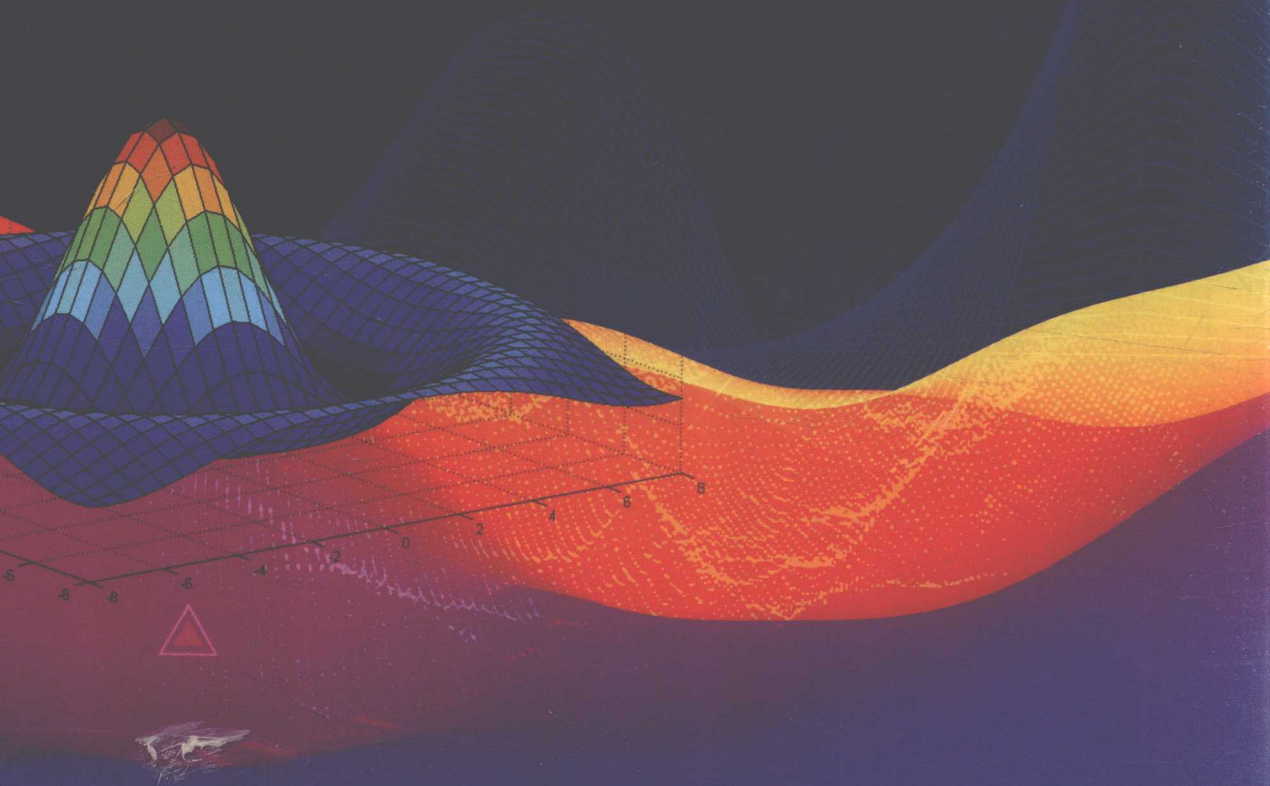


目标定位跟踪 原理及应用 ——**MATLAB** 仿真

© 黄小平 王 岩 缪鹏程 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

目标定位跟踪原理及应用

——MATLAB 仿真

黄小平 王 岩 缪鹏程 编著



電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书主要介绍目标定位原理、定位算法仿真及对动态目标跟踪过程中涉及的状态估计方法。为方便读者快速掌握定位跟踪的算法原理,本书采用“原理介绍+实例应用+MATLAB 程序仿真+注释”的方式,方便读者阅读和理解。本书共分8章,第1章介绍 MATLAB 仿真基础,尤其是数组的相关操作对算法仿真至关重要;第2章简介目标定位跟踪系统,包括观测系统、定位的概念、目标跟踪等;第3章介绍目标定位算法,重点介绍质心定位算法和基于测距的定位算法;第4章介绍线性 Kalman 滤波;第5章介绍非线性 Kalman 滤波;第6章介绍粒子滤波;第7章介绍多目标跟踪算法;第8章介绍 Simulink 仿真。

本书可以作为电子信息类专业的本科生和硕士、博士研究生“数字信号处理”课程的教材,也可以作为从事雷达、无线传感器网络、数字信号处理的教师和科研人员的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

目标定位跟踪原理及应用: MATLAB 仿真/黄小平,王岩,缪鹏程编著. —北京:电子工业出版社,2018.12

ISBN 978-7-121-35345-1

I. ①目… II. ①黄… ②王… ③缪… III. ①Matlab 软件-应用-目标跟踪 IV. ①TN953

中国版本图书馆 CIP 数据核字(2018)第 245088 号

策划编辑:刘海艳

责任编辑:韩玉宏 特约编辑:曹尊颖

印 刷:三河市君旺印务有限公司

装 订:三河市君旺印务有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:15.75 字数:403.2 千字

版 次:2018 年 12 月第 1 版

印 次:2018 年 12 月第 1 次印刷

印 数:1 500 册 定价:59.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888,88258888。

质量投诉请发邮件至 zltts@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: lhy@phei.com.cn。

前 言

随着科技的发展,在雷达、声呐、无线领域,对目标位置锁定的研究在各应用场合有着重要的价值。所有传感器测量的数据都是受到噪声污染的,噪声不能消除,只能最大限度地降低。在目标跟踪中,传感器一般都采集观测站与目标之间的距离、角度等信息,这些信息往往会受到高斯或非高斯噪声的干扰,导致观测站不能准确地估计目标的状态,常用的补偿措施就是滤波。

在现代时间序列里,常用的滤波算法有最小二乘定位算法、Kalman 滤波、粒子滤波等。这些经典的算法已经广泛应用在雷达、声呐、无线传感器网络等领域中。本书主要结合实际应用,如单观测站与多观测站情况下对目标进行状态估计研究,这里的状态主要指的是目标的位置、速度和加速度信息。对目标的状态估计算法,笔者已在电子工业出版社出版的《卡尔曼滤波原理及应用——MATLAB 仿真》和《粒子滤波原理及应用——MATLAB 仿真》两本书中做了详细的描述。本书在写作风格上沿袭了前两本,期望能得到广大读者的认同。

参加本书编写的还有王岩、缪鹏程、胡敦、许蓓蓓、李学森、章华、罗伟、丁成祥、黄平森、高福仙等。本书得以撰写,在很大程度上要感谢我的恩师王岩老师,她给了我一个很好的研究课题,在研究生期间对我的谆谆教诲和无微不至的关怀,让我学到了很多知识,在我遇到困难时给我指点迷津,提出很多建设性的意见供我参考。本书的编辑也得到了北航同课题组的学弟、学妹们的帮助,感谢他们查阅并提供的资料,在此特别感谢钱琛、王夏静、段九鼎、杨刚、徐建伟、王驭风、严芬菲、刘涛、聂金平、陈冰洁、田龙飞、李超等。最后感谢我的妻子许蓓蓓的理解和支持,感谢可爱的女儿给我写作的精神动力!

希望本书对相关领域的研究者有所帮助。由于作者水平有限,书中难免有疏忽及错误之处,恳请读者提出宝贵意见。我的邮箱是 xiaoping_444@126.com。



2018 年 4 月

目 录

第 1 章	MATLAB 仿真基础	1
1.1	MATLAB 简介	1
1.1.1	MATLAB 发展历史	1
1.1.2	MATLAB 7.1 系统	2
1.1.3	M-File 编辑器的使用	4
1.2	数据类型和数组	6
1.2.1	数据类型概述	6
1.2.2	数组的创建	7
1.2.3	数组的属性	8
1.2.4	数组的操作	9
1.2.5	结构体和元胞数组	12
1.3	程序设计	14
1.3.1	条件语句	14
1.3.2	循环语句	15
1.3.3	函数	17
1.3.4	画图	19
1.4	小结	21
第 2 章	目标定位跟踪系统概述	22
2.1	观测系统	22
2.1.1	基本概念	22
2.1.2	坐标系	22
2.1.3	典型的观测站	24
2.2	定位	28
2.2.1	模糊定位	28
2.2.2	精确定位	29
2.2.3	典型的定位系统	30
2.3	目标跟踪	32
2.3.1	多观测站系统架构	32
2.3.2	跟踪过程描述	33
2.3.3	跟踪轨迹	35
2.4	跟踪的数学模型	36
2.4.1	匀速直线运动	36

2.4.2	匀加速运动	37
2.5	小结	38
第3章	目标定位算法	39
3.1	非测距的定位算法	39
3.1.1	质心定位算法	39
3.1.2	加权质心定位算法	42
3.1.3	网格定位算法	45
3.2	基于测距的定位算法	47
3.2.1	最小二乘原理	47
3.2.2	最小二乘定位算法	50
3.2.3	基于 RSSI 测距的定位算法	54
3.2.4	基于 TOA/TDOA 的目标定位算法	58
3.3	基于角度测量的定位算法	62
3.3.1	双站角度定位	62
3.3.2	三角测量法定位	65
3.4	移动目标定位算法	65
3.4.1	移动目标计算机仿真建模	65
3.4.2	基于距离观测的运动目标定位	67
3.4.3	纯方位角的运动目标定位	70
3.5	小结	73
第4章	线性 Kalman 滤波	74
4.1	Kalman 滤波原理	74
4.1.1	射影定理	74
4.1.2	Kalman 滤波器	76
4.1.3	Kalman 滤波的参数处理	79
4.2	Kalman 滤波用于一维观测信号去噪	80
4.2.1	信号测量滤波原理	80
4.2.2	一维 Kalman 滤波的应用仿真	82
4.3	Kalman 滤波在自由落体运动目标跟踪中的应用	84
4.3.1	状态方程的建立	84
4.3.2	自由落体跟踪仿真	86
4.4	Kalman 滤波在船舶 GPS 导航定位系统中的应用	89
4.4.1	GPS 导航系统原理介绍	89
4.4.2	导航定位系统仿真	91
4.5	Kalman 滤波在视频图像目标跟踪中的应用	93
4.5.1	视频捕获和录制	94
4.5.2	视频导入和显示	96
4.5.3	对视频帧的操作	96
4.5.4	Kalman 滤波对自由下落的皮球跟踪应用	100

4.6 小结	106
第5章 非线性 Kalman 滤波	107
5.1 扩展 Kalman 滤波	107
5.1.1 EKF 的基本思想	107
5.1.2 非线性系统的线性化	107
5.1.3 EKF 的滤波原理	108
5.2 EKF 的应用及仿真	110
5.2.1 EKF 应用于一维非线性系统	110
5.2.2 基于距离观测的目标跟踪 EKF 状态估计	114
5.2.3 纯方位角目标跟踪 EKF 状态估计	117
5.2.4 EKF 在纯方位寻的导弹制导中的应用	120
5.3 无迹 Kalman 滤波	125
5.3.1 UKF 的基本思想	125
5.3.2 无迹变换	126
5.3.3 UKF 的滤波过程	127
5.4 UKF 的应用及仿真	128
5.4.1 UKF 在单站观测站目标跟踪中的应用	128
5.4.2 UKF 在六维匀加速直线运动目标跟踪系统中的应用	132
5.5 小结	137
第6章 粒子滤波	138
6.1 粒子滤波简介	138
6.1.1 粒子滤波的发展历史	138
6.1.2 粒子滤波的特点	139
6.1.3 粒子滤波的应用领域	139
6.2 蒙特卡洛原理	140
6.2.1 蒙特卡洛的基本思想	140
6.2.2 蒙特卡洛的理论基础	141
6.2.3 蒙特卡洛的应用实例	143
6.3 粒子滤波原理	147
6.3.1 蒙特卡洛采样原理	147
6.3.2 贝叶斯重要性采样	148
6.3.3 序列重要性采样滤波器	149
6.3.4 Bootstrap/SIR 滤波器	150
6.3.5 权值计算方法	151
6.4 粒子滤波经典采样算法	152
6.4.1 随机重采样	152
6.4.2 多项式重采样	154
6.4.3 系统重采样	156
6.4.4 残差重采样	158

6.5 粒子滤波的应用	160
6.5.1 粒子滤波应用于一维非线性系统	160
6.5.2 粒子滤波应用于高斯模型目标跟踪系统	165
6.5.3 粒子滤波应用于非高斯模型目标跟踪系统	171
6.6 小结	178
第7章 多目标跟踪算法	180
7.1 多目标跟踪系统建模	180
7.1.1 单站多目标跟踪系统建模	180
7.1.2 单站多目标跟踪系统仿真	180
7.1.3 多站多目标跟踪系统建模	183
7.1.4 多站多目标跟踪系统仿真	183
7.2 多目标跟踪分类算法	187
7.2.1 多目标数据融合概述	187
7.2.2 近邻法分类算法及程序	188
7.2.3 近邻法用于目标跟踪中的航迹关联及算法程序	191
7.2.4 k -近邻法分类算法	194
7.3 多目标跟踪算法	195
7.3.1 基于 Kalman 滤波的多目标跟踪算法	195
7.3.2 基于粒子滤波的多目标状态估计	204
7.4 小结	213
第8章 Simulink 仿真	214
8.1 Simulink 概述	214
8.1.1 Simulink 启动	214
8.1.2 Simulink 仿真设置	215
8.1.3 Simulink 模块库简介	219
8.2 S 函数	223
8.2.1 S 函数原理	223
8.2.2 S 函数的控制流程	226
8.3 目标跟踪的 Simulink 仿真	227
8.3.1 状态方程和观测方程的 Simulink 建模	227
8.3.2 基于 S 函数的 Kalman 滤波器设计及其在跟踪中的应用	230
8.3.3 基于 S 函数的粒子滤波器设计及其在跟踪中的应用	236
8.4 小结	243

第 1 章 MATLAB 仿真基础

MATLAB 是美国 MathWorks 公司出品的商业数学软件，是用于算法开发、数据可视化、数据分析及数值计算的高级技术计算语言和交互式环境，主要包括 MATLAB 和 Simulink 两大部分。经过多年的发展和版本的升级，如今的 MATLAB 功能已经非常强大，它是当今最流行的计算机仿真软件之一。

1.1 MATLAB 简介

1.1.1 MATLAB 发展历史

MATLAB 的产生是与数学计算紧密联系在一起的。1980 年，美国新墨西哥州立大学计算机系主任 Cleve Moler 在给学生讲授“线性代数”课程时，发现学生在高级语言编程上花费了很多时间，于是着手编写供学生使用的 Fortran 子程序库接口程序，取名为 MATLAB (Matrix Laboratory 前三个字母的组合，意为“矩阵实验室”)。这个程序获得了很大的成功，受到学生的广泛欢迎。

20 世纪 80 年代初，Moler 等一批数学家与软件专家组建了 MathWorks 软件开发公司，继续从事 MATLAB 的研究和开发工作。1984 年推出第一个 MATLAB 商业版本，其核心是用 C 语言编写的。之后，MATLAB 又增加了丰富多彩的图形图像处理、多媒体、符号运算及其他流行软件的接口功能，至此 MATLAB 的功能逐渐强大。

具有划时代意义的事件是 1992 年 MathWorks 公司正式推出了 MATLAB 1.0；到了 1999 年 MATLAB 5.3 进一步改进了原有功能，同时 Simulink 3.0 版本也达到较高水准；在 2000 年 10 月，MATLAB 6.0 出现，其无论是在操作界面方面，还是在程序发布窗口、历史信息窗口和变量管理窗口方面，在操作和使用上都给用户提供了很大的方便；2001 年，MathWorks 公司又推出了 MATLAB 6.1/Simulink 4.1，其虚拟现实工具箱给仿真结果在三维视景下显示带来了新的解决方案；2003 年 6 月推出了 MATLAB Release 13，即 MATLAB 6.5/Simulink 5.0，在核心数值算法、界面设计、外部接口和应用等诸多方面有了极大的改进；2004 年正式推出 MATLAB Release 14，即 MATLAB 7.0/Simulink 6.0，这是一个具有里程碑意义的版本。此后，几乎每年的 3 月和 9 月 MathWorks 公司都会推出当年的 a 版和 b 版。

MATLAB 是目前国际上最流行的科学计算与工程仿真软件工具之一，现在的 MATLAB 已经不仅是过去的“矩阵实验室”，它已经成为具有广泛应用前景的、全新的计算机高级语言，可以说它是“第四代”计算机语言。自 20 世纪 90 年代以来，美国和欧洲各国已将 MATLAB 正式列入研究生和本科生的教学计划，MATLAB 软件已经成为应用代数、自动控制理论、数理统计、数字信号处理、时间序列分析和动态系统仿真等课程的基本教学工具，

成为学生必须掌握的基本软件之一。在研究所和工业界，MATLAB 也成为工程师必须掌握的一种工具，被认为是进行高效研究与开发的首选软件工具之一。

1.1.2 MATLAB 7.1 系统

自 MATLAB 6.5 版本以后，各版本的界面风格都很相似，操作原理也大同小异。无论采用哪个版本，使用及操作都不会有很大的变化。本书采用的是 2005 年 9 月发布的 MATLAB 7.1。在安装过程中，如果是 Windows XP 系统，会很顺利；如果是 Windows Vista 及更高版本的 Windows 系统，那么在安装好并运行 MATLAB 软件时，会出现启动错误信息，如图 1-1 所示。

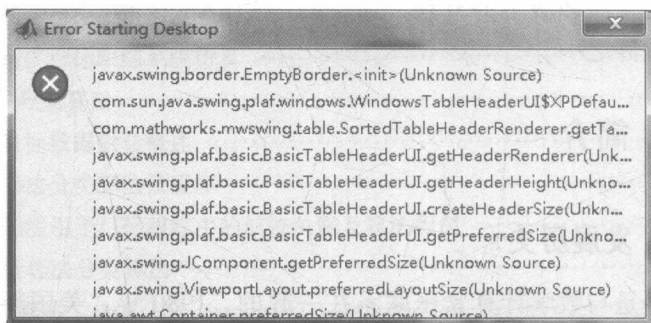


图 1-1 启动错误信息

这时可右键单击桌面上的快捷图标或者在“开始”→“所有程序”中找到 MATLAB，单击“属性”按钮，勾选“以兼容模式运行这个程序”复选框，在下拉列表框中选择 Windows Vista 即可，最后单击“确定”按钮，如图 1-2 所示。重新运行后能正常启动和使用。

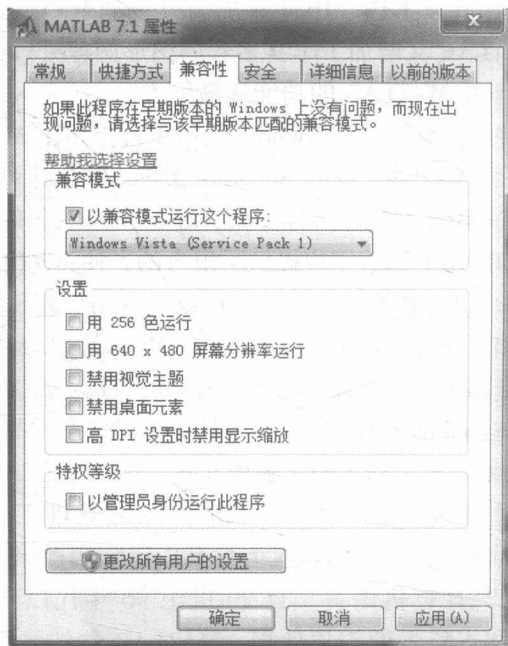


图 1-2 兼容模式设置

MATLAB 7.1 的系统界面如图 1-3 所示，它与之前的 7.0 版、6.5 版系统界面相差无几。系统界面主窗口中包括主菜单、工具栏、当前目录窗口（Current Directory）、工作空间窗口（Workspace）、命令历史窗口（Command History）等。最核心的部分是命令窗口（Command Window），它在界面的右下侧。经常地，在移动各窗口时会发生窗口布局紊乱，如果想恢复系统默认布局，可以单击主菜单 Desktop→Desktop Layer→Default，恢复默认窗口布局。

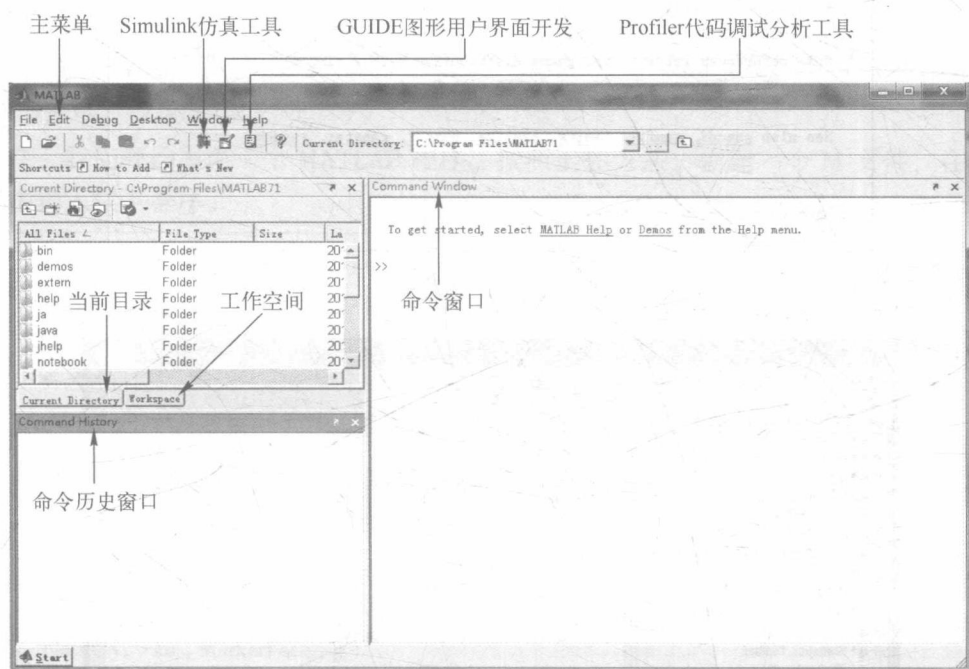


图 1-3 MATLAB 7.1 的系统界面

在命令窗口中，可以执行 MATLAB 的语句指令。例如，想得到 $\sin(\pi/4)$ 的值，在命令窗口中输入 `sin(pi/4)`，可以得到如下结果：

```
>> sin(pi/4)
ans = 0.7071
```

另外，如果遇到任何不懂的函数，都可以直接在命令窗口输入 `help`，查看该函数的功能和实例。通过 `help` 查阅 MATLAB 的文档说明，是每个初学者快速掌握 MATLAB 的有效方法。例如，在做目标跟踪时，噪声符合伽马分布，那么如何用这个伽马分布的函数呢？可以在命令窗口中输入 `help gamrnd`，得到如图 1-4 所示的结果。

如果想再进一步查看它的使用实例，可以使用 `doc gamrnd` 命令，则有关于该函数的使用例子，如图 1-5 所示。

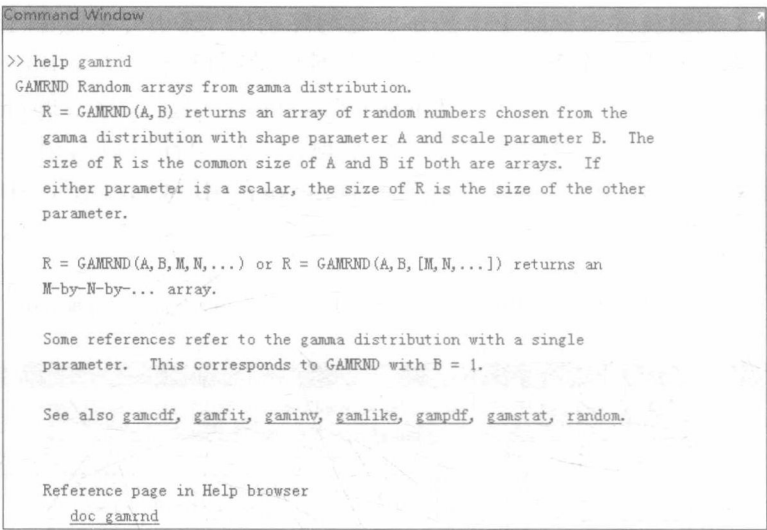


图 1-4 help 帮助系统

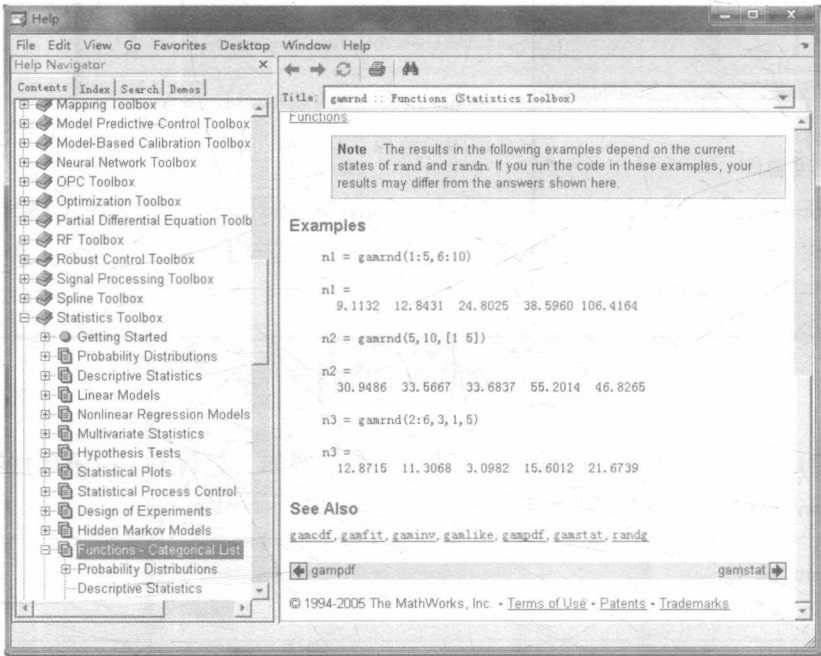


图 1-5 help 文档说明

1.1.3 M-File 编辑器的使用

当用户要运行的指令较多时，不能总在命令窗口中调试，那样太费时间。MATLAB 的 M 文件编辑器能解决这一问题，用户可以将一组相关的指令编辑在同一个 ASCII 码命令文件中，即所谓的编程。在主菜单下面，第一个快捷工具即为新建一个 M 文件，单击它可以建立一个未命名的 M 文件，也可以通过主菜单中的 File→New→M-File 建立一个新的文档，如图 1-6 所示。M 文件编辑器包含主菜单、工具栏、代码编辑窗口等。

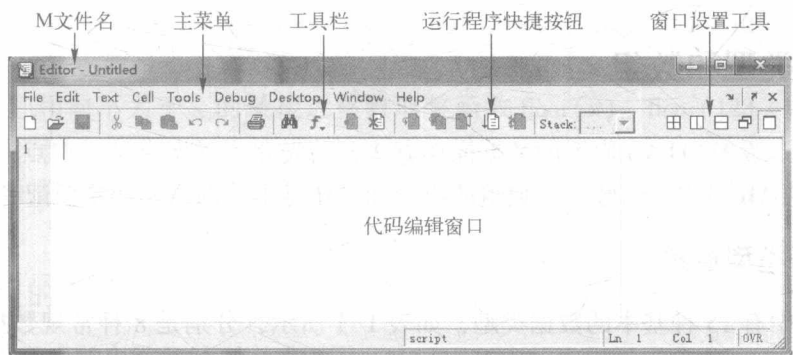


图 1-6 M 文件编辑器

现在，我们开始写第一个 MATLAB 程序。按照上述方法，创建一个 M 文件，在代码编辑窗口中输入以下程序。

```
%%%%%%%%%%  
% 程序说明：第一个 M 程序，做简单的数值计算、输出、画图  
%%%%%%%%%%  
% 数值计算，语句结束不加分号，可以在命令窗口查看结果  
value=sin(pi/4)+cos(1)+log(2)  
% 在命令窗口输出文本  
disp('Hello World');  
% 画一个正弦函数  
t=0:0.5:2*pi;  
y=sin(t);  
plot(t,y,'-ko','MarkerFace','g')
```

代码输入完毕，按 Ctrl+S 组合键或单击 M 编辑器的“保存”按钮，把 M 文件名写为 example1.m，然后单击工具栏中运行程序按钮（Run），运行结果如图 1-7 所示。

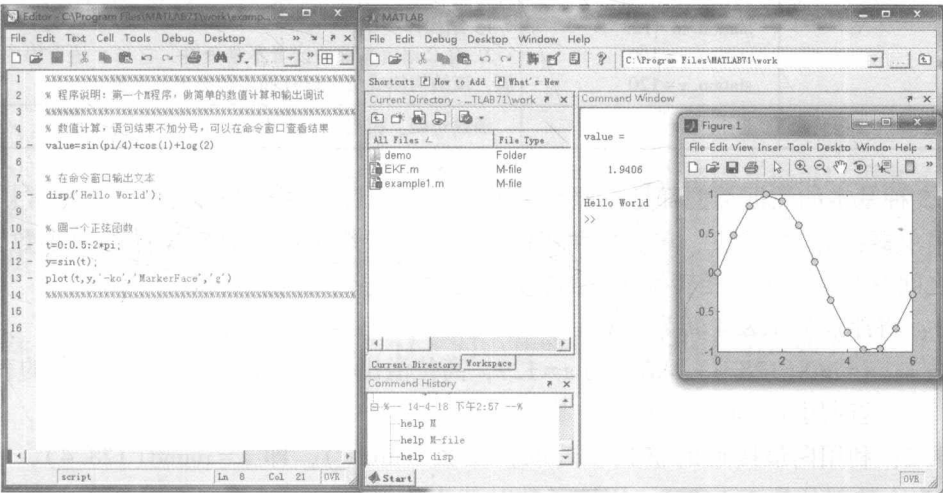


图 1-7 第一个程序运行结果

1.2 数据类型和数组

本节重点介绍 MATLAB 内置的数据操作方法，首先介绍数据类型，重点介绍数值类型，然后介绍 MATLAB 最重要的数组。数组的概念和操作是本节的重点，希望读者熟练掌握。

1.2.1 数据类型概述

MATLAB 中有 15 种基本的数据类型，如表 1-1 所示，分别是 8 种常规数据类型（int8、uint8、int16、uint16、int32、uint32、int64、uint64）、单精度浮点数（single）、双精度浮点数（double）、逻辑数据类型、字符串类型、元胞数组、结构体、函数句柄。另外，为了和高级语言交叉编译，MATLAB 有用户自定义的面向对象的类类型和 Java 类类型。

表 1-1 MATLAB 中的数据类型

数据类型	示例	说明
int8, uint8 int16, uint16 int32, uint32 int64, uint64	a=uint16(8000) b=int8(123.5)	1. 有符号和无符号的整数类型； 2. 大部分整数类型占用比浮点类型更少的内存空间； 3. 除 int64 和 uint64 类型外的所有整数类型都可以用在数学计算中
single	single(383.21)	1. 单精度浮点类型； 2. 它和双精度类型相比，占用内存空间少
double	123.45 4+1.234i	1. 双精度浮点类型； 2. 它比单精度浮点类型表示范围广，是 MATLAB 默认数值类型
logical	a=randn>0.5	1. 逻辑数据类型； 2. 如果 randn 的值大于 0.5，则 a 得到的是逻辑值 1，否则为 0
char	'Hello World!'	字符串类型
cell array	A{1,1}='Jack' A{1,2}=25 A{1,3}=[1,2;3,4]	1. 元胞数组类型； 2. 数组元素可以是不同的数据类型
structure	station.id=123 station.x=100 station.y=120	1. 结构体类型； 2. 有 C 语言基础的读者不难理解，它与 C 语言中的结构体类似，该结构体成员可以存储多种类型的数据
函数句柄	@sin	函数句柄，相当于一个指针

这 15 种基本的数据类型都是按照数组的形式在内存中存储和操作的。8 种常规数据类型可以简单概括为“整数”，单精度和双精度浮点类型数据可以简单概括为“小数”。在这两种数之间有时会出现互相转换，下面通过例子说明它们之间的转换。

例 1：将浮点数 128.4 转为整数。

方法 1：a=int8(128.4)，结果为 a=127，高位溢出，因为 128.4 超出了 int8 的表示范围（ $-2^7 \sim 2^7$ ）。这时用 a=int16(128.4)，则 a=128。

方法 2：利用向最接近的整数靠近取整函数 round()，即 a=round(128.4)，得到 a=128，因为小数部分是 0.4<0.5，用该函数时若小数部分大于等于 0.5 则舍弃小数部分，整数部分加 1，如 round(128.5)=129。

方法3: 利用向0取整函数 $\text{fix}(x)$, 即 $a = \text{fix}(128.4)$, 结果为 $a = 128$ 。 $\text{fix}(-128.4) = -128$ 。

方法4: 利用向不大于 x 的最接近整数取整函数 $\text{floor}(x)$ 。 $\text{floor}(128.4) = 128$, $\text{floor}(-128.4) = -129$, 请注意其与 $\text{fix}(x)$ 函数的区别。

方法5: 利用向不小于 x 的最接近整数取整函数 $\text{ceil}(x)$ 。 $\text{ceil}(128.4) = 129$, $\text{ceil}(-128.4) = -128$ 。请注意其与 $\text{floor}(x)$ 的对比。

1.2.2 数组的创建

按照数组元素的个数和排列方式不同, MATLAB 中的数组可以分为:

- (1) 没有元素的空数组 (empty array);
- (2) 只有一个元素的标量 (scalar), 它实际上是一行一列的数组;
- (3) 只有一行或一列元素的向量 (vector), 分别叫作行向量和列向量, 也统称为一维数组;
- (4) 普通的具有多行多列的二维数组;
- (5) 超过二维的多维数组。

例1: 创建一个空数组 A。

```
>> A = [] % 矩阵右边是一个中括号包围的空矩阵
```

例2: 创建一个一维的数组 A。

方法1: $A = [1, 2, 3, 4, 5]$ 。

方法2: $A = \text{ones}(1, 5)$, 结果为:

```
A =  
    1    1    1    1    1
```

方法3: 通过冒号来创建。

```
>> A = 1:5  
A =  
    1    2    3    4    5
```

例3: 创建一个二维数组 A。

方法1: $A = [1\ 2\ 3; 4, 5, 6; 7\ 8\ 9]$ 。注意元素之间可以用空格隔开也可以用逗号隔开, 同时分行符号必须用分号。

方法2: $A = \text{zeros}(3, 3)$, $A = \text{ones}(3, 3)$ 等。创建一个3行3列的, 每个元素为0或1的数组。

方法3: $A = [1:3; \text{linspace}(4, 6, 3); 7\ 8\ 9]$, 在命令窗口的运行结果如下:

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

冒号的作用, 如 $1:N$, 它表示的意思是从 1 开始, 每次增加 1 直到 N ; 若指定增长步长, 如 $1:2:N$, 则表示从 1 开始每次增加 2, 一直到小于等于 N 的最大整数。例如输入 $1:2:10$ 得到以下结果:

```
>> 1:2:10
ans = 1    3    5    7    9
```

`linspace(start, stop, n)` 函数的作用是产生一个从 `start` 开始的, 最后一个是 `stop` 的等差数列, 公差为 $(\text{stop} - \text{start}) / (n - 1)$ 。

例 4: 创建一个三维数组。

创建三维或更高维的数组, 无法直接赋值得到, 需要通过指定索引把二维数组扩展成多维数组, 或者通过 MATLAB 内部的函数实现, 如 `ones()`、`zeros()`、`cat()` 等。

方法 1: 通过索引将二维数组扩展成多维数组。

```
A=zeros(2,2,3)           % 定义一个三维数组 A 并将其所有元素初始化为 0
A(:,:,1)=[1,2;3,4]       % 将阵列 1 赋值一个 2 行 2 列的矩阵
A(:,:,2)=[5,6;7,8]       % 将阵列 2 赋值一个 2 行 2 列的矩阵
A(:,:,3)=[9,10;11,12]    % 将阵列 3 赋值一个 2 行 2 列的矩阵
```

在这里, 冒号所在的位置表示该维的全部索引。例如, `A(:,:,1)` 表示阵列 1 的所有行 (2 行) 和所有列 (2 列)。

方法 2: 用 `cat()` 函数创建数组。

```
A=[1,2;3,4];
B=[5,6;7,8];
C=cat(3,A,B)             % 按照第 3 维将 A 和 B 连接起来
```

注意, 在创建任何一个数组时, 建议先用 `zeros()` 函数初始化, 再对各元素赋值。

1.2.3 数组的属性

MATLAB 中提供了大量的函数, 用于返回数组的各种属性, 包括数组的排列结构、数组的大小、维度、数组数据类型, 以及数组在内存中占用的空间情况等。这里通过例子重点介绍数组的大小和维度的获取方法。

例 1: 用 `size()` 函数获取任意一个数组的大小。

```
>> A=[];                 % 空数组
>> size(A)
ans =
     0     0
>> B=[1,2,3;4,5,6];      % 2 行 3 列的数组
>> size(B)
ans =
     2     3
```


可见 `size(A)` 函数得到的是数组的行和列。另外也可以用 `length(A)` 函数, 当 `A` 是一维数组时, `length(A)` 返回的是 `A` 数组的元素个数; 当 `A` 是二维数组时, `length(A)` 返回 `size(A)` 得到的行和列中较大的那个值。

例 2: 用 `length` 获得矩阵 `A` 的每一维度的元素个数。

```
>> A=randn(3,4,5,2);           % 用随机函数产生了一个 4 维数组
>> n1=length(A(:,4,5,2))        % 返回第 1 维的长度
n1 = 3
>> n2=length(A(3,:,5,2))        % 返回第 2 维的长度
n2 = 4
>> n3=length(A(3,4,:,2))        % 返回第 3 维的长度
n3 = 5
>> n4=length(A(3,4,5,:))        % 返回第 4 维的长度
n4 = 2
```

例 3: 获得数组的维度。

空数组、单个元素、一维数组, 在 MATLAB 里都被视为二维数组, 因为它们都至少具有两个维度 (至少具有行和列两个方向)。可以用 `ndims()` 函数获取数组的维度。

```
>> A=[];
>> ndims(A)
ans=2
>> A=zeros(2,3);
>> ndims(A)
ans = 2
>> A=randn(2,2,3);
>> ndims(A)
ans = 3
```

1.2.4 数组的操作

数组的操作主要有对数组的索引和寻址、数组的裁剪和元素删除、数组形状的改变、数组的运算及数组元素的排序等。下面通过例子来讲解数组的主要操作方法。

例 1: 对数组元素的索引与寻址。

```
>> A=rand(3,4)                  % 用随机分布函数产生一个 3 行 4 列数组
A =
    0.9501    0.4860    0.4565    0.4447
    0.2311    0.8913    0.0185    0.6154
    0.6068    0.7621    0.8214    0.7919
>> A(2,3)                       % 双下标索引访问数组第 2 行第 3 列元素
ans = 0.0185
>> A(3)                         % 单下标索引第 3 个元素 (即第 3 行第 1 列元素)
```