

控制工程基础

杨建玺 徐莉萍 等 编著

本书由河南科技大学教材出版基金资助出版

科学出版社

北 京

内 容 简 介

本书作为一门技术基础课教材,在介绍控制理论的基本原理和基本概念的同时,以机械系统、电路系统为背景,阐明了控制理论在工程中的应用。本书的主要内容包括数学基础——拉普拉斯变换、传递函数、时间响应分析、系统误差分析、频率特性分析、系统稳定性分析和系统的校正设计。每章都给出应用实例、MATLAB 实现以及习题。

本书可作为机械类各专业大学本科学生的教材,也适于工程技术人员参考。

图书在版编目(CIP)数据

控制工程基础 / 杨建玺, 徐莉萍等编著. — 北京: 科学出版社, 2008

ISBN 978-7-03-021307-5

I. 控… II. 杨… III. 自动控制理论 IV. TP13

中国版本图书馆 CIP 数据核字(2008)第 031287 号

责任编辑: 耿建业 于宏丽 / 责任校对: 陈玉凤

责任印制: 刘士平 / 封面设计: 耕者设计工作室

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

印刷

科学出版社发行 各地新华书店经销

*

2008 年 4 月第 一 版 开本: B5 (720×1000)

2008 年 4 月第一次印刷 印张: 15 1/4

印数: 1—3 000 字数: 290 000

定价: 38.00 元

(如有印装质量问题, 我社负责调换〈长虹〉)

前 言

应用控制理论解决工程实际问题越来越受到工程技术人员的重视,控制理论和控制工程解决问题的方法已成为工程技术人员分析问题和解决问题的有效手段。

本书作为一门技术基础课的教材,在明确基本控制原理和控制理论基本概念的前提下,以机械系统(特别是质量-弹簧-阻尼系统)、电路系统等为背景,力求阐明控制工程的基本知识和解决问题的基本方法。同时,本书还增加了很多应用实例,不仅有利于加强控制工程基础与专业课之间的联系,而且为学生今后将控制理论应用于工程实际打下良好的基础。此外,本书还在每一章的最后给出了 MATLAB 的实现方法,增加了解决控制工程基础问题的手段。

全书由杨建玺、徐莉萍等编著,第 1 章绪论,由杨建玺编写;第 2 章数学基础——拉普拉斯变换和第 8 章系统的校正设计,由张明柱、赵书尚编写;第 3 章传递函数,由陈龙编写;第 4 章时间响应分析和第 5 章系统误差分析,由张书涛编写;第 6 章频率特性分析,由徐莉萍编写;第 7 章系统稳定性分析,由赵书尚编写。全书由杨建玺统稿。

限于编者的水平,书中难免有不妥之处,敬请读者批评指正。

编 者

2008 年 1 月

目 录

前言

第 1 章 绪论	1
1.1 控制论概述	1
1.2 控制系统的工作原理与组成	2
1.2.1 工作原理	2
1.2.2 控制系统的组成	4
1.3 控制系统的分类与基本要求	5
1.3.1 控制系统的分类	5
1.3.2 控制系统的基本要求	7
1.4 MATLAB 语言简介	8
1.4.1 MATLAB 的系统界面	8
1.4.2 MATLAB 数学运算	11
1.4.3 MATLAB 绘图	18
1.5 本课程的特点及要求	27
习题	28
第 2 章 数学基础——拉普拉斯变换	29
2.1 复数和复变函数	29
2.1.1 复数的概念	29
2.1.2 复数的表示法	29
2.1.3 复变函数	30
2.1.4 极点与零点概念	30
2.2 拉氏变换	31
2.2.1 拉氏变换定义	31
2.2.2 典型时间函数的拉氏变换	32
2.2.3 拉氏变换的性质	34
2.3 拉氏反变换及其数学方法	39
2.3.1 拉氏反变换	39
2.3.2 拉氏反变换的数学方法	40
2.4 拉氏变换的应用	43
2.5 拉氏变换的 MATLAB 实现	45

习题	46
第 3 章 传递函数	47
3.1 概述	47
3.1.1 数学模型的概念	47
3.1.2 线性系统与非线性系统	48
3.2 系统微分方程的建立	48
3.2.1 机械系统的微分方程	48
3.2.2 电气系统的微分方程	52
3.2.3 系统元件间的负载效应	54
3.2.4 非线性微分方程的线性化	54
3.3 传递函数	58
3.3.1 传递函数的概念	58
3.3.2 传递函数的零、极点	59
3.3.3 典型环节及其传递函数	59
3.4 系统框图及简化	67
3.4.1 系统框图的组成	67
3.4.2 方框图的绘制	68
3.4.3 系统方框图的简化	69
3.5 信号流图与 Mason 公式	73
3.5.1 信号流图	74
3.5.2 Mason 公式	75
3.6 应用实例	76
3.6.1 机床进给传动链	76
3.6.2 汽车悬挂系统	77
3.6.3 车削过程	78
3.6.4 打印轮控制系统	80
3.7 系统数学模型的 MATLAB 实现	82
3.7.1 传递函数模型	82
3.7.2 零极点增益模型	83
3.7.3 模型转换	83
3.7.4 动态结构图	84
习题	84
第 4 章 时间响应分析	88
4.1 时间响应	88
4.1.1 时间响应的概念	88

4.1.2 典型输入信号	89
4.2 一阶系统的时间响应	91
4.2.1 一阶系统的数学模型	91
4.2.2 一阶系统的单位阶跃响应	91
4.2.3 一阶系统的单位脉冲响应	92
4.2.4 一阶系统的单位斜坡响应	93
4.3 二阶系统的时间响应	94
4.3.1 二阶系统的数学模型	94
4.3.2 二阶系统的单位阶跃响应	96
4.3.3 二阶系统的单位脉冲响应	98
4.4 高阶系统的时间响应	100
4.5 瞬态响应的性能指标	101
4.5.1 瞬态响应的性能指标	101
4.5.2 二阶系统的瞬态响应指标	102
4.6 时间响应的 MATLAB 实现	108
习题	110
第 5 章 系统误差分析	111
5.1 概述	111
5.1.1 误差与偏差	111
5.1.2 系统的类型	113
5.2 稳态误差的计算分析	113
5.2.1 静态误差系数与稳态误差	113
5.2.2 扰动作用下的稳态误差	117
习题	119
第 6 章 频率特性分析	121
6.1 概述	121
6.2 频率响应与频率特性	122
6.2.1 频率响应	122
6.2.2 频率特性	123
6.2.3 频率特性的求法	124
6.2.4 频率特性的特点	126
6.2.5 频率特性的表示方法	126
6.3 频率特性的极坐标图(Nyquist 图)	127
6.3.1 极坐标图	127
6.3.2 典型环节的极坐标图	127

6.3.3 系统 Nyquist 图的一般画法	132
6.4 频率特性的对数坐标图(Bode 图)	138
6.4.1 对数坐标图	138
6.4.2 典型环节的 Bode 图	138
6.4.3 绘制系统 Bode 图的步骤	147
6.5 开环频率特性对数坐标图的含义	151
6.6 最小相位系统及频域性能指标	153
6.6.1 最小相位系统的概念	153
6.6.2 由 Bode 图估计最小相位系统的传递函数	153
6.6.3 频域性能指标	154
6.7 频率特性的 MATLAB 实现	158
6.7.1 利用 MATLAB 绘制 Nyquist 图	158
6.7.2 利用 MATLAB 绘制 Bode 图	159
6.7.3 利用 MATLAB 求系统的频域特征量	160
习题	161
第 7 章 系统稳定性分析	163
7.1 稳定性概述	163
7.2 Routh-Hurwitz 稳定性判据	165
7.2.1 Routh 稳定性判据	165
7.2.2 Hurwitz 稳定性判据	169
7.3 Nyquist 稳定性判据	171
7.3.1 基本原理	171
7.3.2 Nyquist 法判别系统的稳定性举例	174
7.3.3 具有延时环节的系统的稳定性分析	177
7.4 系统的相对稳定性	178
7.4.1 相位裕量和幅值裕量	178
7.4.2 条件稳定系统	182
7.5 应用实例	183
7.5.1 Routh-Hurwitz 稳定判据应用实例	183
7.5.2 Nyquist 稳定判据应用实例	184
7.5.3 相对稳定性应用实例	186
7.6 稳定性分析的 MATLAB 实现	188
7.6.1 利用 MATLAB 求系统的特征根	188
7.6.2 利用 MATLAB 中的函数直接求解	189
思考题	191

习题	191
第 8 章 系统的校正设计	194
8.1 概述	194
8.1.1 系统的时域和频域性能指标	194
8.1.2 校正的概念	195
8.1.3 校正的类型	195
8.2 串联校正	196
8.2.1 增益校正	197
8.2.2 相位超前校正	197
8.2.3 相位滞后校正	200
8.2.4 相位滞后-超前校正	204
8.3 并联校正	206
8.3.1 反馈校正	207
8.3.2 顺馈校正	208
8.4 PID 校正	212
8.4.1 PD 控制作用	213
8.4.2 PI 控制作用	216
8.4.3 PID 控制作用	217
8.5 应用实例	218
8.5.1 电压-转角位置随动系统	218
8.5.2 直流电机调速系统	222
8.6 系统校正的 MATLAB 实现	224
习题	227
参考文献	231

第 1 章 绪 论

1.1 控制论概述

控制论是自动控制、电子技术、计算机科学等多种学科相互渗透的产物,是在 20 世纪 40 年代酝酿形成的。控制论研究的是一个系统,这个系统可以是具体的一个执行机构,如厕所马桶的液面自动调节装置、机床进给系统等,也可以是一个工厂管理系统、人体自身控制系统等。控制论主要研究系统的性能,包括静态性能和动态性能,也就是研究系统的稳定性、准确性和快速性。

自从有了人类社会,控制论的应用就已经开始了。因为人们总是根据眼睛等感觉器官将得到的信息反馈给大脑,从而指导手、腿等执行器官完成计划的工作。其实,人类社会的发展过程就是一个完整的控制系统改造过程。如图 1.1 所示,人们为了做成某项产品,首先在大脑中产生一个计划,大脑指挥人体活动器官,通过手作用在控制对象上,形成产品雏形,通过人体感觉器官将产品雏形反馈给大脑,大脑做出与原计划差异的判断,再指挥手进行修正,直到完成满意的产品为止。从这个过程我们可以看出,人参与了整个控制过程。人类社会的发展也是一个控制系统发展的过程。随着社会文明的进步,出现了石器时代,也就是控制对象是石器,而到了铁器时代的时候,控制对象更新为铁器,后来有了机器,由机器作为控制对象,这使人们得以大大减轻体力劳动,而测量工具仍是离线测量工具,由人眼测量到利用测量工具如直尺、千分尺等,这一切都仅仅是控制对象和系统中相应的环节发生了改变,却让人类奋斗了上千年的时间。今天很多问题又进一步得到解决,开始使用各种各样的传感器来替代人的感觉器官,使用计算机来替代人的大脑,可以说,人类社会的进步实质上就是逐步用机器、传感器、电脑等将人从控制系统中解放出来。

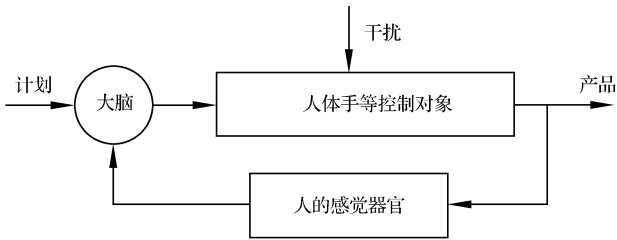


图 1.1 人类社会发展控制系统框图

控制论作为一门系统科学,其理论发展时间还很短暂。1788 年 Watt 发明了蒸汽机,其工作原理如图 1.2 所示,但没有对其离心调速器所带来的调速问题进行分析研究,1868 年 Maxwell 在 *Proceeding of the Royal Society of London* 第 16 卷上发表了“论调速器”一文,专门对离心调速问题进行了深入细致的数学分析,成为控制理论的奠基人。之后的 1884 年 Routh 又提出了线性系统稳定性的判据,1932 年 Nyquist 提出了系统稳定性的又一种判别方法等,这些都奠定了古典控制论的基础。1948 年 Wiener 将前人所做的工作进行了系统性的总结,出版了 *Cybernetics*(含义为控制论、方法论),形成了完整的经典控制理论。第二次世界大战期间,由于战争的需要,欧洲一些科学家致力于研究状态空间法、极大值理论、动态规划、Kalman 滤波、最优控制等,形成了现代控制理论。随着计算机技术的不断发展,各种复杂的问题通过数值方法得以解决,各种新的控制算法不断涌现,构成了所谓智能控制理论。

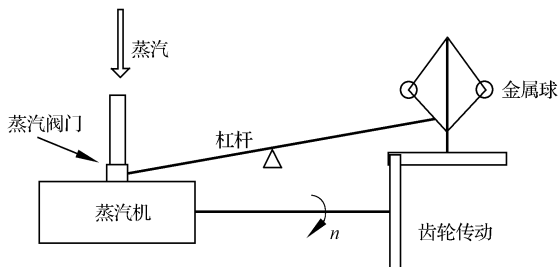


图 1.2 蒸汽机调速器调速工作原理图

本课程主要讲述经典控制理论在机电工程方面的应用,也就是经典控制论的内容,更多的列举机械和电路方面的例子。

1.2 控制系统的工作原理与组成

1.2.1 工作原理

系统是指能够完成某种功能的元件或者元件集合。控制工程中所指的系统是广义的。广义系统不限于具体的物理系统(如质量-弹簧-阻尼系统,一台机器);也可以是一个过程(如切削过程,生产过程);同时,也可以是一些抽象的动态现象。所谓控制系统,是指系统的输出能按照要求的输入进行调节的系统。图 1.3 为人工控制水箱液位的示意图。人工调节进水阀门,控制输入流量 q_1 ,克服水箱出水口扰动量 q_2 的变化,达到控制水箱液位的目的。其调节过程为:给定液面高度 h_0 ,当人眼观察得到的液面高度小于给定值 h_0 时,开启阀门加大流量 q_1 ,直至达到 h_0 为止。

图 1.4 为水箱液位自动控制系统。采用浮子作为测量元件代替人眼,用电位器作为比较器代替人脑,用放大器、电动机以及减速器作为执行元件代替人手。水箱液位的自动控制过程为:当电位器的电刷位于中点位置,其输入电压为 u_1 时,控制电压为 $\Delta u = u_1$,电机 M 不动,控制阀门有一定开度,水箱中进水量与出水量相等,液位保持在希望值上。一旦进水量 q_1 或出水量 q_2 发生变化,如使液面上升时,浮子位置也相应升高,通过杠杆作用使电位器电刷下移,此时输入电压 $\Delta u = u_1 - u_2$ 升高,驱动电机和减速器减小阀门开度,使水箱的进水量减少。水箱液面开始下降,浮子位置也相应下降,直至偏差 $\Delta u = 0$,电位器电刷回到中点位置,系统重新处于平衡状态,液面恢复到给定的高度。反之,若水箱液位下降,则系统会自动增大阀门开度,加大进水量,使液位上升到指定高度。

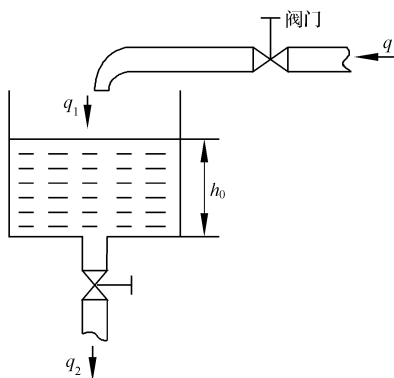


图 1.3 液位人工控制系统示意图

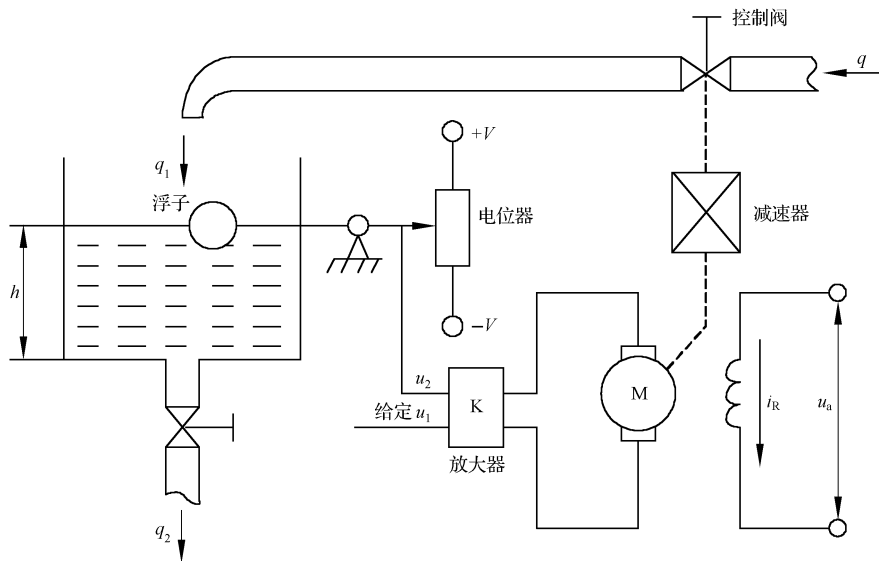


图 1.4 水箱液位自动控制系统示意图

由此可见,控制系统的工作原理可以归纳如下:

- (1) 检测被控制量或输出量的实际值。
- (2) 将实际值与给定值进行比较得出偏差值。

(3) 用偏差值产生控制调节去消除偏差。

这种基于反馈原理,通过检测偏差再纠正偏差的系统称为反馈控制系统或闭环控制系统。通常反馈控制系统至少具备检测、比较和执行三个基本功能。

为了表明控制系统的调节原理以及信号的传递情况,通常把系统的各个元件用方框图表示,并用箭头标明各作用量的传递方向,用 $\otimes$ 表示比较元件,每一个方框代表一个环节。每个环节的作用是单向的,且输出受输入的控制。水箱自动控制系统框图如图 1.5 所示。

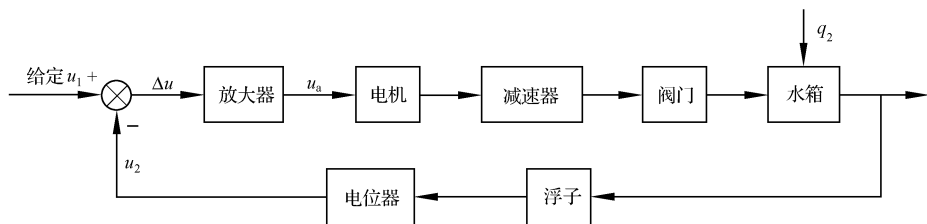


图 1.5 水箱液位自动控制系统框图

1.2.2 控制系统的组成

上述水箱液位自动控制系统中的电机、减速器和阀门合在一起完成了一个执行元件所完成的工作,浮子和电位器可以看作是一个检测元件,同时,电位器还是一个比较元件。从而可以将一般控制系统的框图归纳表示为图 1.6 所示的形式。

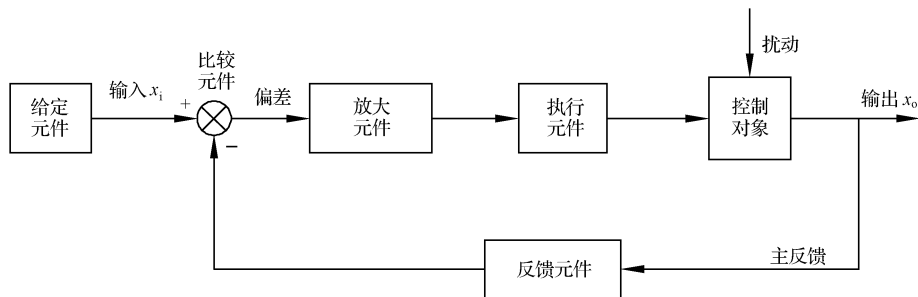


图 1.6 自动控制系统框图

由图 1.6 可以看出,一般的控制系统包括:

- (1) 给定元件——主要用于产生给定信号或输入信号。
- (2) 检测元件——测量被控量或输出量,产生反馈信号,并反馈到输入端。
- (3) 比较元件——用于比较输入信号和反馈信号的大小,产生反映两者差值的偏差信号。
- (4) 放大元件——对较弱的偏差信号进行放大,以推动执行元件动作。放大

元件有电气的、液压的和机械的。

(5) 执行元件——用于驱动被控对象的元件。如伺服电机、液压马达、液压缸以及减速器和调压器等。

(6) 控制对象——亦称被调对象。在控制系统中,运动规律或状态需要控制的装置称为控制对象,如水箱液位控制系统中的水箱。

由图 1.6 还可以看出,系统的各作用信号和被控制信号有:

(1) 输入信号——又称为控制量或调节量,它通常由给定信号电压构成,或通过检测元件将非电输入量转换成信号电压。如给定电压 u_1 。

(2) 输出信号——又称为输出量、被控制量或者被调节量。它是被控制对象的输出,表征被控对象的运动规律或状态的物理量。如液位控制系统中的液面高度。

(3) 反馈信号——它是输出信号经过反馈元件变换后加到输入端的信号。若反馈信号的符号与输入信号相同,称为正反馈;反之,称为负反馈。控制系统中的主反馈一般采用负反馈,以免系统失控。

(4) 偏差信号——它是系统输入信号与反馈信号叠加的结果,是比较环节的输入。

(5) 扰动信号——又称为干扰信号。扰动信号是指偶然的无法加以人为控制的信号。扰动信号也是一种输入信号,通常对系统的输出产生不利的影响。

1.3 控制系统的分类与基本要求

1.3.1 控制系统的分类

控制系统的种类很多,按照实际应用情况和分类方法的不同,可对控制系统做如下分类:

1) 按输入量的变化规律进行分类

(1) 恒值控制系统。恒值控制系统的输入量是一个恒定值,运行过程中就不再改变。这种控制系统的任务是保证在任何扰动作用下系统的输出量为恒定值。

1.2 节中的液位控制系统就是一个恒值控制系统。工业生产中的温度、压力、流量等参数的控制,以及某些动力机械的速度控制、机床的位置控制等均属此类系统。

(2) 程序控制系统。这种控制系统的输入量不为恒定值,其变化规律是预先知道的。可将输入量的变化规律预先编成程序,由程序发出控制指令,在输入装置中再将控制指令转换为控制信号,经过全系统的作用,使控制对象按照指令的要求运动。

图 1.7 所示为一个数控机床切削加工的程序控制系统。将待加工的工件按图

样要求预先编制成加工程序,将程序指令通过输入装置送到数控装置进行计算,产生的控制脉冲使刀具和工件按程序指令的要求运动,这样就加工出所需的工件。

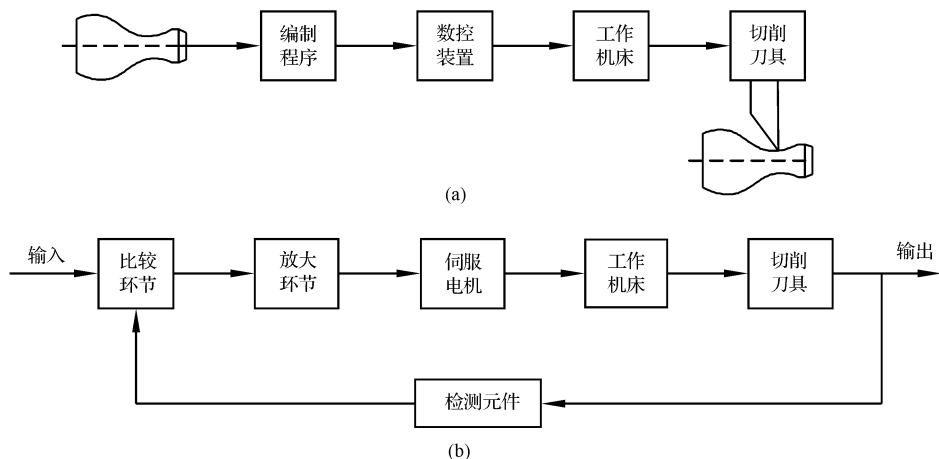


图 1.7 程序控制系统

(a) 数控加工过程原理图; (b) 数控机床进给系统框图

(3) 随动系统。随动系统又称伺服系统。这种控制系统输入量的变化规律是不能预先确定的。当系统的输入量发生变化时,要求输出量迅速平稳地随着输入量变化,并且能排除各种干扰因素的影响,准确地复现控制信号的变化规律。控制指令可以由操作者根据需要随时发出,也可以由目标物或相应的测量装置发出。如机械加工中的仿形机床、武器装备中的火炮自动瞄准系统以及导弹自动跟踪系统等均属于随动系统。

2) 按系统中传递信号的性质分类

(1) 连续控制系统。连续控制系统是指系统中各部分传递的信号都是连续时间变量的系统。连续控制系统又可分为线性系统和非线性系统。能用线性微分方程描述的系统称为线性系统,不能用线性微分方程描述、存在着非线性部件的系统称为非线性系统。

(2) 离散控制系统。离散控制系统是指系统中某一处或几处的信号是以脉冲序列或数字量传递的系统,又称数字控制系统。由于连续控制系统和离散控制系统的信号形式差别较大,因此在分析方法上有明显的不同。连续控制系统以微分方程来描述系统的运动状态,并用拉普拉斯变换法求解微分方程;而离散控制系统则用差分方程来描述系统的运动状态,用 z 变换法引出脉冲传递函数来研究系统的动态特性。

此外,还可以按描述系统的数学模型将控制系统分为线性控制系统和非线性控制系统;按系统部件的类型分为机电控制系统、液压控制系统、气动控制系统、电气控制系统等。

1.3.2 控制系统的基本要求

评价一个控制系统的好坏,其指标是多种多样的。对每一个具体系统,由于控制对象不同,工作的方式不同,完成的任务不同,因此,对系统性能的要求往往也不完全一样,甚至差异很大。但是,对控制系统的基本要求(即控制系统所需的基本性能)一般可归纳为:稳定性、准确性、快速性,即“稳、准、快”。系统的“稳、准、快”与有关的性能一起,统称为系统的动态性能或动态特性。因此,又可以说,本门课程是研究机械或者电路等控制系统的稳定性、准确性和快速性的。在学习本课程时可知,这又集中体现在系统的单位脉冲响应(时域中的根本特性)与频域特性(频域中的根本特性)上。现将“稳、准、快”的基本概念介绍如下:

1) 系统的稳定性

稳定性的要求是一般的控制系统正常工作的首要条件,而且是最重要的条件。所谓稳定性是指系统在干扰信号作用下,偏离原来的平衡状态,当干扰取消之后,随着时间的推移,系统恢复到原来平衡状态的能力。一个系统如果不稳定或失稳,它的行为便不受预定的约束,受控量将忽大忽小,摇摆不定,或者使运动发散,以致不能保持原定的工作状态不变。这种系统是不能完成控制任务的。因此,任何一个控制系统,要想完成令人满意的工作,首先应该是稳定的,也就是说应该具有这样的性质:输出量对给定的输入量的偏离初始值应该随着时间的推移逐渐趋近于零。但必须指出,稳定性的要求应该考虑到满足一定的稳定裕度,以便照顾到系统工作时参数可能发生的变化,以免由此变化而导致系统失稳。

2) 响应的准确性

准确性是指在过渡过程结束后输出量与给定的输入量(或同给定输入量相应的稳态输出量)的偏差,它又称为静态偏差或稳态精度。它也是衡量系统工作性能的重要指标。人们总是希望由一个稳态过渡到另一个稳态,输出量尽量接近或复现给定的输入量,或者说要求稳态精度尽可能高。由于外界干扰和给定的输入量经常在变化,系统实际上经常处在不断调整的过程中,但对缓慢变化的干扰或输入在一定时间内,系统的输出大致可视作是不变的。值得指出的是,对于同一系统,输入的变化规律不同,系统的稳态精度也不同。

3) 响应的快速性

这是在系统稳定的前提下提出来的。所谓快速性,就是指当系统的输出量与给定的输入量(或同给定输入量相应的稳态输出量)之间产生偏差时,消除这种偏差的快慢程度。可见快速性是衡量系统性能的一个很重要的指标。在分析和研究控制系统时,对衡量其快速性一般有两种提法:

一种提法是在阶跃信号作用下,用系统的过渡过程时间的长短(图 1.8 的 t_s)来衡量。当 $t > t_s$, 有 $|x_o(t) - x_o(\infty)| \leq \Delta \cdot x_o(\infty)$, 式中 Δ 为给定的允差,一般

取 $\Delta=0.02$ 或 0.05 , $x_o(\infty)$ 为稳态输出量, 过渡过程时间或称调整时间 t_s 越小, 说明系统从一个稳态过渡到另一个稳态所需要的时间越短, 也就是系统的响应速度越快。

另一种提法是用在过渡过程中。出现 $x_o(t) > |x_o(\infty)|$ 的情况 (表示系统产生了超调现象) 时的超调量 M_p 来衡量, 见图 1.8。 M_p 越小, 则说明系统的过渡过程进行得越平稳。值得注意的是, 超调现象严重的系统, 不仅使组成系统的各个元件处于恶劣的工作条件下, 而且过渡过程在长时间内不能结束, 致使系统的误差不能很快地减小到允许范围之内。

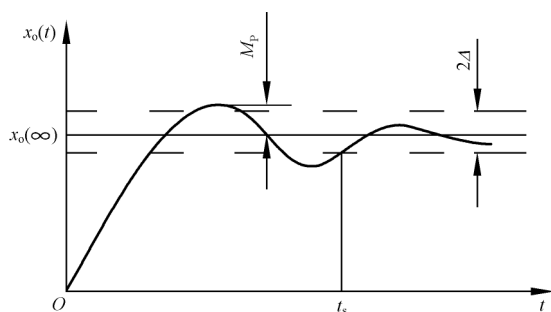


图 1.8 振荡系统的阶跃响应

综上所述, 人们要求控制系统中被控对象的行为 (响应、输出、动态历程) 应尽可能迅速而准确地实现它所应遵循的变化规律。这规律由“给定环节”的输出即系统的输入决定。当然, 毫无疑问, 这里就隐含了系统必须是稳定的。由此可见, “稳、准、快”显然是系统的主要动态性能。

1.4 MATLAB 语言简介

MATLAB 是美国 Mathworks 公司开发的大型数学计算软件, 它提供了强大的矩阵处理和绘图功能, 并具有界面友好的用户环境。MATLAB 带有一些强大的具有特殊功能的工具箱, 几乎涵盖了所有工业、电子、医疗、建筑等领域, 已经成为国际上最流行的软件之一。现在的 MATLAB 已不仅仅是一个“矩阵实验室” (matrix laboratory), 而且已经成为一种实用的计算机高级编程语言, 是工程技术人员的必备软件。

1.4.1 MATLAB 的系统界面

1.4.1.1 MATLAB 窗口

MATLAB 具有强大的编程功能和易操作的交互式计算环境。MATLAB 语言

被认为是一种解释性语言,在其工作空间(Workspace),用户可以在 MATLAB 的命令窗口中键入一个命令,就可以直接进行数字运算,也可以应用 MATLAB 语言编写应用程序、运行程序及跟踪调试程序。MATLAB 软件会对命令和程序的各项语句进行翻译,然后在 MATLAB 环境中对它进行处理,最后返回结果。

启动 MATLAB 6.5 后,出现 MATLAB 的系统界面如图 1.9 所示。MATLAB 系统界面的最上面是“MATLAB”标题栏,标题栏下面是条形菜单栏,菜单栏下面是工具栏按钮与设置当前目录(Current Directory)的弹出式菜单框及其右侧的查看目录树的按钮。在工具栏下面的大窗口就是 MATLAB 的主窗口,在大窗口里设置有四个小窗:工作空间(Workspace)、当前目录(Current Directory)、命令历史(Command History)与命令窗口(Command Window)。在最下面是“Start”开始按钮。命令窗口是用户和 MATLAB 进行交互的主要场所。

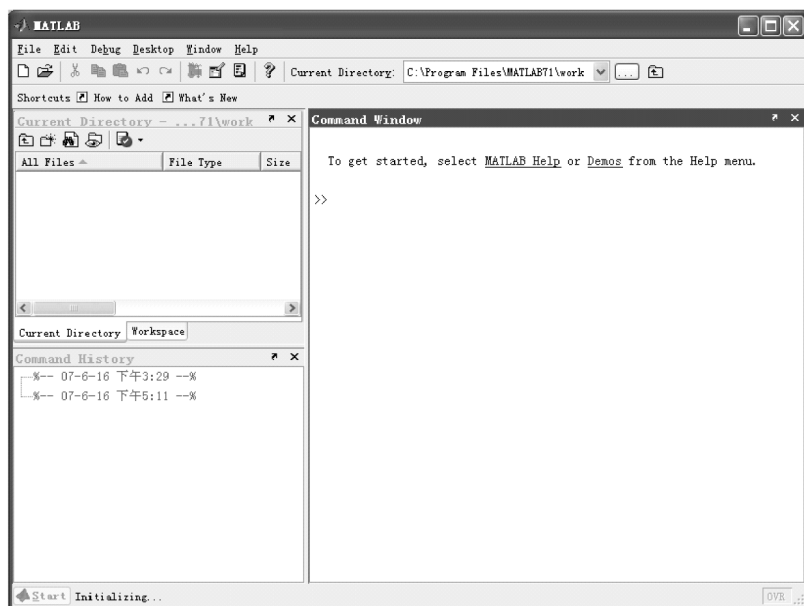


图 1.9 MATLAB 的系统界面

MATLAB 命令窗口的最上面两行是系统初始提示信息,第三行就会出现 MATLAB 环境提示符号“>>”和光标位置符。在命令窗口中,命令的执行类似 DOS 的行命令形式。从提示符处输入命令,按回车键后执行命令,并且一行可以输入多条命令,各命令用逗号或分号隔开。逗号告诉 MATLAB 显示结果,分号禁止显示。

如果一个表达式一行写不下, MATLAB 允许续行,键入“...”回车,即可在下一行继续输入。注意在“...”前面留有空格,并且变量名不能分成两行,注释语句不能

续行。在%后的所有文字为注释,执行 MATLAB 时忽略%及其后的文本。用户可以在任何时刻用【Ctrl+C】中断 MATLAB 的运行。

1.4.1.2 MATLAB 常用的操作命令

MATLAB 对命令窗口可以通过菜单命令进行操作,也可以通过键盘输入控制指令。窗口中常用的操作指令如表 1.1 所示。

表 1.1 命令窗口中常用的操作命令

指令名称	指令功能
cd	改变当前工作目录
clear	清除工作空间中的所有变量和函数
clc	擦除 MATLAB 命令窗口中所有显示的内容
clf	擦除 MATLAB 当前窗口中的图形
dir	列出指定目录下的文件和子目录清单
disp	在运行中显示变量或文字内容
echo	控制运行文字指令是否显示
hold	控制当前图形窗口对象是否被刷新
pack	收集内存碎块以扩大内存空间
quit	关闭并退出 MATLAB
type	显示所指定文件的全部内容

在 MATLAB 命令窗口里,当输入、编辑并运行命令、函数与程序进行各种不同类型的数学运算时,有很多控制键和方向键可用于语句行的编辑。表 1.2 列出了 MATLAB 命令窗口中的快捷键。

表 1.2 MATLAB 命令窗口中的快捷键

键盘操作	功能
↑	Ctrl+P 调出前一个命令行
↓	Ctrl+N 调出后一个命令行
←	Ctrl+B 光标左移一个字符
→	Ctrl+F 光标右移一个字符
	Ctrl+R 光标右移一个单词
	Ctrl+I 光标左移一个单词
Home	Ctrl+A 光标移至行首
End	Ctrl+E 光标移至行尾
Esc	Ctrl+U 清除当前行
Del	Ctrl+D 清除光标所在位置后的字符
Backspace	Ctrl+H 清除光标所在位置前的字符
	Ctrl+K 删至行尾

1.4.2 MATLAB 数学运算

1.4.2.1 算术运算

运用 MATLAB 可以完成一般常用的加(+)、减(-)、乘(*)、除(/)、幂次(^)等数学运算,最简单的方法是在 MATLAB 命令窗口(Command Window)内的提示符号(>>)之后输入表达式,并按下【Enter】键即可。例

```
>> (25*2+8.5)/5  
ans=  
11.7000
```

MATLAB 会将运算结果直接存入默认变量 ans,代表 MATLAB 运算后的答案(answer),并在屏幕上显示其运算结果的数值。

若不想让 MATLAB 每次都显示运算结果,只需在表达式最后加上分号(;)即可,例

```
>> (25*2+8.5)/5;
```

在上例中,由于表达式后面加入了分号,因此 MATLAB 只会将运算结果储存在默认变量 ans 内,不会显示于屏幕上;在需要时取用或显示此运算结果,可直接输入变量 ans,例(接上例)

```
>>ans  
ans=  
11.7000
```

使用者也可将运算结果储存于使用者自己设定的变量 x 内,例

```
>>x=(25*2+8.5)/5;  
x=  
11.7000
```

MATLAB 变量命名遵循以下规则:

- (1) 第一个字母必须是英文字母。
- (2) 字母间不可留空格。
- (3) 最多只能有 31 个字母, MATLAB 会忽略多余字母。

(4) MATLAB 在使用变量时,不需预先经过变量声明(variable declaration)的程序,而且所有数值变量均以默认的双精度数据类型(double)储存。

MATLAB 会将所有在百分比符号(%)之后的文字视为程序的注解(comments),例

```
>>x=(25*2+8.5)/5; %将运算结果储存于变量 x 内,但不显示于屏幕  
>>y=x^2 %将运算结果储存于变量 y 内,并显示于屏幕  
y=
```

21.7000

在上例中,百分比符号之后的文字会被 MATLAB 忽略不执行,但它的使用可提高 MATLAB 程序的可读性。

MATLAB 可同时执行以逗号(,)或分号(;)隔开的几个表达式,例

```
>>x=sin(pi/3);y=x^2;z=y*10,
```

```
z=
```

7.5000

若一个数学表达式太长,可用三个句点(...)将其延伸到下一行,例

```
>>z=10*sin(pi/3)*...
```

```
sin(pi/3)
```

```
z=
```

7.5000

1.4.2.2 复数运算

在 MATLAB 中,复数可以用两种形式表示。

(1) 直接使用 MATLAB 的缺省值 $i=j=\sqrt{-1}$ 来表示虚部。例

$1+2i$ 或 $1-3j$

(2) 可以在表达式中直接对负数开根号。例

```
a=4*(2+sqrt(-1)^2),MATLAB 执行结果为
```

```
a=
```

8.0000+8.0000i

1.4.2.3 数组与矩阵的运算

矩阵是 MATLAB 的核心, MATLAB 的所有数据都以矩阵形式存储。在 MATLAB 中,矩阵和数组这两个术语并无严格的区分,可以混用。准确地讲,矩阵是一个二维的实数或复数的数组。例

```
>>s=[2 3 6 2]; %注意[]的使用及各数字间的空格
```

```
>>t=2*s+1
```

```
t=
```

5 7 13 5

在上例中, MATLAB 使用中括号[], 来建立一个数组[2 3 6 2](或是 1×4 大小的矩阵[2 3 6 2]), 将其储存在变量 s 中, 再对其进行运算产生另一新的数组[5 7 13 5], 并将其结果储存在变量 t 内。

注意: $s=[2 \ 3 \ 6 \ 2]$ 与 $s=[2,3,6,2]$ 的效果是一样的。

MATLAB 也可取出数组中的一个元素或一部分来做运算, 例(接上例)

```
>>t(3)=10
```

%将数组 t 的第三个元素更改为 10

```

t=
    5    7   10    5
>>t(6)=12           %在数组 t 中加入第六个元素,其值为 12
t=
    5    7   10    5    0   12
>>t(4)=[ ]          %将数组 t 的第四个元素删除
t=
    5    7   10    0   12
>>s(2)*3+t(4)        %取出数组 s 的第二个元素和数组 t 的第四个元素作运算
ans=
     9

```

用类似上述建立数组的方法,可以建立 $m \times n$ 大小的矩阵(m 代表矩阵的行数, n 代表矩阵的列数),但必须在每一行结尾加上分号“;”,例

```

>>A=[1  2  3  4; 5  6  7  8; 9  10  11  12];    %建立 3 × 4 的矩阵 A
>>A
%显示矩阵 A 内容
A=
     1     2     3     4
     5     6     7     8
     9    10    11    12

```

同样,还可以对矩阵进行各种处理,例(接上例)

```

>>A(2,3)=5          %将矩阵 A 第二行、第三列的元素值,改变为 5
A=
     1     2     3     4
     5     6     5     8
     9    10    11    12
>>B=A(2,1:3)         %取出矩阵 A 第二行、第一列至第三列的元素值
%储存成一新矩阵 B
B=
     5     6     5
>>A=[A B']           %将矩阵 B 转置后,再以列向量并入矩阵 A
A=
     1     2     3     4     5
     5     6     5     8     6
     9    10    11    12     5
>>A(:,2)=[ ]         %删除矩阵 A 第二列(:,代表所有行,[ ]代表空矩阵)
A=
     1     3     4     5
     5     5     8     6

```

```
9 11 12 5
>>A=[A; 4 3 2 1] %在原矩阵 A 中,加入第四行
A=
1 3 4 5
5 5 8 6
9 11 12 5
4 3 2 1
```

1.4.2.4 MATLAB 常用数学函数

MATLAB 是一个科学计算软件,因此它支持很多数学函数,例

```
>>x=4;
>>y=abs(x) %取 x 的绝对值
y=
4
>>y=sin(x) %取 x 的正弦值
y=
-0.7568
```

MATLAB 有相当完整的数学函数及三角函数,如表 1.3 和表 1.4 所示。这些基本数学函数,也都可用于数组或矩阵,例

```
>>x=[4 2j 9];
>>y=sqrt(x) %对 x 开平方
y=
2.0000 1.0000+1.0000i 3.0000
```

表 1.3 MATLAB 常用数学函数

函数	说明	函数	说明
abs (x)	标量的绝对值或复数的幅值	fix (x)	对原点方向取紧邻整数
exp (x)	自然指数 e^x	ceil (x)	对 $+\infty$ 方向取紧邻整数
pow2 (x)	2 的指数 2^x	floor (x)	对 $-\infty$ 方向取紧邻整数
sqrt (x)	开平方	gcd (x , y)	整数 x 和 y 的最大公约数
log (x)	自然对数 $\ln (x)$	lcm (x , y)	整数 x 和 y 的最小公倍数
log2 (x)	以 2 为底的对数 $\log_2 (x)$	rem (x , y)	求 x 除以 y 的余数
log10 (x)	以 10 为底的对数 $\log_{10} (x)$	round (x)	四舍五入到最近的整数
angle (z)	复数 z 的相角	符号函数	
real (z)	复数 z 的实部	当 $x<0$ 时, $\text{sign} (x) = -1$	
imag (z)	复数 z 的虚部	sign(x)	当 $x=0$ 时, $\text{sign} (x) = 0$
conj (z)	复数 z 的共轭复数	当 $x>0$ 时, $\text{sign} (x) = 1$	

表 1.4 MATLAB 常用三角函数

函数	说明	函数	说明
$\sin(x)$	正弦函数	$\operatorname{asin}(x)$	反正弦函数
$\cos(x)$	余弦函数	$\operatorname{acos}(x)$	反余弦函数
$\tan(x)$	正切函数	$\operatorname{atan}(x)$	反正切函数
$\sinh(x)$	双曲正弦函数	$\operatorname{asinh}(x)$	反双曲正弦函数
$\cosh(x)$	双曲余弦函数	$\operatorname{acosh}(x)$	反双曲余弦函数
$\tanh(x)$	双曲正切函数	$\operatorname{atanh}(x)$	反双曲正切函数

在上例中, $\operatorname{sqrt}(x)$ 函数会对 x 的每一个元素进行开平方的运算。

另外, MATLAB 还有一些函数是特别针对数组设计的, 例

```
>>x=[1 2 3 0 12];
>>y=min(x)           %数组 x 的最小值
y=
0
```

计算数组和统计量的常用函数见表 1.5 所示。

表 1.5 计算数组元素统计量的常用函数

函数	说明	函数	说明
$\min(x)$	数组 x 的元素的最小值	$\operatorname{norm}(x)$	数组 x 的欧几里得(Euclidean)长度
$\max(x)$	数组 x 的元素的最大值	$\operatorname{sum}(x)$	数组 x 的元素总和
$\operatorname{mean}(x)$	数组 x 的元素平均值	$\operatorname{prod}(x)$	数组 x 的元素总乘积
$\operatorname{median}(x)$	数组 x 的元素的中位数	$\operatorname{cumsum}(x)$	数组 x 的累积元素总和
$\operatorname{std}(x)$	数组 x 的元素的标准差	$\operatorname{cumprod}(x)$	数组 x 的累积元素总乘积
$\operatorname{diff}(x)$	数组 x 的相邻元素的差	$\operatorname{dot}(x, y)$	数组 x 和 y 的内积
$\operatorname{sort}(x)$	对数组 x 的元素进行排序	$\operatorname{cross}(x, y)$	数组 x 和 y 的外积
$\operatorname{length}(x)$	数组 x 的元素个数		

以上这些函数是针对数组设计的, 无论输入是列向量或行向量, 都可返回正确的结果。若输入为矩阵时, 这些函数将输入矩阵看成是列向量的集合, 并选一对列向量进行运算, 例

```
>>x=[1 2 3; 4 5 6; 7 8 9];
>>y=median(x)         %x 每个列向量的中位数
y=
4 5 6
```

1.4.2.5 多项式运算

多项式运算是线性代数、线性系统分析中的重要内容。MATLAB 提供了多条

命令,可以进行多项式运算。

一个 n 次的多项式可以表示为

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

因此,在 MATLAB 中,可以用一个长度为 $n+1$ 的行向量来表示 $p(x)$ 为

$$p = [a_n, a_{n-1}, \cdots, a_1, a_0]$$

例如,可用 $p=[1,2,0,-5,6]$ 来表示一个 4 次多项式

$$p(x) = x^4 + 2x^3 - 5x + 6$$

在 MATLAB 的命令窗口内输入为

```
>>p=[1 2 0 -5 6];
```

(1) 求根及逆运算 roots 命令可以求解多项式 $p(x)$ 的根,求出的根按列向量存储。例如,求上面给出的多项式的根,则键入

```
>>r=roots(p)
```

```
r=-1.8647+1.3584i
```

```
-1.8647-1.3584i
```

```
0.8647+0.6161i
```

```
0.8647-0.6161i
```

poly 命令可以由根的列向量表示求多项式的系数,得到的多项式系数按行向量存储。例如对于上面求出的根,键入

```
>>p=poly(r)
```

得到

```
D=1.0000    2.0000    0.0000   -5.0000    6.0000
```

(2) 有理多项式在线性系统的 Fourier 变换、Laplace 变换和 z 变换中,经常用到有理多项式。MATLAB 提供了一些命令可以进行有理多项式的运算。MATLAB 中的有理多项式是由分子多项式和分母多项式表示的,可以用 residue 命令进行部分分式展开。该命令语法为

```
[r,p,k]=residue(num,den)
```

其中,num 和 den 分别表示分子和分母多项式的系数行向量。

分解的结果形式为

$$G(s) = \frac{\text{num}(s)}{\text{den}(s)} = \frac{r(1)}{s+p(1)} + \frac{r(2)}{s+p(2)} + \cdots + \frac{r(n)}{s+p(n)} + k(s)$$

式中,而 $k(s)$ 为常数项或纯微分项。

例如,对已知传递函数

$$G(s) = \frac{10(s+1)}{(s+1)(s^2+s+3)} = \frac{10s+30}{s^3+2s^2+4s+3}$$

进行部分分式分解,运算过程为

```
>>num=[10 30]; %定义分子多项式的系数
```

```

>>den=[1 2 4 3];      %定义分母多项式的系数
>>[r,p,k]=residue(num,den)  %进行部分分式展开
-3.3333 -4.0202i
-3.3333 +4.0202i
6.6667
p=
-0.5000+1.6583i
-0.5000 -1.6583i
-1.0000
k=
[]

```

其中, $k=[]$ 表示没有常值项。

结果表明, 传递函数 $G(s)$ 被分解为下面的部分分式

$$G(s) = \frac{\text{num}(s)}{\text{den}(s)} = \frac{-3.3333 - 4.0202i}{s + 0.5 - 1.6583i} + \frac{-3.3333 + 4.0202i}{s + 0.5 + 1.6583i} + \frac{6.6667}{s + 1}$$

注意, 该命令得到的低阶分式都是一阶形式, 分子上是常值。显然, 由于原传递函数有复特征值, 利用该命令得到的展开式不是常用的实数形式。当然, 可以利用多项式的乘法命令进行通分得到实数形式。

1.4.2.6 MATLAB 中的永久常数

MATLAB 有一些永久常数(permanent constants), 如表 1.6 所示。虽然在工作空间中看不到, 但使用者可直接取用, 例

```

>>pi
ans=
3.1416

```

表 1.6 MATLAB 永久常数

常 数	说 明
i 或 j	基本虚数单位(即 $i^2 = -1$)
eps	系统的浮点(floating-point)精确度
inf	无限大, 如 $1/0$
Nan 或 NaN	非数值(not a number.), 如 $0/0$
pi	圆周率($=3.1415926\cdots$)
Realmax	系统所能表示的最大数值
Realmin	系统所能表示的最小数值
Nargin	函数的输入变量个数
Nargout	函数的输出变量个数

1.4.3 MATLAB 绘图

MATLAB 不但擅长与矩阵相关的各种数值运算,也具有非常强大的绘图功能,特别适用于各种科学计算可视化(scientific visualization)。

1.4.3.1 基本绘图命令 plot

MATLAB 的 plot 是最基本的绘图命令,可以对一组 x 坐标及相对应的 y 坐标进行描点绘图,例

```
>>x=linspace(0,2*pi);      %在 0 到 2 $\pi$  间,等分取 100 个点
>>y=sin(x);                %计算 x 的正弦函数值
>>plot(x,y);               %进行二维平面描点绘图
```

各命令执行的结果如图 1.10 所示的图形。

在上例中,linspace(0,2*pi)产生从 0 到 2π 且长度为 100(默认值)的矢量 x , y 则是相对应的 y 坐标值,plot(x , y)则可对这 100 个二维平面上的点进行描点绘图。

若要利用 plot 命令一次画出多条曲线,可将 x 及 y 坐标依次送入 plot 命令即可,例

```
>>x=linspace(0,2*pi);
>>plot(x,sin(x),'-',x,cos(x),'-',x,sin(x)+cos(x),'-');
```

各命令执行的结果如图 1.11 所示的图形。

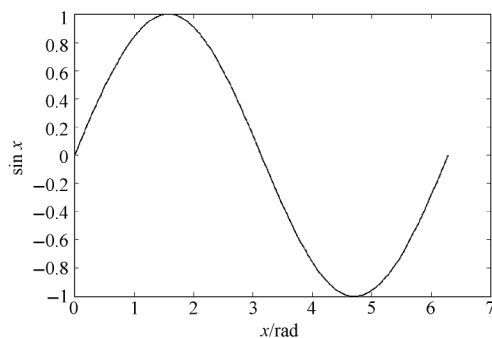


图 1.10 二维平面扫描绘图

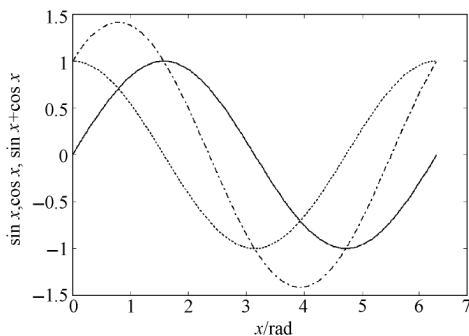


图 1.11 多条曲线描点

MATLAB 在绘制多条曲线时,会自动转换曲线颜色,以利于分辨(也可由使用者自行设定曲线颜色及其他相关属性)。还可以不同的标志(marker)来绘图,可输入

```
>>x=linspace(0,2*pi);
>>plot(x,sin(x),'o',x,cos(x),'x',x,sin(x)+cos(x),'*');
```

各命令执行的结果如图 1.12 所示的图形。

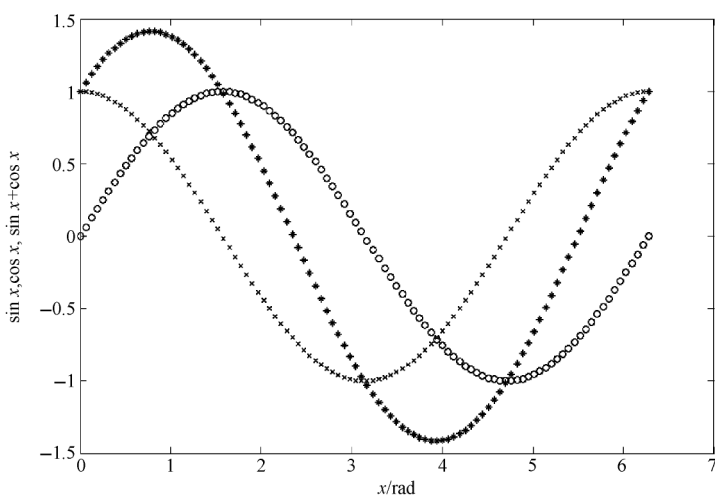


图 1.12 以不同的标志绘图

MATLAB 的基本二维绘图命令,可参见表 1.7 所示。

表 1.7 MATLAB 的基本二维绘图命令

命令	说明
plot	x 轴和 y 轴均为线性刻度(linear scale)
loglog	x 轴和 y 轴均为对数刻度(logarithmic scale)
semilog x	x 轴为对数刻度, y 轴为线性刻度
semilog y	y 轴为对数刻度, x 轴为线性刻度
ploty y	画出两个刻度不同的 y 轴

例如,若要使 x 轴为对数刻度,来对正弦函数绘图,可如下进行

```
>>x=linspace(0,8*pi);           %在 0 到 8 $\pi$ 间,等分取 100 个点
>>semilogx(x,sin(x));           % $x$  轴为对数刻度,并对其正弦函数绘图
```

各命令执行的结果如图 1.13 所示的图形。

1.4.3.2 图形的修饰

plot 命令除了接受 x 及 y 坐标外,还可以接受一个字符串输入,用以控制曲线的颜色、格式及标志,其指令调用格式为

```
plot(x,y,'CLM')
```

其中,C 代表曲线的颜色(colors),L 代表曲线的格式(line styles),M 代表曲线所用的标志(markers)。例如,若要用黑色点线画出正弦波,并在每一数据点上画一个

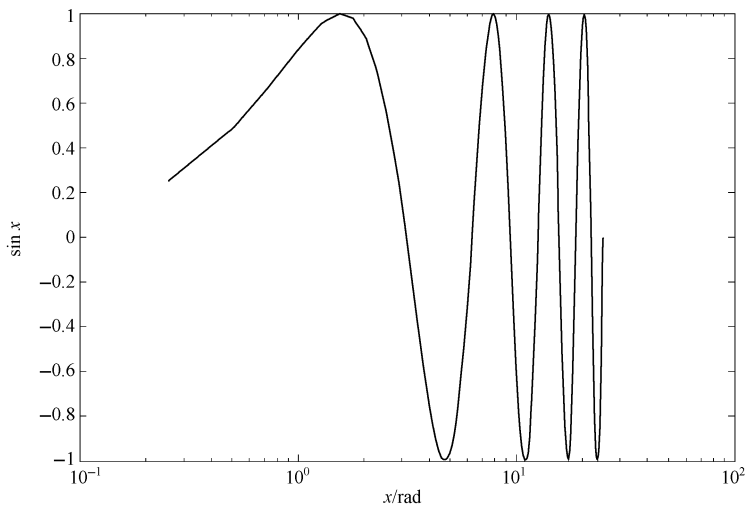


图 1.13 x 轴为对数刻度的正弦函数图

小菱形,可输入

```
>>x=0:0.5:4*pi;           % x 向量的起始与结束元素为 0 及 4π
                             % 0.5 为各元素相差值(间隔为 0.5)

>>y=sin(x);
>>plot(x,y,'k:diamond')    %其中“k”代表黑色,“:”代表点线
                             %而“diamond”则指定菱形为曲线的标志
```

各命令执行的结果如图 1.14 所示的图形。

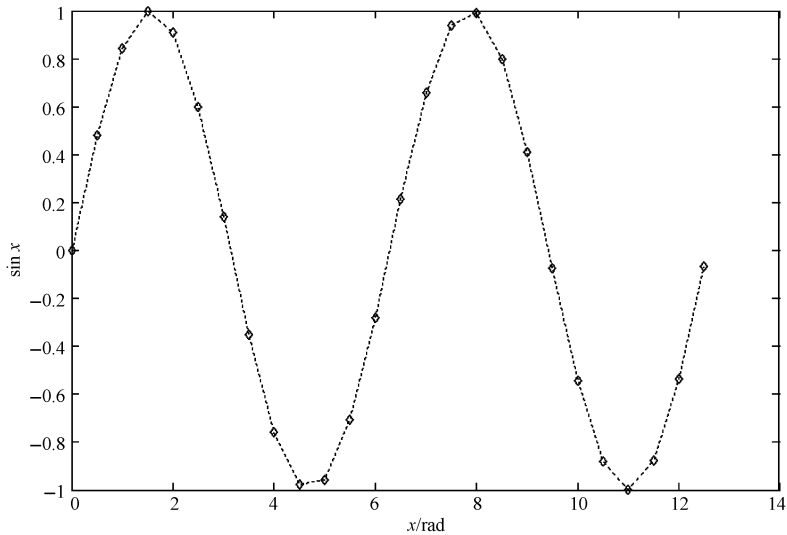


图 1.14 用黑色点线画出正弦波并在数据点画上小菱形

表 1.8 plot 命令中曲线颜色选项参考表

plot 命令	曲线颜色	RGB 值
b	蓝色 (blue)	(0,0,1)
c	青蓝色 (cyah)	(0,1,1)
g	绿色 (green)	(0,1,0)
k	黑色 (black)	(0,0,0)
m	紫红色 (magenta)	(1,0,1)
r	红色 (red)	(1,0,0)
w	白色 (white)	(1,1,1)
y	黄色 (yellow)	(1,1,0)

plot 命令	曲线格式	plot 命令	曲线格式
-	实线(默认值)	:	点线
--	虚线	-	点画线
o	圆形	>	朝右三角形
+	十字形	<	朝左三角形
×	叉号	square/s	方形
*	星号	diamond/d	菱形
.	点号	pentagram/p	五角星形
^	朝上三角形	hexagram/h	六角星形
v	朝下三角形	none	无符号(默认值)

```
例
>>x=0:0.2:12;
>>plot(x,sin(x),'-')
>>hold on
>>plot(x,1.5 * cos(x),'-')
```

以上程序中使用了两条 plot 命令绘制出两条曲线,如图 1.15 所示。

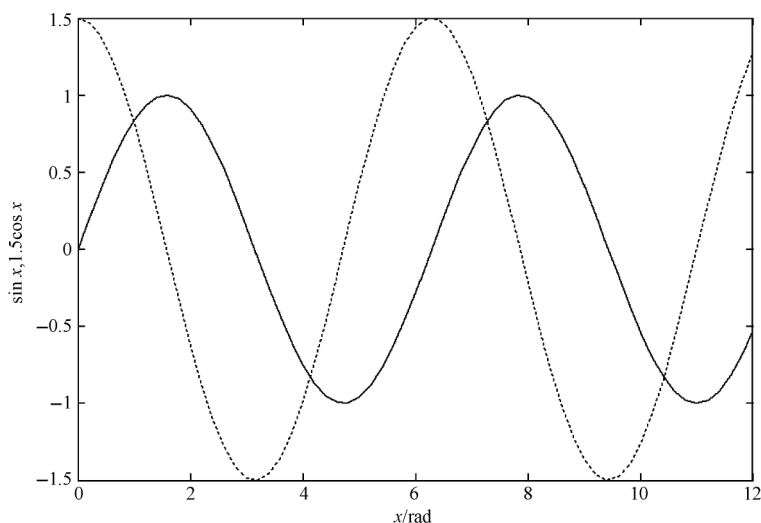


图 1.15 利用 hold 命令绘出的两条曲线

1.4.3.3 坐标轴的定制

一般而言,plot 命令会根据所给的坐标点来自动决定图轴的范围,但是也可以使用 axis 命令来指定,其指令调用格式为

```
axis([xmin,xmax,ymin,ymax])
```

其中,xmin 和 xmax 指定 x 轴的最小和最大值,ymin 和 ymax 则指定 y 轴的最小和最大值。如果要画出正弦波在 y 轴介于 0 和 1 的部分,可输入

```
>>x=0:0.1:4*pi;           % x 矢量的起始与结束元素为 0 及 4π,
                           % 0.1 为各元素相差值
```

```
>>y=sin(x);
```

```
>>plot(x,y);
```

```
>>axis([-inf,inf,0,1]);    %画出正弦波 y 轴介于 0 和 1 的部分
```

各命令执行的结果如图 1.16 所示的图形。

其中,-inf 及 inf 并不是代表“无穷大”,而是代表以数据点(上例中即是 z 轴的数据点)的最小值和最大值来取代,因此上述 axis 命令等效于

```
axis([min(x),max(x),0,1])。
```

1.4.3.4 图形窗口分割

如果要在一个窗口产生多个图形(即图轴),可在 plot 命令之前加上 subplot,其一般形式为 subplot(m, n, p),表示将窗口划分为 $m \times n$ 个区域,而下一个 plot 命令则绘图于第 p 个区域,其中 p 的算法为由左至右,一行一行算起。如果在一