

自动化科学作为一门学科起源于 20 世纪初,自动化科学与技术的基础理论来自于物理学等自然科学和数学、系统科学、社会科学等基础科学^[1],在现代科学技术的发展中有着重要的地位,起着重要的作用。在第 40 届 IEEE 决策与控制年会(CDC)全会开篇报告中美国学者 John Doyle 教授引用国际著名学者、哈佛大学的何毓琦(Larry Yu-Chi Ho)教授的一个振奋人心的新观点:“控制将是 21 世纪的物理学(Control will be the physics of the 21st century)”^[2]。

自动化科学的进展是与控制理论的发展和完善分不开的。控制理论发展初期,为控制系统设计控制器一般采用简单的试凑方法,随着控制理论的发展和计算机技术的进步,控制系统的计算机辅助设计技术作为一门学科也发展起来了。本章首先介绍控制系统计算机辅助设计领域的形成与发展情况,然后介绍与之密切相关的计算机软件和语言,特别是 MATLAB 语言的发展概况,还将对本书的基本框架做一个简要的概述,以便读者更好地学习本书的内容。

1.1 控制系统计算机辅助设计技术的发展综述

早期的控制系统设计可以由纸笔等工具容易地计算出来,如 Ziegler 与 Nichols 于 1942 年提出的 PID 经验公式^[3]就可以十分容易地设计出来。随着控制理论的迅速发展,控制的效果要求越来越高,控制算法越来越复杂,控制器的设计也越来越困难,这样光利用纸笔以及计算器等简单的运算工具难以达到预期的效果,加之计算机技术的迅速发展,于是很自然地出现了控制系统的计算机辅助设计(computer-aided control systems design, CACSD)技术。

控制系统的计算机辅助设计技术的发展目前已达到了相当高的水平,并一直受到控制界的普遍重视。早在 1982 年 12 月和 1984 年 12 月,控制系统领域在国际上最权威的 IEEE 控制系统学会(Control

Systems Society, CSS) 的控制系统杂志 (Control Systems Magazine) 和 IEEE 学会的科研报告集 (Proceedings of IEEE) 分别第一次出版了关于 CACSD 的专刊^[4,5], 美国著名学者 Jamshidi 与 Herget 分别于 1985 年和 1992 年出版了两本著作来展示 CACSD 领域的最新进展^[6,7]。在如国际自动控制联合会世界大会 (IFAC World Congress)、美国控制会议 (American Control Conference, ACC) 及 IEEE 的决策与控制会议 (Conference on Decision and Control, CDC) 等各种国际控制界的重要学术会议上都有有关 CACSD 的专题会议及各种研讨会, 可见该领域的发展是异常迅速的。控制系统计算机辅助设计又常常称做计算机辅助控制系统工程 (computer-aided control systems engineering, CACSE)。

近三十年来, 随着计算机技术的飞速发展, 出现了很多优秀的计算机应用软件, 在控制系统的计算机辅助设计领域更是如此, 各类 CACSD 软件频繁出现且种类繁多, 有的是用 Fortran 语言编写的软件包, 有的是人机交互式软件系统, 还有专用的仿真语言, 在国际控制界广泛使用的这类软件就有几十种之多。MATLAB 语言出现以来, 就深受控制领域学生和研究者的欢迎, 已经成为控制界最流行、最有影响的通用计算机语言, 成为控制界学者的首选。

国内外在介绍控制系统计算机辅助设计的早期教材中, 都采用通用的计算机语言如 BASIC 语言^[6,8]、Fortran 语言^[9] 或 C 语言作为辅助的计算机语言。随着计算机语言的发展和日益普及, 特别是代表科学运算领域最新成果的 MATLAB 语言的出现, 较新的著作中, 很多都采用 MATLAB 作为主要程序设计语言来介绍控制系统计算机辅助设计的算法^[10~17], 在新型的自动控制理论教材中也有这样的趋势^[18~20]。采用新型的计算机语言为主线介绍控制系统计算机辅助设计的理论与方法, 可以使读者将主要精力集中在控制系统理论和方法上, 而不是将主要精力花费在没有太大价值的底层重复性机械性劳动上, 这样可以对控制系统计算机辅助设计技术有较好的整体了解, 避免“只见树木, 不见森林”的认识偏差, 提高控制器设计的效率和可靠性。

子曰: “工欲善其事, 必先利其器”。跟踪国际最先进的 CACSD 软件环境及发展, 以当前国际上最流行的 CACSD 软件环境 MATLAB 为基本出发点来系统地介绍控制系统计算机辅助设计技术及软件实现, 从而大大提高 CACSD 算法研究与实际应用的水平和可靠性, 这是本书的一个主要目的。

1.2 控制系统计算机辅助设计语言环境综述

1973 年美国学者 Melsa 教授和 Jones 博士出版了一本专著^[9], 书中给出了许多当时流行的控制系统计算机辅助分析与设计的源程序, 包括求取系统的根轨迹、频域响应、时域响应以及各种控制系统设计的子程序如 Luenberger 观测器、Kalman 滤波等。瑞典 Lund 工学院 Karl Åström 教授主持开发的一套交互式 CACSD 软件 INTRAC (IDPAC、MODPAC、SYNPAC、POLPAC 等, 以及仿

真语言 SIMNON)^[21], 其中的 SIMNON 仿真语言要求用户依照它所提供的语句编写一个描述系统的程序, 然后才可以对控制系统进行仿真。日本的古田胜久 (Katsuhisa Furuta) 教授主持开发的 DPACS-F 软件^[22], 在处理多变量系统的分析和设计上还是很有特色的。在国际上流行的仿真语言 ACSL、CSMP、TSIM、ESL 等也同样要求用户编写模型程序, 并提供了大量的模型模块。在这一阶段还出现了很多的专用程序, 如英国剑桥大学推出的线性系统分析与设计软件 CLADP (Cambridge linear analysis and design programs)^[23,24]与美国 NASA Langley 研究中心的 Armstrong 开发的线性二次型最优控制器设计的 ORACLS (optimal regulator algorithms for the control of linear systems)^[25]等。

1980 年美国学者 Cleve Moler 等人推出的交互式 MATLAB 语言逐渐受到了控制界研究者的普遍重视, 从而陆续出现了许多专门用于控制理论及其 CAD 的工具箱, 为控制系统的分析与设计提供了极大的方便, 也为研究者开发测试新的方法提供了强有力的工具。图形交互式的模型输入计算机仿真环境 Simulink 的出现为 MATLAB 应用的进一步推广起到了积极性的推动作用。现在, MATLAB 已经风靡全世界, 成为控制系统仿真与计算机辅助设计领域最普及也是最受欢迎的首选计算机语言。

在 MATLAB 迅速发展的同时, 很多软件开发针对控制系统领域的实际问题开发了专用的 CACSD 计算机辅助设计软件, 如美国系统控制技术公司 (Systems Control Technology Inc.) 的 Jack Little 等人研制的 CTRL-C^[26], Boeing 公司的 EASY 5 及 EASY5x, Integrated Systems 公司的 Matrix-X 及 Xmath, Systems Technology Incorporated 公司的 CC 程序, Visual Simulation 公司的 VisSim、O-Matrix, 韩国汉城国立大学权旭铉教授主持开发的 CemTool 以及现在仍作为免费软件的 Octave^[27]、Scilab^[28]等, 虽然其中很多软件是并行于 MATLAB 而独立开发的, 但或多或少都会从这些软件的语句结构或使用方法中看出明显受到 MATLAB 影响的痕迹, 所以说, 从国际上最流行的 MATLAB 出发来介绍控制系统的计算机辅助设计技术是再合适不过的了。这就是本书在众多 CACSD 软件中挑选 MATLAB 作为基本语言的一个最主要的原因。

国际上控制系统计算机辅助设计软件的发展大致分为几个阶段: 软件包阶段、交互式语言阶段及当前的面向对象的程序环境阶段^[29]。

在早期的工作中, CACSD 主要集中在软件包的编写上, 如前面提及的 Melsa 和 Jones 的著作。从数值算法的角度上也出现了一些著名的软件包, 如美国的基于特征值的软件包 EISPACK^[30,31]和线性代数软件包 LINPACK^[32], 英国牛津数值算法研究组 (Numerical Algorithm Group) 开发的 NAG 软件包^[33]及文献^[34]中给出的声誉颇高的数值算法程序集等, 在 CACSD 领域的经典软件包作品有英国 Kingston Polytechnic 控制系统研究组开发的 SLICE (subroutine library in control engineering) 软件包^[35], 前面提及的 DPACS-F、ORACLS 等。这些软件

包大都是用 Fortran 语言编写的源程序组成的,给使用者提供了较好的接口,但和 MATLAB 相比,调用方法和使用明显显得麻烦、不便。

以著名的 EISPACK 软件包为例,若想求出 N 阶实矩阵 A 的全部特征值(用 W_R 、 W_I 数组分别表示其实虚部)和对应的特征向量矩阵 Z ,则 EISPACK 软件包给出的子程序建议调用路径为^[30]:

```
CALL BALANC(NM,N,A,IS1,IS2,FV1)
CALL ELMHES(NM,N,IS1,IS2,A,IV1)
CALL ELTRAN(NM,N,IS1,IS2,A,IV1,Z)
CALL HQR2(NM,N,IS1,IS2,A,WR,WI,Z,IERR)
IF (IERR.EQ.0) GOTO 99999
CALL BALBAK(NM,N,IS1,IS2,FV1,N,Z)
```

由上面的叙述可以看出,要求取矩阵的特征值和特征向量,首先要给一些数组和变量依据 EISPACK 的格式作出定义和赋值并编写出主程序,再经过编译和连接过程形成可执行文件,最后才能得出所需的结果。用软件包的形式来编写程序有如下的缺点:

① **使用不方便**。对不是很精通 EISPACK 的用户来说,直接利用软件包来编写程序是相当困难的,也是相当容易出错的,其中一个子程序调用格式发生微小的错误将可能导致最终得出错误的结果。

② **调用过程繁琐**。首先需要编写主程序,确定对软件包的调用过程,再经过必要的编译和连接过程,有时还要花大量的时间去调试程序以保证其正确性,而不是想得出什么马上就可以得出的。

③ **执行程序过多**。想求解一个特定的问题就需要编写一个专门的程序,并形成可执行文件,如果要求解的问题很多,就需要在计算机硬盘上同时保留很多这样的可执行文件,这样计算机磁盘空间的利用不是很经济。

④ **不利于传递数据**。通过软件包调用方式针对每个具体问题就能形成一个孤立的可执行文件,在一个程序中产生的数据无法传入另一个程序,更无法使几个程序同时执行来解决所关心的问题。

⑤ **维数指定困难**。在 CACSD 中最重要的变量是矩阵,如果要求解的问题维数较低,则形成的程序就不能用于求解高阶问题,例如文献 [9] 中的程序均定为 10 阶。所以有时为使得程序通用,往往将维数设置得很大,这样在解小规模问题时会出现空间的浪费,而大规模问题仍然求解不了。在优秀的软件中往往需要动态地进行矩阵定维。

此外,这里介绍的大多数早期软件包都是用 Fortran 语言编写的,由于众所周知的原因,以前 Fortran 语言绘图并不是轻而易举的事情,这就需要再调用相应的软件包来做进一步处理,在绘图方面比较实用和流行的软件包是 GINO-F^[36],但这种软件包只给出绘图的基本子程序,所以要绘制较满意的图形需要用户自己用这些低级命令去编写出合适的绘图子程序来。

英国剑桥大学学者 John Edmunds 和 Jan Maciejowski 等人开发的 CLADP 在控制界享有盛誉,它包括了多变量系统分析与设计的多种方法,其中有 Nyquist 类以及特征轨迹等多变量频域设计方法,也有线性二次型 Gauss 控制器 (LQG) 与 Kalman 滤波等时域设计方法,还可以处理时间延迟及分布系统等非有理问题。日本东京工学院的古田胜久教授主持开发的 DPACS-F 软件是用 Fortran 语言编写的,它可以由状态空间和频域方法来分析多变量线性系统,并可以由极点配置和 LQG 等方法来设计控制器,此外还可以进行多变量系统辨识等工作。NASA 的 Armstrong 教授编写的 ORACLS 则是一个十分专用的软件,它可以用于多变量系统的 LQG 设计,该软件也是用 Fortran 语言编写的。

20 世纪 70 年代末期和 80 年代初期出现了很多实用的具有良好人机交互功能的软件, MATLAB 就是其中的一个成功的范例,此外前面提及的 INTRAC 和 CTRL-C 等也是优秀的人机交互式软件。

正因为存在多种多样的 CACSD 软件,而它们之间又各有所长,所以在 CACSD 技术的发展过程中曾有过几次将若干常用软件集成在一起的尝试,例如 1984 年前后美国学者 Spang III 教授曾将当时流行的 SIMNON、CLADP、IDPAC 及他自己研制的 SSDP (state space design program) 集成在一起,形成了一个强大的软件^[37],各个组成软件之间是靠读写文件的方式来传递数据的,这多少可以解决前面提及的程序之间不能传递数据的弊病。1986 年前后由英国科学与工程委员会 (SERC) 资助, Howard H Rosenbrock 教授和 Neil Munro 教授主持的、英国多所大学和研究机构参与的 ECSTASY (environment for control system theory and synthesis) 软件环境的开发项目^[38],在该软件中试图将流行的新一代软件如 MATLAB、ACSL、TSIM 甚至当时刚出现的 Mathematica^[39] 等集成到一个框架之下,该软件还可以同时自动采用 L^AT_EX^[40] 和 FrameMaker 等来输出专业的排版结果,并取得了一些成效。各个软件之间的数据传递是通过数据库来实现的, ECSTASY 定义了方便实用的 CACSD 新命令,比当时的 MATLAB 更为简洁。ECSTASY 这样的软件是一个有益的尝试,但该软件当时只可以在 SUN 工作站上运行,并没有考虑 PC 的兼容性。

依作者之见,这些单纯集成出来的软件并不是很成功的,因为它们并没有达到预期的效果。事实上,从那以后每个软件的功能都有了明显的改善, MATLAB 语言有了自己的仿真功能,其仿真工具 Simulink 从某种意义上讲功能和接口远远优于 ACSL, MATLAB 和 Mathematica 之间也有了较好的接口, MATLAB 的符号运算工具箱也可以进行解析推导,它们的优势可以得到充分地互补。

我国较有影响的控制系统仿真与计算机辅助设计成果是中科院系统科学研究所韩京清研究员等主持的国家自然科学基金重大项目开发的 CADCSO 软件^[41]、清华大学孙增圻、袁曾任教授的著作和程序^[8] 与北京化工学院吴重光、沈承林教授的著作和程序^[42] 等。

1.3 仿真软件的发展概况

从前面提及的软件包的局限性看,直接调用它们进行系统仿真将有较大的困难,因为要掌握这些函数的接口是一件相当复杂的事,准确调用它们将更难;此外,有的软件包函数调用直接得出的结果可信度也不是很高,因为软件包的质量和水平参差不齐。

抛弃成型的软件包另起炉灶自己编写程序也不是很现实,毕竟在成型软件包中包含有很多同行专家的心血,有时自己从头编写程序很难达到这样的效果,所以必须采用经过验证且信誉著称的高水平软件包或计算机语言来进行仿真研究。

仿真技术引起该领域各国学者、专家们的重视,建立起国际的仿真委员会(Simulation Councils Inc., SCi),该委员会于1967年通过了仿真语言规范。仿真语言CSMP(computer simulation modelling program)应该属于建立在该标准上的最早的专用仿真语言。中科院沈阳自动化研究所在1988年推出了该语言的推广版本—CSMP-C。

20世纪80年代初期,美国 Mitchell and Gauthier Associates 公司推出了符合该标准的著名连续系统仿真语言 ACSL(advanced continuous simulation language)^[43],该语言出现后,由于其功能较强大,并有一些系统分析的功能,很快就在仿真领域占据了主导地位。

和 ACSL 大致同时产生的还有瑞典 Lund 工学院 Karl Åström 教授主持开发的 SIMNON,英国 Salford 大学的 ESL^[44]等,这些语言的编程语句结构也是很类似的,因为它们所依据的标准都是相同的。

计算机代数系统是在本领域中又一个吸引人的主题,而解决数学问题解析计算又是 C 语言直接应用的难点。于是国际上很多学者在研究、开发高质量的计算机代数系统。早期 IBM 公司开发的 muMATH^[45]和 REDUCE^[46]等软件为解决这样的问题提出了新的思路。后来出现的 Maple 和 Mathematica 逐渐占领了计算机代数系统的市场,成为比较成功的实用工具。

早期的 Mathematica 可以和 MATLAB 语言交互信息,比如通过一个称为 MathLink 的软件接口就可以很容易地完成这样的任务。为了解决计算机代数问题, MATLAB 语言的开发者—美国 MathWorks 公司也研制开发了符号运算工具箱(Symbolic Math Toolbox),该工具箱将 Maple 语言的内核作为 MATLAB 符号运算的引擎,使得二者能更好地结合起来。

这些软件和语言还是很昂贵的,所以有人更倾向于采用免费的,但编程结构类似于 MATLAB 的计算机语言,如 Octave^[27]和 Scilab 等, Scilab 配套的 Scicos 也支持基于框图的建模与仿真方法,这些软件的全部源程序也是公开的,有较高的透明度,但目前它们的功能已经无法与越来越强大的 MATLAB 语言相比。

系统仿真领域有很多自己的特性,如果能选择一种能反映当今系统仿真领域最高水平,也是最实用的软件或语言介绍仿真技术,使得读者能直接采用该语言解

决自己的问题,将是很有意义的。实践证明, MATLAB 就是这样的仿真软件,由于它本身卓越的功能,已经使得它成为自动控制、航空航天、汽车设计等诸多领域仿真的首选语言。所以在本书中将介绍基于 MATLAB/Simulink 的控制系统仿真与设计方法及其应用。

1.4 MATLAB/Simulink 与 CACSD 工具箱

MATLAB 语言的首创者 Cleve Moler 教授在数值分析,特别是在数值线性代数的领域中很有影响^[30~32, 47~49]。他曾在密西根大学、斯坦福大学和新墨西哥大学任数学与计算机科学教授。1980 年前后,时任新墨西哥大学计算机系主任的 Moler 教授在讲授线性代数课程时,发现了用其他高级语言编程极为不便,便构思并开发了 MATLAB (MATrix LABoratory, 即矩阵实验室)^[50], 这一软件利用了他参与研制的、在国际上颇有影响的 EISPACK^[31] (基于特征值计算的软件包) 和 LINPACK^[32] (线性代数软件包) 两大软件包中可靠的子程序,用 Fortran 语言编写了集命令翻译、科学计算于一身的一套交互式软件系统。

所谓交互式语言,是指用户给出一条命令,立即就可以得出该命令的结果。该语言无须像 C 和 Fortran 语言那样,首先要求使用者去编写源程序,然后对之进行编译、连接,最终形成可执行文件。这无疑会给使用者带来极大的方便。在 MATLAB 下,矩阵的运算变得异常容易,所以它一出现就广受欢迎,这一系统逐渐发展、完善,逐步走向成熟,形成了今天的模样。

早期的 MATLAB 只能作矩阵运算;绘图也只能用极其原始的方法,即用星号描点的形式画图;内部函数也只提供了几十个。但即使其当时的功能十分简单,当它作为免费软件出现以来,还是吸引了大批的使用者。

Cleve Moler 和 Jack Little 等人于 1984 年成立了一个名为 The MathWorks 的公司, Cleve Moler 任该公司的首席科学家。当时的 MATLAB 版本已经用 C 语言作了完全的改写,其后又增添了丰富多彩的图形图像处理、多媒体功能、符号运算和它与其他流行软件的接口功能,使得 MATLAB 的功能越来越强大。

最早的 PC 版又称为 PC-MATLAB,其工作站版本又称为 Pro MATLAB。1990 年推出的 MATLAB 3.5i 版是第一个可以运行于 Microsoft Windows 下的版本,它可以在两个窗口上分别显示命令行计算结果和图形结果。稍后推出的 SimuLAB 环境首次引入了基于框图的建模与仿真功能,其模型输入的方式令人耳目一新,该环境就是现在所知的 Simulink 的前身。The MathWorks 公司于 1992 年推出了具有划时代意义的 MATLAB 4.0 版本,并于 1993 年推出了其 PC 版,充分支持在 Microsoft Windows 进行界面编程。1994 年推出的 4.2 版本扩充了 4.0 版本的功能,尤其在图形界面设计方面更提供了新的方法。1996 年 12 月推出的 MATLAB 5.0 版支持了更多的数据结构,如单元数据、数据结构体、多维数组、对象与类等,使其成为一种更方便、完美的编程语言。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,随之推出的全新版本的最优化工具箱和 Simulink 3.0 版达到了很高的档次。2000 年 9 月, MATLAB 6.0 问世,在操作界面上有了很大改观,同时还给出了程序发布窗口、历史信息窗口和变量管理窗口等,为用户的使用提供了很大的方便;在计算内核上抛弃了其一直使用的 LINPACK 和 EISPACK,而采用了更具优势的 LAPACK 软件包和 FFTW 系统,速度变得更快,数值性能也更好;在用户图形界面设计上也更趋合理;与 C 语言接口及转换的兼容性也更强;与之配套的 Simulink 4.0 版的新功能也特别引人注目。2004 年 6 月推出的 MATLAB 7.0 版引入的多领域物理建模仿真策略为控制系统仿真技术提供了全新的仿真实念和平台。2012 年 9 月推出的 MATLAB 8.0 版提供了全新的操作界面,如图 1-1 所示。另外,该版本还提供了更强大的工具、全新的 Simulink 编辑器与更强大的仿真功能。

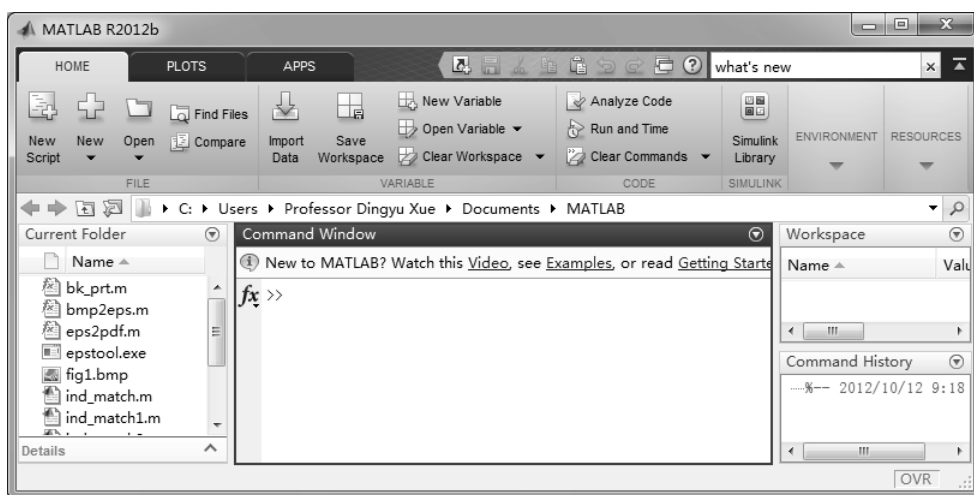


图 1-1 MATLAB 8.0 程序界面

MathWorks 公司每年在 3 月和 9 月分别推出 a 版和 b 版,当前最新的版本是 MATLAB R2012b (8.0 版),其功能越来越强大,然而其符号运算功能由于引擎的更换有所减弱。在使用符号运算功能时建议使用 MATLAB R2008a 或以前的版本,其他内容可以尽量采用新的版本。如果有条件的话建议同时安装两个版本,兼顾各方面的应用需求。本书以 MATLAB R2012b 为主介绍 MATLAB/Simulink 及其应用,其中绝大部分内容同样适用于 MATLAB 的早期版本。

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,现在的 MATLAB 已经不仅仅是一个“矩阵实验室”了,它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了,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 语言的功能也越来越强大,不断适应新的要求提出新的解决方法。另外,很多长期以来对 MATLAB 有一定竞

争能力的软件(如 Matrix-X)已经被 MathWorks 公司吞并,所以可以预见,在科学运算与系统仿真领域 MATLAB 语言将长期保持其独一无二的地位。

MATLAB 目前已经成为控制界国际上最流行的软件,它除了传统的交互式编程之外,还提供了丰富可靠的矩阵运算、图形绘制、数据处理、图像处理、方便的 Microsoft Windows 编程等便利工具。此外,控制界很多学者将自己擅长的 CAD 方法用 MATLAB 加以实现,出现了大量的 MATLAB 配套工具箱,如控制界最流行的控制系统工具箱(Control System Toolbox)、系统辨识工具箱(System Identification Toolbox)、鲁棒控制工具箱(Robust Control Toolbox)、多变量频域设计工具箱(Multivariable Frequency Design Toolbox)、 μ 分析与综合工具箱(μ -Analysis and Synthesis Toolbox)、神经网络工具箱(Neural Network Toolbox)、最优化工具箱(Optimization Toolbox)、信号处理工具箱(Signal Processing Toolbox)以及仿真环境 Simulink。参与编写这些工具箱的设计者包括国际控制界的名流,如 Alan Laub、Michael Sofanov、Leonard Ljung、Jan Maciejowski 等这些在相应领域的著名专家,这当然地提高了 MATLAB 的声誉与可信度,使得 MATLAB 风靡国际控制界,成为最重要也是最流行的语言。

下面列出一些有价值的网站供读者参考

<http://www.mathworks.cn> MathWorks 官网, MATLAB 及工具箱手册

<http://www.mathworks.cn/matlabcentral/fileexchange> 共享资源

<http://www.matlabsky.com> 国内比较好的 MATLAB 用户论坛

<http://www.matlab-edu.com> 本书作者维护的 MATLAB 教学网站

1.5 控制系统计算机辅助设计领域方法概述

在自动控制理论作为一门单独学科刚刚起步的时候,控制系统的设计是相当简单的,比如可以用 Ziegler-Nichols 经验公式^[3]利用纸和笔等简单的工具来设计较实用的 PID 控制器,这种现象持续了很长的时间。

随着计算机技术的发展,特别是像 MATLAB 这样方便可行的 CACSD 工具的出现,控制系统的计算机辅助设计在理论上也有了引人注目的进展,人们已经不满足于由纸和笔这样的简单工具设计出来的控制器了,而是期望越来越高。例如人们往往期望获得某种意义下的“最优”控制效果,而这样的控制效果确实是原来依赖纸和笔这样简单工具所实现不了的,而必须借助于计算机这样的高级工具,从而控制系统计算机辅助设计技术也就应运而生了。

早期的 CACSD 研究侧重于对控制系统的计算机辅助分析上,开始时人们利用计算机的强大功能把系统的频率响应曲线绘制出来,并根据频率响应的曲线及自己的控制系统设计经验用试凑的方法设计一个控制器,然后利用仿真的方法去观察设计的效果,比较成功的试凑设计方法有超前滞后校正方法等,当然这样的方法更适合于单变量系统的设计,前面提及以 Rosenbrock 教授和 MacFarlane 教

授为代表的英国学派多变量频域设计方法就是这种设计风格的范例。以色列裔美国学者 Issac Horowitz 教授在频域设计方法中独辟蹊径,创立了比较完善的设计方法—定量反馈理论(quantitative feedback theory, QFT)^[51],在反馈的效果上大作文章,在频域设计领域发展过程中,这些学者往往依赖于他们自己编写的 CACSD 工具来进行研究,并出现了很多值得提及的软件(如 CLADP),后来随着 MATLAB 的发展,也出现了各种各样的 MATLAB 工具箱,如 Jan Maciejowski 等学者开发的多变量系统频域设计工具箱^[52]和美国学者 Craig Borghesani 和 Yossi Chait 等编写的 QFT 设计工具箱^[53]等。

除了经典的多变量频域方法之外,还出现了一些基于最优化技术的控制方法,其中比较著名的是英国学者 John Edmunds 提出的多变量参数最优化控制方法和英国学者 Zakian 提出的不等式控制方法(method of inequalities)^[54]等,这些方法都是行之有效的实用设计方法。

与此同时,美国学者似乎更倾向于状态空间的表示与设计方法(往往又称为时域方法, time-domain),首先在线性二次型指标下引入了最优控制的概念,并在用户的干预下(如人工选择加权矩阵)得出某种最优控制的效果,这样的控制又往往需要引入状态反馈或状态观测器等新的控制概念。此后为了考虑随机扰动的情况引入了 LQG 最优控制的设计方法,后来随着 LQG 控制固有的弊病提出了回路传输恢复(loop transfer recovery, LTR)等新技术,但直到这类状态空间方法找到了合适的频域解释之后才开始有了应用。此外在状态空间的设计方法中比较成型的方法有极点配置方法、多变量系统解耦控制设计等,这些状态空间方法在计算方法和理论证明上取得了很多成果。

从控制系统的鲁棒性(robustness)角度也出现了各种各样的控制方法。首先由美国学者 Zames 提出的最小灵敏度控制策略引起了各国研究者的瞩目,并对之加以改进,出现了各种 \mathcal{H}_∞ 最优控制的方案。所谓 \mathcal{H}_∞ 实际上是物理可实现的稳定系统集合的一种数学描述(因满足 Hardy 空间而得名), \mathcal{H}_∞ 控制的一个关键问题是 Youla 参数化方法,该方法可以给出所有满足要求的控制器的通式。 \mathcal{H}_∞ 的解法也是多种多样的,首先人们考虑通过 Youla 参数化方法构造出全部镇定控制器,并将原始问题转化成模型匹配(如 Hankel 近似)的一般问题,然后再对该问题求解,后来多采用状态空间的解法,因为这样的解法更直观、容易,也更简洁。后来随着控制器的阶次越来越高,还出现了很多的控制器降阶方法来实现设计出的控制器。线性矩阵不等式(linear matrix inequalities, LMI)及 μ 分析与综合等控制系统设计方法也在控制界有较大的影响,而这些方法不通过计算机这样的现代化工具是不能完成的。

瑞典学者 Karl Åström 教授的研究更加切合于过程控制的实际应用,在他的研究成果中经常可以发现独创性的内容,例如他和合作者对传统的,也是工业中应用最广泛的 PID 控制器进行了改进,提出了自整定 PID 控制器^[55]的思想,使得原

来需要离线调节的 PID 控制器参数能够容易地在线自动调节,并在研究中取得了丰硕的成果,还推出了自整定 PID 控制器的硬件产品。在自整定 PID 控制器的领域也出现了很多比较显著的进展,这类研究的基本思想是使得复杂问题简单化,并易于实际应用。

分数阶控制是近年蓬勃发展起来的较新的研究方向^[56],领域也出现了很多新的研究成果,是控制理论的一个较新的研究领域。

国际上也出版了关于 MATLAB 及 CACSD 的专著和教材^[10~12],但它们大都是 MATLAB 的入门教材,并没有真正深入、系统地探讨 CACSD 技术及 MATLAB 实现,将 MATLAB 的强大功能与控制领域成果有机结合是本书力图解决的主要问题。本书第一版是国内第一部将 MATLAB 语言和控制系统设计技术有机结合的著作,曾在控制系统及相关学科的教学与研究中有较大的影响,本版进一步充实相关内容,大大提高了所涉及问题的深度与广度。

1.6 本书的基本结构和内容

对控制系统进行仿真与计算机辅助设计的工作可以认为是三个阶段的有机结合,即所谓的 MAD(modelling、analysis、design,即系统建模、分析与设计)过程,首先需要给系统建立起数学模型,然后根据数学模型进行仿真分析。在系统分析时如果发现与实际系统不符,则可能是系统的数学模型有问题,需要重新建模再进行分析。建立起准确的数学模型,并分析了系统的性质后,就可以根据要求给系统设计控制器。设计后可以对系统在控制器作用下的性质进行分析,如果不理想则应该重新设计控制器,再返回分析过程,直至得到满意的控制效果。当然,在系统分析与设计的过程中有时还需要对系统模型进行修正。

围绕控制系统仿真与计算机辅助设计的几个阶段,本书各章内容安排如下:

本章对国际上最流行的一些 CACSD 专用软件,如 ACSL、MATLAB、Mathematica 等做简要的介绍,然后对 CACSD 领域的新策略和新成果做一个概略的叙述,阐述了为什么在控制系统仿真与计算机辅助设计领域应该采用 MATLAB 作为主要计算机语言的原因。

第 2 章系统地介绍 MATLAB 编程的基础,包括对赋值语句、控制结构和绘图语句等程序设计问题,还介绍了 MATLAB 主流的程序设计方法,即函数编写方法与技巧。该章介绍了 MATLAB 语言二维、三维图形显示方法,并介绍了图形用户界面设计方面的基础知识与设计技巧。

第 3 章叙述了和控制系统仿真与设计领域密切相关的科学运算问题求解方法。介绍了 MATLAB 语言在线性代数、代数方程、微分方程、最优化及 Laplace 和 z 变换等几个与控制系统密切相关的科学运算问题上的应用。

第 4 章介绍在 MATLAB 环境中如何表示各种各样的线性系统数学模型。这里介绍的方法适用于连续与离散模型、单变量与多变量模型,为本书的理论基础。

该章还对方框图模型的连接与化简、各种数学模型之间的相互转换等给出较详细的叙述,还介绍了高阶模型的各种降阶方法和离散模型的辨识方法,探讨了辨识用激励信号问题。

第 5 章介绍线性控制系统的基本分析方法。首先介绍了控制系统的定性分析方法,如稳定性、可控性与可观测性等,还给出了线性系统的范数测度等概念与求解方法。该章讨论了线性系统的时域解析解方法与仿真方法,介绍了单变量与多变量系统的频域分析方法与根轨迹分析方法,侧重于如何用 MATLAB 语言解决相关问题的方法。学习这些方法将有助于控制系统的控制器设计。

第 6 章介绍基于 Simulink 的非线性系统建模方法与技巧。先介绍了各个常用的 Simulink 模块组,然后介绍了利用 Simulink 模块搭建仿真系统的方法,并介绍了仿真参数的设置方法、仿真中的子系统与模块封装技术以及 S-函数的编写方法。掌握了这些方法,理论上可以搭建起任意复杂系统的仿真模型。

第 7 章介绍各种经典的控制系统设计方法,包括系统的串联控制器的设计方法,基于二次型最优控制和极点配置的状态反馈控制方法,观测器的设计与基于观测器的控制器设计方法,最优控制器的设计方法与设计程序,多变量系统的频域设计方法和多变量控制系统的解耦方法等。

第 8 章侧重于工业上最常用的 PID 类控制器设计方法,包括各种 PID 控制器的结构、过程模型的一阶延迟近似、常规 PID 控制器整定方法、一般模型的 PID 控制器整定方法,并着重介绍作者编写的 OptimPID 用户界面,设计出性能最优的 PID 控制器。

第 9 章介绍各类系统的鲁棒控制器设计方法,包括基于 LQG/LTR 的鲁棒控制器设计方法、 \mathcal{H}_∞ 鲁棒控制问题及求解方法等,并介绍基于定量反馈理论的鲁棒控制器设计方法与工具。

第 10 章介绍自适应控制与智能控制系统的建模、仿真与设计问题,包括模型参考自适应系统、自校正控制与广义预测控制的仿真,模糊逻辑与模糊逻辑控制器设计、仿真研究,神经网络与基于各种神经网络结构的 PID 控制器设计与仿真,遗传算法、粒子群算法在最优化问题求解中的应用及最优控制器设计等。

第 11 章介绍分数阶系统的控制问题,包括分数阶微积分的定义,分数阶微分方程的框图解法等基础内容,也将以分数阶传递函数为例,介绍 MATLAB 的类与对象编程方法,介绍基于面向对象技术的分数阶系统建模、分析与设计技术。

第 12 章搭建起控制系统设计软件与硬件实现之间的桥梁,系统地介绍了基于 dSPACE、Quanser 等软硬件系统的控制系统半实物仿真及实时控制方法,为控制理论及方法的工程应用打下一定的基础。

本书附录介绍几个用于控制系统计算机辅助设计软件测试用的基准问题与控制应用问题,用户可以从这些典型的受控对象模型出发,测试各种自己设计出来的控制器,得到公平的比较结果。

1.7 习 题

- (1) MathWorks 公司网站 (<http://www.mathworks.cn>) 上提供了 MATLAB 及所有工具箱手册的电子版, 如果需要, 可以将感兴趣的工具箱手册下载阅读。由于工具箱规模过于庞大, 不可能在一本书中全盘介绍, 所以本书的作用只是作为读者学习和使用 MATLAB 语言解决控制中问题的入门指导性材料。
- (2) MATLAB 语言是控制系统研究的首选语言, 本书以该语言为主线介绍课程的内容。请在机器上安装 MATLAB 程序, 在提示符下输入 `demo` 命令, 运行演示程序, 领略 MATLAB 语言的基本功能。
- (3) 学会利用 MATLAB 语言提供的联机帮助功能, 更好地学习 MATLAB 语言, 熟练掌握查找需要了解内容的方法和技巧。联机帮助可以由 `help` 命令或 `Help` 菜单来实现, 还可以用 `doc` 命令获得帮助。
- (4) MATLAB 语言中求取两个矩阵的乘积用 $C = A * B$ 即可, 再试用 `C` 或其他计算机语言编写一个通用的子程序, 计算两个矩阵 A 和 B 的乘积, 体验一下通用程序编写和 MATLAB 现成功能调用的区别。

- (5) 矩阵运算是 MATLAB 最传统的特色, 用 $B = \text{inv}(A)$ 命令即可以求出 A 的逆矩阵, 感受 MATLAB 在求解逆矩阵时的运算效率。想求一个 n 阶随机矩阵的逆, 分别取 $n=550$ 和 $n=1550$, 测试矩阵求逆所需的时间及结果的正确性。具体语句:

```
>> tic, A = rand(550); B = inv(A); toc
>> norm(A*B-eye(size(A)), norm(B*A-eye(size(A))))
```

(注: 由矩阵的性质可知, $AB = BA = I$, 故用 $\|AB - I\|$ 即可以得出误差)

- (6) 在求解数学问题时, 不同的算法在求解精度与速度上是不同的。考虑求取矩阵行列式的代数余子式方法, 可以将 n 阶矩阵的行列式问题转化成 n 个 $n-1$ 阶矩阵的行列式问题, $n-1$ 阶矩阵又可以转化成 $n-2$ 阶。因而可以得出结论: 任意阶矩阵行列式均可以由代数余子式方法求出解析解。然而, 这样的结论忽略了计算量的问题, 用这样的方法, n 阶矩阵行列式求解的计算量为 $(n-1)(n+1)! + n$, $n=20$ 的计算量相当于每秒百亿次的巨型计算机求解 3000 多年, 所以用该方法不可能真正用于大型矩阵的行列式求解。

由矩阵运算可知, 可以将矩阵进行 LU 分解, 计算出矩阵的行列式, MATLAB 的解析解运算也实现了这样的算法, 可以在短时间内求解出矩阵的行列式。试用 MATLAB 语言求解 20 阶矩阵的行列式解析解, 需要使用多少时间。具体参考语句

```
tic, A = sym(hilb(20)); det(A); toc
```

- (7) 试用不同的计算机数学语言如 Mathematica、Maple、MATLAB 及 MATLAB 的符号运算工具箱分别求解代数方程 $(x+1)^{20} = 0$, 比较得出结果, 并说明解决这样的问题应该注意什么 (注: 求解前应该先将方程左侧展开成一般的多项式)。

- (8) MATLAB 语言的 Simulink 仿真程序允许用户用直观的方法搭建控制系统的框图, 试利用 Simulink 提供的模块搭建起一个如图 1-2 所示的模型, 请研究 $\delta = 0.3$ 时不同输入信号激励下系统的响应曲线, 另外请研究不同的 δ 值对系统的阶跃响应有何影响, 通过这个例子可以领略利用仿真工具的优势及方便程度。

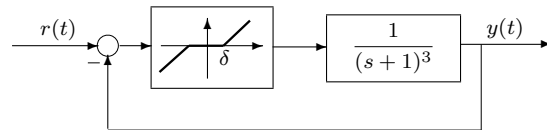


图 1-2 某非线性反馈系统框图

参考文献

- 1 戴先中. 自动化科学与技术学科的内容、地位与体系. 北京: 高等教育出版社, 2003
- 2 Doyle J C. A new physics? 40th IEEE Conference on Decision and Control 全会开篇大会报告. Orlando: IEEE Publisher, 2000
- 3 Ziegler J G, Nichols N B. Optimum settings for automatic controllers. Transaction of ASME, 1944, 64:759~768
- 4 Herget C J, Laub A J (eds.). Special issue on computer-aided control system design. IEEE Control Systems Magazine, 1982, 2(4):2~37
- 5 Herget C J, Laub A J (eds.). Special issue on computer-aided control system design. Proceedings of IEEE, 1984, 72:1714~1805
- 6 Jamshidi M, Herget C J. (eds.). Computer-aided control systems engineering. Amsterdam: Elsevier Science Publishers B V, 1985
- 7 Jamshidi M, Herget C J. (eds.). Recent advances in computer-aided control systems engineering. Amsterdam: Elsevier Science Publishers B V, 1992
- 8 孙增圻, 袁曾任. 控制系统的计算机辅助设计. 北京: 清华大学出版社, 1988
- 9 Melsa J L, Jones S K. Computer programs for computational assistance in the study of linear control theory. New York: McGraw-Hill, 1973
- 10 Shahian B, Hassul M. Computer-aided control system design using MATLAB. Englewood Cliffs: Prentice-Hall, 1993
- 11 Leonard N E, Levine W S. Using MATLAB to analyze and design control systems. Redwood City: Benjamin Cummings, 1993
- 12 Ogata K. Solving control engineering problems with MATLAB. Englewood Cliffs: Prentice Hall, 1994
- 13 Mościński J, Ogonowski Z. Advanced control with MATLAB and Simulink. London: Ellis Horwood, 1995
- 14 薛定宇. 控制系统计算机辅助设计——MATLAB 语言与应用. 北京: 清华大学出版社, 1996
- 15 张晓华. 控制系统计算机仿真与 CAD. 北京: 机械工业出版社, 1999

- 16 薛定宇. 反馈控制系统的设计与分析——MATLAB 语言应用. 北京: 清华大学出版社, 2000
- 17 薛定宇. 控制系统仿真与计算机辅助设计. 北京: 机械工业出版社, 2005
- 18 Kuo B C. Automatic control systems. Wiley, 8th edition, 2003
- 19 D’Azzo J J, Houpis C H. Linear control system analysis and design: conventional and modern. New York: McGraw-Hill, 4th edition, 1995
- 20 Ogata K. Modern control engineering. Englewood Cliffs: Prentice Hall, 4th edition, 2001
- 21 Åström K J. Computer aided tools for control system design, In: Jamshidi M, Herget C J. Computer-aided control systems engineering. Amsterdam: Elsevier Science Publishers B V, 1985, 3~40
- 22 Furuta K. Computer-aided design program for linear control systems. Proceedings of IFAC Symposium on CACSD, 1979, 267~272. Zurich, Switzerland
- 23 Edmunds J M. Cambridge linear analysis and design programs. Proceedings IFAC Symposium on CACSD, 1979, 253~258. Zurich, Switzerland
- 24 Maciejowski J M, MacFarlane A G J. CLADP: the Cambridge linear analysis and design programs, In: Jamshidi M, Herget C J. Computer-aided control systems engineering. Amsterdam: Elsevier Science Publishers B V, 1985, 125~138
- 25 Armstrong E S. ORACLS — a design system for linear multivariable control. New York: Marcel Dekker Inc., 1980
- 26 Little J N, Emami-Naeini A, Bangert S N. CTRL-C and matrix environments for the computer-aided design of control systems, In: Jamshidi M, Herget C J. Computer-aided control systems engineering. Amsterdam: Elsevier Science Publishers B V, 1985, 191~205
- 27 Octave 语言主页. <http://www.octave.org>
- 28 SciLAB 语言主页. <http://scilabsoft.inria.fr>
- 29 Jobling C P, Grant P W, Barker H A, Townsend P. Object-oriented programming in control system design: a survey. Automatica, 1994, 30:1221~1261
- 30 Smith B T, Boyle J M, Dongarra J J. Matrix eigensystem routines — EISPACK guide, *Lecture Notes in Computer Sciences*, volume 6. New York: Springer-Verlag, 2nd edition, 1976
- 31 Garbow B S, Boyle J M, Dongarra J J, Moler C B. Matrix eigensystem routines — EISPACK guide extension, *Lecture Notes in Computer Sciences*, volume 51. New York: Springer-Verlag, 1977
- 32 Dongarra J J, Bunch J R, Moler C B, Stewart G W. LINPACK user’s guide. Philadelphia: Society of Industrial and Applied Mathematics (SIAM), 1979
- 33 Numerical Algorithm Group. NAG FORTRAN library manual, 1982
- 34 Press W H, Flannery B P, Teukolsky S A, Vetterling W T. Numerical recipes, the art of scientific computing. Cambridge: Cambridge University Press, 1986
- 35 Atherton D P. Control systems computed. Physics in Technology, 1985, 16:139~140
- 36 CAD Center. GINO-F Users’ manual, 1976

- 37 Spang III H A. The federated computer-aided design system, In: Jamshidi M, Herget C J. Computer-aided control systems engineering. Amsterdam: Elsevier Science Publishers B V, 1985, 209~228
- 38 Munro N. ECSTASY — a control system CAD environment. Proceedings IEE Conference on Control 88, 1988, 76~80. Oxford
- 39 Wolfram S. Mathematica: a system for doing mathematics by computer. Redwood City, California: Addison-Wesley Publishing Company, 1988
- 40 Lamport L. \LaTeX : a document preparation system — user's guide and reference manual. Reading MA: Addison-Wesley Publishing Company, 2nd edition, 1994
- 41 王治宝, 韩京清. CAD/CSC 软件系统——控制系统计算机辅助设计. 北京: 科学出版社, 1997
- 42 吴重光, 沈承林. 控制系统计算机辅助设计. 北京: 机械工业出版社, 1988
- 43 Mitchell E E L, Gauthier J S. Advanced continuous simulation language (ACSL) — user's manual. Mitchell & Gauthier Associates, 1987
- 44 Hay J L, Pearce J G, Turnbull L, Crosble R E. ESL software user manual. Salford: ISIM Simulation, 1988
- 45 Wooff C, Hodgkinson D. muMATH: A microcomputer algebra system. London: Academic Press, 1987
- 46 Rayna G. REDUCE software for algebraic computation. New York: Springer-Verlag, 1987
- 47 Forsythe G E, Malcolm M A, Moler C B. Computer methods for mathematical computations. Englewood Cliffs: Prentice-Hall, 1977
- 48 Forsythe G E, Moler C B. Computer solution of linear algebraic systems. Englewood Cliffs: Prentice-Hall, 1967
- 49 Moler C B. Numerical computing with MATLAB. MathWorks Inc, 2004
- 50 Moler C B. MATLAB — An interactive matrix laboratory. Technical Report 369, Department of Mathematics and Statistics, University of New Mexico, 1980
- 51 Horowitz I. Quantitative feedback theory (QFT). Proceedings IEE, Part D, 1982, 129:215~226
- 52 Boyel J M, Ford M P, Maciejowski J M. A multivariable toolbox for use with MATLAB. IEEE Control Systems Magazine, 1989, 9:59~65
- 53 Borghesani C, Chait Y, Yaniv O. The QFT frequency domain control design toolbox for use with MATLAB. Terasoft Inc, 2003
- 54 Zakian V, Al-Naib U. Design of dynamical and control systems by the method of inequalities. Proceedings of IEE, Part D, 1973, 120:1421~1427
- 55 Åström K J, Hägglund T. Automatic tuning of simple regulators with specification on phase and amplitude margins. Automatica, 1984, 20:645~651
- 56 Monje C A, Chen Y Q, Vinagre B M, Xue D, Feliu V. Fractional-order systems and controls — fundamentals and applications. London: Springer, 2010

MATLAB 语言是当前国际上自动控制领域的首选计算机语言,也是很多理工科专业最适合的计算机数学语言。本书以 MATLAB 语言为主要计算机语言,系统、全面地介绍其在控制系统建模、分析与计算机辅助设计中的应用。掌握该语言不但有助于更深入理解和掌握控制理论中的概念与算法,提高分析与设计控制系统的的能力,而且还可以充分利用该语言,在其他专业课程的学习中得到积极的帮助。

和其他程序设计语言相比,MATLAB 语言有如下的优势:

① **简洁高效性**。MATLAB 程序设计语言集成度高,语句简洁,往往用 C/C++ 等程序设计语言编写的数百条语句,用 MATLAB 语言一条语句就能解决问题,其程序可靠性高、易于维护,可以大大提高解决问题的效率和水平。

② **科学运算功能**。MATLAB 语言以矩阵为基本单元,可以直接用于矩阵运算。另外,最优化问题、数值微积分问题、微分方程数值解问题、数据处理问题等都能直接用 MATLAB 语言求解。

③ **绘图功能**。MATLAB 语言可以用最直观的语句将实验数据或计算结果用图形的方式显示出来,并可以将以往难以显示出来的隐函数直接用曲线绘制出来。MATLAB 语言还允许用户用可视的方式编写图形用户界面,其难易程度和 Visual Basic 相仿,这使得用户可以容易地利用该语言编写通用程序。

④ **庞大的工具箱与模块集**。MATLAB 是被控制界的学者“捧红”的,是控制界通用的计算机语言,在应用数学及控制领域几乎所有的研究方向均有自己的工具箱,而且由领域内知名专家编写,可信度比较高。随着 MATLAB 的日益普及,在其他工程领域也出现了工具箱,这也大大促进了 MATLAB 语言在各个领域的应用。

⑤ **强大的动态系统仿真功能**。Simulink 提供的面向框图的仿真及多领域物理仿真功能,使得用户能容易地建立复杂系统模型,准确地对其进行仿真分析。Simulink 的多领域物理仿真模块集允许用户在一个

统一的框架下对含有控制环节、机械环节和电子、电机环节的机电一体化系统进行建模与仿真,这是目前其他计算机语言无法做到的。

本章中的内容安排如下:2.1节介绍 MATLAB 语言编程的最基本内容,包括数据结构、基本语句结构和重要的冒号表达式与子矩阵提取方法。2.2节介绍 MATLAB 语言中矩阵的基本数学运算,包括代数运算、逻辑运算、比较运算及简单的数论运算函数。2.3节介绍 MATLAB 语言的基本编程结构,如循环语句结构、条件转移语句结构、开关结构和试探结构,介绍各种结构在程序设计中的应用。2.4节介绍 MATLAB 语言编程中主流结构——M-函数的结构与程序编写技巧。2.5节介绍基于 MATLAB 语言的二维图形绘制的方法,如各种二维曲线绘制、隐函数的曲线绘制等,并将介绍图形修饰方法等。2.6节介绍三维图形的绘制方法、三维图形旋转与视角设置等。2.7节介绍图形用户界面的设计方法与技巧。

2.1 MATLAB 程序设计语言基础

2.1.1 MATLAB 语言的变量与常量

MATLAB 语言变量名应该由一个字母引导,后面可以跟字母、数字、下划线等。例如,MYvar12、MY_Var12 和 MyVar12_ 均为有效的变量名,而 12MyVar 和 _MyVar12 为无效的变量名。在 MATLAB 中变量名是区分大小写的,也就是说,Abc 和 ABc 两个变量名表达的是不同的变量,在使用 MATLAB 语言编程时一定要注意。

在 MATLAB 语言中还为特定常数保留了一些名称,虽然这些常量都可以重新赋值,但建议在编程时应尽量避免对这些量重新赋值。

① eps,机器的浮点运算误差限。PC 上 eps 的默认值为 2.2204×10^{-16} ,若某个量的绝对值小于 eps,则可以认为这个量为 0。

② i 和 j。若 i 或 j 量不被改写,则它们表示纯虚数量 j。但在 MATLAB 程序编写过程中经常事先改写这两个变量的值,如在循环过程中常用这两个变量来表示循环变量,所以应该确认使用这两个变量时没有被改写。如果想恢复该变量,则可以用语句 `i = sqrt(-1)` 设置,即对 -1 求平方根。

③ Inf,无穷大量 $+\infty$ 的 MATLAB 表示,也可以写成 inf。同样地, $-\infty$ 可以表示为 -Inf。在 MATLAB 程序执行时,即使遇到了以 0 为除数的运算,也不会终止程序的运行,而只给出一个“除 0”警告,并将结果赋成 Inf,这样的定义方式符合 IEEE 的标准。从数值运算编程角度看,这样的实现形式明显优于 C 语言这类的非专业的计算机语言。

④ NaN,不定式(not a number, NaN),通常由 0/0 运算、Inf/Inf 及其他可能的运算得出。NaN 是一个很奇特的量,如 NaN 与 Inf 的乘积仍为 NaN。

⑤ pi,圆周率 π 的双精度浮点表示。

2.1.2 数据结构

1. 数值型数据

强大方便的数值运算功能是 MATLAB 语言的最显著特色。为保证较高的计算精度, MATLAB 语言中最常用的数值量为双精度浮点数, 占 8 个字节(64 位), 遵从 IEEE 记数法, 有 11 个指数位、53 位尾数及一个符号位, 值域的近似范围为 -1.7×10^{308} 至 1.7×10^{308} , 其 MATLAB 表示为 `double()`。考虑到一些特殊的应用, 比如图像处理, MATLAB 语言还引入了无符号的 8 位整型数据类型, 其 MATLAB 表示为 `uint8()`, 其值域为 $0 \sim 255$, 这样可以大大地节省 MATLAB 的存储空间, 提高处理速度。此外, 在 MATLAB 中还可以使用其他的数据类型, 如 `int8()`、`int16()`、`int32()`、`uint16()`、`uint32()` 等, 每一个类型后面的数字表示其位数, 其含义不难理解。

2. 符号型

MATLAB 还定义了“符号”型变量, 以区别于常规的数值型变量, 可以用于公式推导和数学问题的解析解法。进行解析运算前需要首先将采用的变量申明为符号变量, 这需要用 `syms` 命令来实现。该语句具体的用法为 `syms vars props`, 其中, `vars` 给出需要申明的变量列表, 可以同时申明多个变量, 中间用空格分隔, 而不是用逗号等分隔。如果需要, 还可以进一步申明变量的类型 `props`, 可以使用的类型为 `real`、`positive` 等。如果需要将 a 、 b 均定义为符号变量, 则可以用 `syms a b` 语句声明, 该命令还支持对符号变量具体形式的设定, 如 `syms a real`。

符号型数值可以通过变精度算法函数 `vpa()` 以任意指定的精度显示出来。该函数的调用格式为 `vpa(A)`, 或 `vpa(A, n)`, 其中, A 为需要显示的数值或矩阵, n 为指定的有效数字位数, 前者以默认的十进制位数(32 位)显示结果。

例 2-1 自然对数底 e 的前 300 位有效数字可以由下面的语句直接显示出来。

```
>> vpa(exp(sym(1)), 300) % 需要首先将 1 转换成符号量
```

显示的结果为 2.718281828459045235360287471352662497757247093699959574966967
62772407663035354759457138217852516642742746639193200305992181741359662904
35729003342952605956307381323286279434907632338298807531952510190115738341
87930702154089149934884167509244761460668082264800168477411853742345442437
10753907774499207。若不指定位数则显示为 2.7182818284590452353602874713527。

3. 其他数据结构

除了用于数学运算的数值数据结构外, MATLAB 还支持下面的数据结构:

① **字符串型数据**。MATLAB 支持字符串变量, 可以用它来存储相关的信息。和 C 语言等程序设计语言不同, MATLAB 字符串是用单引号括起来的。

② **多维数组**。三维数组是一般矩阵的直接拓展。在控制系统的分析中也可以直接用于多变量系统的表示上。在实际编程中还可以使用维数更高的数组。

③ **单元数组**。单元数组是矩阵的直接扩展, 其存储格式类似于普通的矩阵, 而

矩阵的每个元素不是数值,可以认为能存储任意类型的信息,这样每个元素称为“单元”(cell),例如, $A\{i,j\}$ 可以表示单元数组 A 的第 i 行,第 j 列的内容。

④ 类与对象。MATLAB 允许用户自己编写包含各种复杂信息的变量,称为类,而对象是类的一个实例。类变量可以包含各种下级的信息,还可以重新对类定义其计算,这在控制系统描述中特别有用。例如,在 MATLAB 的控制系统工具箱中定义了传递函数类,可以用一个变量来表示整个传递函数,还重新定义了该类的运算,如加法运算可以直接求取多个模块的并联连接,乘法运算可以求取若干模块的串联。这些内容将在后面的章节中详细介绍。

2.1.3 MATLAB 的基本语句结构

MATLAB 的语句有两种结构。

1. 直接赋值语句

直接赋值语句的基本结构为赋值变量 = 赋值表达式,这一过程把等号右边的表达式直接赋给左边的赋值变量,并返回到 MATLAB 的工作空间。如果赋值表达式后面没有分号,则将在 MATLAB 命令窗口中显示表达式的运算结果。若不想显示运算结果,则应该在赋值语句的末尾加一个分号。如果省略了赋值变量和等号,则表达式运算的结果将赋给保留变量 `ans`。所以说,保留变量 `ans` 将永远存放最近一次无赋值变量语句的运算结果。

例 2-2 矩阵的输入在 MATLAB 下是非常简单、直观的,考虑矩阵 $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$,由

下面的 MATLAB 语句将该矩阵直接输入到工作空间中。

```
>> A=[1,2,3; 4 5,6; 7,8 0] % 逗号、空格分隔元素,分号换行
```

其中,>> 为 MATLAB 的提示符,由机器自动给出,用户在提示符下可以输入各种各样的 MATLAB 命令。矩阵的内容由方括号括起来的部分表示,在方括号中的分号表示矩阵的换行,逗号或空格表示同一行矩阵元素间的分隔。百分号(%)引导注释语句。给出了上面的命令,就可以在 MATLAB 的工作空间中建立一个 A 变量并在 MATLAB 命令窗口中显示出来。在本书后续正文中为便于理解,将采用数学形式显示得出的结果。如果不想显示中间结果,则应该在语句末尾加一个分号,如

```
>> A=[1,2,3; 4 5,6; 7,8 0]; % 不显示结果,但进行赋值
A=[[A; [1 2 3]], [1;2;3;4]]; % 矩阵维数动态变化
```

例 2-3 复数矩阵在 MATLAB 下也可以很直观地输入,考虑下面的复数矩阵

$$B = \begin{bmatrix} 1+9j & 2+8j & 3+7j \\ 4+6j & 5+5j & 6+4j \\ 7+3j & 8+2j & 0+j \end{bmatrix}$$

利用 MATLAB 语言定义的两个记号 i 和 j ,可以直接输入复数矩阵,该复数矩阵可以由下面的语句直接输入