编程和 Simulink 模块编程两大部分。在 MATLAB 环境下进行控制系统的仿真实验，可以通过两个渠道实现：

1) 基于 MATLAB 的语言编程方法，即在 M 文件 (M-File) 或命令窗口 (Command Windows) 中，通过由函数命令格式组成的语句建立控制系统模型，并对控制系统进行相关的仿真实验与分析。

2) 基于 MATLAB 的 Simulink 模块编程方法，即在 MATLAB 的 Model 文件或模型编辑窗口中，通过 MATLAB 控制系统工具箱 (Control System Toolbox) 提供的典型模块，进行模块选择、模块连接与模块参数设置等编辑操作，最后得到含有输入信号、系统模型和输出显示 (或输出转存) 的系统模型。


本书以自动控制原理的基本内容为核心，分别介绍 MATLAB 的语句编程仿真实验方法 (第 2~8 章) 和 Simulink 模块化仿真实验方法 (第 9 章)。以下简要介绍这两种方法的编程环境及编程概要。

1.2.1 基于 MATLAB 的语言编程

语言编程是 MATLAB 为用户提供的可利用常用函数和专用函数命令格式进行书写的编程方法，是一种 MATLAB 最通用和最灵活的编程方式。MATLAB 所提供的常用函数可实现几乎所有的数学运算和图形绘制，所以利用 MATLAB 的常用函数命令可以实现控制理论所需的运算与绘图。另一方面，MATLAB 还提供了用于自动控制系统研究所需的专用函数命令，从而使控制理论与控制系统的编程、仿真实验与分析变得更加便捷和直观。也就是说，采用常用函数命令下的编程来实现系统仿真，需要包括数学运算和图形绘制两个主要部分，而专用函数命令自身包含了数学运算、图形绘制或参数返回功能。MATLAB 是进行控制理论与控制系统的仿真实验、分析研究和灵活应用等内容的重要方法之一，熟悉和掌握 MATLAB 常规的语言编程方法可为学习和掌握 MATLAB 控制系统的专业语言编程提供帮助。

1. MATLAB 语言编程环境

要进行 MATLAB 语言编程，首先要进入 MATLAB 语言编程环境，即需要启动已安装在电脑中的 MATLAB 软件，并进入程序编辑环境。

1) MATLAB 软件的启动：双击桌面上的 MATLAB 快捷图标，或者单击桌面左下角的“开始”图标进入路径“程序/MATLAB*/*/MATLAB*.*”随后打开的 MATLAB 主界面。图 1-1 所示为启动 MATLAB7.0 后的主界面 (其他版本的主界面也基本类同)。

2) 进入 MATLAB 语言编程环境 (即打开编辑窗口)，有 3 种编程方式：①在命令窗口中编程。在 MATLAB 主界面的命令窗口 Command Window 中，将光标移至编程提示符“>>”后，输入语句命令，按〈Enter〉键执行程序；②在语言编程器中编程。在主界面的主菜单栏中选择“File”→“New”→“M-File”命令 (M-File 也称 M 文件，即保存时扩展名为“.m”)，随后弹出一个未命名 (Untitled.m) 的语言程序编辑窗口 (Editor)。用户可在其中进行程序的设计、编辑和运行等操作，如图 1-2 所示；③在 Word 文档中编程。MATLAB 兼

容一些 Office 下的文档类型，如 Word 文档、Excel 文件和记事本文件均可以作为数据文件的脚本；Word 文档可作为程序文件的编辑平台，在命令窗口和 Word 文件之间可进行内容的复制和格式转存。

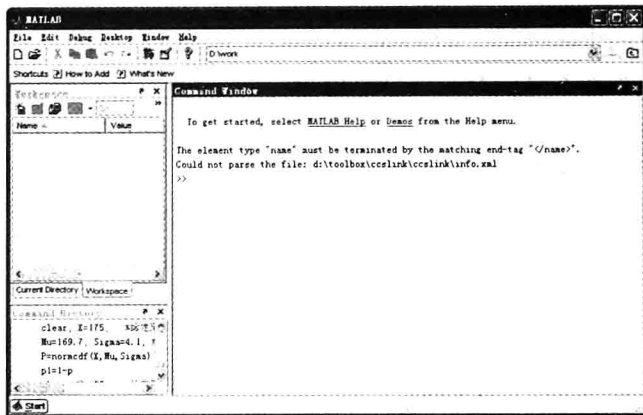


图 1-1 MATLAB 主界面



图 1-2 MATLAB 语言程序编辑窗口

2. MATLAB 语言程序的运行与结果显示

(1) 语言程序的运行

在上述编程方法中，直接在 Command Window（命令窗口）中编程的内容可按回车键直接运行。对于在 M 文件中编辑的内容，如果是一个完整的程序，则可以选择该 M 文件的窗口中的“Debug”→“Run”命令（见图 1-2 中的“Debug”下拉菜单；实际上，M 文件中程序的运行也是在 Command Window 中进行），或将欲运行的程序复制到 Command Window 中后，按回车键运行。对于 Word 文档中的程序，必须将其程序复制到 Command Window 中后，按回车键运行。

(2) 结果的显示

程序中的赋值变量和运算式，会在 MATLAB 运行后显示在 Workspace（工作窗口）中，双击 Workspace 中的变量符可查看到变量的内容。如果运算语句命令之后为无标点或逗号，则该运算语句执行的结果还将在 Command Window 中显示，如果之后为分号则运行结果不在 Command Window 中显示。

3. MATLAB 语言程序运行结果保存

语言程序运行结果包括 Command Window 中的程序和结果、Workspace 中的变量和运算结果（数据），以及由绘图命令得到的图形结果。当一个程序通过在 Command Window 中反复调试完成后，需要进行保存。程序内容保存方式有两种：复制到程序 M 文件中进行保存，或复制到 Word 文档中保存。对于 Workspace 中的数据，可通过主界面菜单“File”→“Save Workspace As”命令进行保存，即在 Workspace 中进行选择和保存（MATLAB 的数据文件以.mat 或.dat 为扩展名，分别称为二进制文件“.mat”和 ASCII 文件“.dat”）。数据文件主要用于 Workspace 窗口中的数据导入和保存。

4. MATLAB 语言程序的调用

在需要对已有的 MATLAB 文件中的程序或数据进行运行时，需要从相关的文件中调用所需的程序或数据。

1) 程序调用：对于 MATLAB 的程序 M 文件，打开该文件后可以在其窗口中选择“Debug”→“Run”命令，或复制到 Command Window 中运行，也可以将 Word 文档中的程序复制到命令窗口中运行。

2) 数据调用：对于 MATLAB 的数据 M 文件，可选择“File”→“Import Data”命令，在随后打开的数据文件列表中选择要打开的文件名，确认后该数据文件中的数据会导入到 Workspace 中。同样也可以将 Word 文档、Excel 表格和记事本中的数据导入到 Workspace 中。

5. MATLAB 语言编程的基本符号与函数命令

(1) 常用标点操作符

在 MATLAB 中，标点符号有着严格的规定，并具有特定的含义和功能，在操作和程序设计起着非常重要的作用，应当熟悉记牢。常用标点符号的功能与说明见表 1-1。

表 1-1 常用标点与符号的功能与说明

标点符号	说明
>>	双右大于号，命令输入符，在其后的内容为用户输入的程序
%	百分号，其后是用户的注释说明，MATLAB 不予执行。其按行有效，%的使用个数不限
空格	数据、向量或矩阵行中各元素的分隔
,	逗号，向量中或矩阵行中各元素的分隔，此时等价于空格。显示逗号前指令运行后的结果
;	分号，同行中命令间的分隔；不显示分号前指令运行后的结果。矩阵中每行元素的分隔
...	续行号，一段命令过长时的换行，在命令窗口编辑时，在其后使用〈Shift+Enter〉键完成换行
'	单引号，定义单引号中的内容为字符、符号、函数或指定名。用于向量或矩阵时表示转置
:	冒号，数据向量或序列的步长间隔，如 $x=0:0.1:2$ 表示生成从 0 到 2 步长为 0.1 的向量； $y=[1:10]$ 表示生成 1 至 10 的整数向量。表示矩阵的任意行或列
.	小数点，小数点及扩展名符
[]	方括号，定义矩阵、向量的标识。单元数组的结构标识
()	圆括号，用于函数参数、公式运算等标识。矩阵或向量元素、数组单元结构的提取或查询
{}	花括号，定义单元数组的标识。对单元数组元素内容的提取或查询
=	等号，赋值或定义运算式。在 MATLAB 运算结果中表示所得变量的结果

需要注意的是，由于 MATLAB 只接受英文格式的标点符号，所以在编程中要注意标点符号的切换和正确使用。如 MATLAB 接受单引号格式*’，而不接受其他形式的引号。

(2) 命令窗口的基本操作命令


用户除了可以利用菜单功能对 MATLAB 主界面进行操作外，还可以在命令窗口中输入命令语句进行相关操作。表 1-2 给出常用的基本命令及其功能说明。这些命令语句可在编程时使用。

表 1-2 命令窗口中的常用操作命令

命令	功能	命令	功能
cd(D:***)	将当前路径改变为 D:**	help fn	显示文件名为 fn 的帮助信息
clear	清除当前内存中的所有变量，恢复默认状态，同 clear all；若要清除指定变量 x 则用 clear x	hold	保持当前图形，等价于 hold on
cla	清除图形坐标	hold off	结束当前图形的保持状态
clc	清除命令窗口中的所有内容	matlabroot	返回 MATLAB 的根目录
clf	清除已打开的图形	openvar a	打开工作空间中的变量 a
copyfile(S,D)	复制源文件 S 至新目标文件 D	pwd	显示当前目录路径
delete	删除文件	quit	退出 MATLAB。等价于“File”→“Exit MATLAB”，同 exit
dir	显示当前路径下的文件目录和信息。同指令 what	support	打开 Math Works 公司的技术支持网站
disp(*) disp(b)	显示单引号中的内容* 显示指定变量 b 的内容	type p	显示文件名为 p 的内容
edit, edit p	创建一个 M 文件，打开文件 p 的窗口	who	显示当前工作空间的变量

在命令窗口中运行程序时，所要运行的程序文件必须在主菜单的当前路径上，即在 Current Direction 中显示的路径。如果不在，则要将相关文件的路径变为当前路径。具体有两种操作方法。

1) 通过图标工具栏中的当前路径栏 (Current Direction) 进行变更；

2) 通过命令窗口执行由命令 cd(*)所构成的语句来实现。如当前的默认 (开机) 路径为 “C:\work”，而需要执行的程序文件在 “D:\080707” 路径上。此时改变路径的具体操作：先单击当前路径的栏右端下拉目录按钮 ，从中查找，如果存在 “D:\080707”，则单击后即变为当前路径。如果栏中不存在，则在命令窗口中的提示符 “>>” 后输入语句: cd(D:\080707)并回车执行，则当前路径即变为 “D:\080707”。也可以选择 “File” → “Set Path” 命令，并在弹出的对话框进行设置。

(3) 算术、逻辑和比较运算符

MATLAB 的算术运算分为矩阵运算和数组运算，还有 4 种基本的逻辑运算和 6 种比较判断运算。矩阵运算是指按矩阵理论的运算规则进行定义的运算方式。当矩阵为一个元素 (即 1×1 矩阵) 时所进行的运算，则是通常所指的数的运算。数组运算是指矩阵或向量之间对应元素的算术运算，也称点运算。参与点运算的变量必须是同阶的。当变量是两个同阶矩阵时，称矩阵点运算。在加减运算时，矩阵运算和矩阵点运算一样，都是点运算。常用的运算符及其功能说明如表 1-3 所示。

表 1-3 常用算术、逻辑和比较运算符及功能说明

运算符	功能说明	运算符	功能说明
+	矩阵或数组的加 (对应元素相加)	-	矩阵或数组的减 (对应元素相减)
*	矩阵的乘, 如 $A*B=AB$.*	数组中对应元素相乘, $[x_1, x_2, x_3].*[y_1, y_2, y_3]=[x_1*y_1, x_2*y_2, x_3*y_3]$
^	矩阵的乘方, $A^3=A*A*A=A^3$.^	数组中各元素的乘方或对应乘方, $[x_1, x_2, x_3].^a=[x_1^a, x_2^a, x_3^a]$; $[x_1, x_2, x_3].^[y_1, y_2, y_3]=[x_1^{y_1}, x_2^{y_2}, x_3^{y_3}]$
\	矩阵左除, $A\B=A^{-1}*B$.\	数组中各元素的左除或对应左除, $[x_1, x_2, x_3].\a=[a/x_1, a/x_2, a/x_3]$ $[x_1, x_2, x_3].\[y_1, y_2, y_3]=[y_1/x_1, y_2/x_2, y_3/x_3]$
/	矩阵右除, $A/B=A*B^{-1}$	/	数组中各元素的右除或对应右除, $[x_1, x_2, x_3]/a=[x_1/a, x_2/a, x_3/a]$ $[x_1, x_2, x_3]/[y_1, y_2, y_3]=[x_1/y_1, x_2/y_2, x_3/y_3]$
&	与, 结果为 0 代表“假”; 1 代表“真”	~	非, 结果为 0 代表“假”; 1 代表“真”
	或, 结果同上	xor(a,b)	a 和 b 异或, 结果同上
>	大于关系, $a>b$ 真为 1, 假为 0	>=	大于等于关系, $a\geq b$ 真为 1, 假为 0
<	小于关系, $a<b$ 真为 1, 假为 0	<=	小于等于关系, $a\leq b$ 真为 1, 假为 0
==	等于关系, $a=b$ 真为 1, 假为 0	~=	不等于关系, $a\neq b$ 真为 1, 假为 0

(4) 常用默认变量

MATLAB 默认和自定义了一些变量 (或常量), 但这些默认变量也可由用户作为任意变量自行定义和使用, 且用户拥有对默认变量的优先定义和使用权。使用后的默认变量可由清除命令形式“clear 变量”来恢复为默认变量。如 i 和 j 是 MATLAB 默认的复数的虚部常量 $\sqrt{-1}$, 但用户可作为任何变量来自由使用, 使用后若想将其恢复为 MATLAB 默认常量, 可用“clear i”或“i=sqrt(-1)”来恢复其为默认常量。通常情况, 建议尽量避开对默认常量的使用。常用的默认常量如表 1-4 所示。

表 1-4 默认常量及说明

常量符	说明
Ans	临时常量, 表示程序中那些无赋值算式的计算结果, 但在 Workspace 中只给出当前程序最后一个无赋值算式的计算结果
end	默认指对矩阵或向量、数组中取值可达的最后一个数。在循环结构中为循环结束命令
eps	$\text{eps} = 2^{(-52)}$ 或 $2.22e-16$ 代表最小数, 小于此数时则用 0 表示, 运算时 eps 代表 0+, -eps 代表 0-, 即 $\text{eps} = 0+$, $-\text{eps} = 0-$
exp(a)	指数 ea, $e = \exp(1) = 2.7183$
i, j	虚数 $\sqrt{-1}$, 恢复默认值时用语句 $i = \text{sqrt}(-1)$ 或 clear i
Inf 或 inf	无穷, $\text{Inf} = +\infty$, $-\text{Inf} = -\infty$
NaN	非数, 不确定计算时, 如 Inf/Inf 、 $0/0$ 、 $0 \times \infty$ 等, MATLAB 当作有效结果处理
pi	圆周率 π
vpa(c,n)	可变精度显示, c 为变量, n 为显示的有效位数, n 默认时为 32 位精度显示。如: $\text{vpa}(\pi) = 3.1415926535897932384626433832795$, $\text{vpa}(1/3, 7) = .3333333$; $\text{vpa}(\exp(1), 4) = 2.718$

(5) 常用数组和矩阵的生成函数命令

MATLAB 提供了一些常用数组和矩阵的生成函数命令, 如表 1-5 所示。

表 1-5 常用矩阵生成函数命令与说明

命令格式	说 明	命令格式	说 明
cell(n)	生成 $n \times n$ 的空单元数组	cell(n,m)	生成 $n \times m$ 的空单元数组, 同 cell([n,m])
eye(n)	生成 n 阶单位阵	ones(n)	生成 n 阶全 1 矩阵
ones(n,m)	生成 $n \times m$ 阶全 1 矩阵	ones(n,m,p)	生成 $n \times m$ 阶 p 页 (维) 全 1 矩阵
magic(n)	生成 n 阶魔方阵	zeros(n)	生成 n 阶零矩阵
rand(n)	生成 n 阶均匀分布的随机矩阵	rand(n,m)	生成 $n \times m$ 阶均匀分布的随机矩阵
randn(n,m)	生成 $n \times m$ 阶正态分布的随机矩阵	randn(n)	生成 n 阶正态分布的随机矩阵
randn(n,m,p)	生成 $n \times m$ 阶 p 页 ($p+1$ 维) 正态分布随机矩阵	A=[]	生成任意阶任意页的空矩阵 A , 常作为初始设定

在表 1-5 中, eye(n)和 magic(n)只能生成 n 阶二维 (1 页) 方阵; 格式 $A=\text{函数}(\text{size}(B))$ 可以生成与 B 同阶的函数矩阵 A , 如 $A=\text{cell}(\text{size}(B))$ 、 $C=\text{eye}(\text{size}(B))$ 生成与 B 同阶的空单元数组 A 、单位矩阵 C 。

(6) 常用矩阵变换函数命令格式

矩阵、向量或元素的组合或变换是指对它们的如转置、旋转或排列等操作。常用的函数命令及说明见表 1-6。

表 1-6 矩阵变换的常用格式及说明

命 令 格 式	说 明
A.'	将矩阵 A 转置, 即 $[A(i,j)] \rightarrow [A(j,i)]=A.'$
A'	将矩阵 A 共轭转置, 即 $[A(i,j)] \rightarrow [\bar{A}(j,i)]=A'$
cat(d,A1,A2,...,An)	将同行列矩阵 A_1, A_2, \dots, A_n 组合为一个矩阵, $d=1$ 时, 对同列矩阵按列组合, 等价于 $[A_1; A_2; \dots; A_n]$; $d=2$ 时, 按行组合, 等价于 $[A_1, A_2, \dots, A_n]$
horzcat(A1,A2,...,An)	将同行矩阵 A_1, A_2, \dots, A_n 水平组合为一个矩阵
vertcat(A1,A2,...,An)	将同列矩阵 A_1, A_2, \dots, A_n 垂直组合为一个矩阵
fliplr(A)	将矩阵 A 左右翻折
flipud(A)	将矩阵 A 上下翻折
rot90(A)	将矩阵 A 逆时针旋转 90° 。如对一向量旋转两次可得原向量的逆排序
[B]=sort(A,d) [B,IX]=sort(A,d)	对矩阵 A 中的元素按从小到大排序。 $d=1$ 时, 对每列元素进行排列; $d=2$ 时, 对每行元素进行排列。 IX 为排序后 B 中元素在 A 中各列或各行的序号

(7) 常用函数

用 MATLAB 表示的函数可以是符号型函数 (运算之前必须对常量、自变量或函数进行符号型定义), 也可以是数值型函数 (运算之前对变量进行赋值操作)。常用函数的命令格式及功能说明如表 1-7 所示。

表 1-7 常用函数的命令格式及其功能说明

命令格式	说 明	命令格式	说 明
sym(A)	将数值型变量（或矩阵） A 转为符号型变量	subs(f,x,v)	用变量 v 替代符号函数 f 中的变量 x 即 $f x=v$ 。可以嵌套使用 subs(), 如 subs(subs(f,x,v1),y,v2)
y=subs(y,v)	将符号函数 y 中的默认变量替换为数值变量 v	collect(f)	将函数 f 合并同类项
factor(f)	将函数 f 因式分解	expand(f)	将函数 f 展开
[fs,H]=simple(f)	将函数 f 简化为最短形式 fs, 并返回简化方法 H	simple(f)	用所有方法对 f 化简, 显示所有简化结果, 最后给出的是最短简化形式, 其中 convert(*) 表示用函数 * 转换的结果
pretty(f)	将函数 f 转为手写形式	simplify(f)	将函数 f 简化为一种简洁形式
y=sin(x)	正弦函数。三角函数的自变量 x 默认为弧度, 后同	x=asin(y)	反正弦函数
y=cos(x)	余弦函数	x=acos(y)	反余弦函数
y=tan(x)	正切函数	x=atan(y)	反正切函数
y=cot(x)	余切函数	x=acot(y)	反余切函数

(8) 基本绘图函数命令格式

MATLAB 不仅具有强大的计算功能, 还具有非常好的数据可视化的绘图功能, 几乎可以满足一般科学计算中所需的图形绘制与交互功能, 如二维平面曲线、三维空间曲线和曲面等静态和动态显示等。也可以实现不同坐标系之间的转换和图形绘制。基本绘图函数命令格式及说明见表 1-8。

表 1-8 基本绘图函数命令格式及说明

命令格式	说明
figure 或 figure(m)	打开一个新的绘图窗口 (定义标号为 m)
subplot(n,m,k) 或 subplot nmk	在一个图形窗口中设置 n 行 m 列 ($n \times m$) 个子图形窗口, 并选定第 k 个子窗口作为当前绘图窗口
plot(x,y,'color line marker')	绘制曲线 $y(x)$, color 为图形颜色, line 为线的类型, marker 为采样点符号标记类型。具体参数类型见表 1-9
plot(x1,y1,'color1 line1 marker1', x2,y2,'color2 line2 marker2',...)	同时绘制多条曲线 $y_1(x_1), y_2(x_2), \dots$ 其他参数见上
grid 或 grid on	为图形加网格
grid minor	为图形加密集网格
grid off	去除图形网格
label='*'	在图形区添加说明的内容*
xlabel('**')	为 x 轴 (横轴) 添加标识*
ylabel('**')	为 y 轴 (纵轴) 添加标识*
title('**')	为图形加标题*
legend('*', '**',...)	在图形右上角处加曲线标注*和**, 标注数与曲线数对应

表 1-9 给出了绘制图形时, 图形的颜色、线型和标记的字符表示及含义。